Efficient High-order Time Domain Finite Element Methods in Electromagnetics.

Neilen Marais

Dissertation presented for the degree of Doctor of Engineering at Stellenbosch University

Supervisor: Prof. D.B. Davidson
March 2009

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Neilen Marais

Abstract

The Finite Element Method (FEM) as applied to Computational Electromagnetics (CEM), can be used to solve a large class of Electromagnetics problems with high accuracy and good computational efficiency. For solving wide-band problems time domain solutions are often preferred; while time domain FEM methods are feasible, the Finite Difference Time Domain (FDTD) method is more commonly applied. The FDTD is popular both for its efficiency and its simplicity. The efficiency of the FDTD stems from the fact that it is both explicit (i.e. no matrices need to be solved) and second order accurate in both time and space. The FDTD has limitations when dealing with certain geometrical shapes and when electrically large structures are analysed. The former limitation is caused by stair-casing in the geometrical modelling, the latter by accumulated dispersion error throughout the mesh.

The FEM can be seen as a general mathematical framework describing families of concrete numerical method implementations; in fact the FDTD can be described as a particular FETD (Finite Element Time Domain) method. To date the most commonly described FETD CEM methods make use of unstructured, conforming meshes and implicit time stepping schemes. Such meshes deal well with complex geometries while implicit time stepping is required for practical numerical stability. Compared to the FDTD, these methods have the advantages of computational efficiency when dealing with complex geometries and the conceptually straight forward extension to higher orders of accuracy. On the downside, they are much more complicated to implement and less computationally efficient when dealing with regular geometries.

The FDTD and implicit FETD have been combined in an implicit/explicit hybrid. By using the implicit FETD in regions of complex geometry and the FDTD elsewhere the advantages of both are combined. However, previous work only addressed mixed first order (i.e. second order accurate) methods. For electrically large problems or when very accurate solutions are required, higher order methods are attractive. In this thesis a novel higher order implicit/explicit FETD method of arbitrary order in space is presented.

A higher order explicit FETD method is implemented using Gauss-Lobatto lumping on regular Cartesian hexahedra with central differencing in time applied to a coupled Maxwell's equation FEM formulation. This can be seen as a spatially higher order generalisation of the FDTD. A convolution-free perfectly matched layer (PML) method is adapted from the FDTD literature to provide mesh termination. A curl conforming hybrid mesh allowing the interconnection of arbitrary order tetrahedra and hexahedra without using intermediate pyramidal or prismatic elements is presented. An unconditionally stable implicit FETD method is implemented using Newmark-β time integration and the standard curl-curl FEM formulation. The implicit/explicit hybrid is constructed on the hybrid hexahedral/tetrahedral mesh using the equivalence between the coupled Maxwell's formulation with central differences and the Newmark-β method with $\beta = 0$ and the element-wise implicitness method. The accuracy and efficiency of this hybrid is numerically demonstrated using several test-problems.

Opsomming

Die metode van Eindige Element-analise (FEM) is in staat om 'n wye verskeidenheid probleme in Berekende Elektromagnetika (CEM) effektief en akkuraat op te los. Vir wyeband-probleme word oplossings in die tydgebied verkies, en selfs al is tydgebied FEM-metodes toepaslik word Eindige-verskil Tydgebied-metodes (FDTD) meer algemeen gebruik. FDTD is gewild omdat dit beide doeltreffend en eenvoudig is. Die lae berekeningskoste van FDTD spruit daaruit dat dit eksplisiet (m.a.w. sonder matriksoplossings) geskied en die oplossing tot tweede orde akkuraat bepaal in beide tyd en ruimte. FDTD het egter beperkinge wanneer dit kom by die oplossing van sekere geometriese strukture en stelsels wat groot is in vergelyking met die elektromagnetiese golflengte. Die eersgenoemde beperking word veroorsaak deur trapvorming tydens geometriese modellering, en laasgenoemde deur 'n ophoping van dispersiefoute regoor die maas.

Eindige Element-analise is 'n wyer wiskundige raamwerk waarbinne verskeie implementerings van numeriese metodes beskryf kan word. FDTD self is byvoorbeeld 'n realisering van 'n Eindige Element Tydgebied-metode. Tot op hede maak die meeste FETD CEM-metodes gebruik van ongestruktureerde, vormgetroue mase en implisiete tydstap-skemas. Sulke mase pas goed aan by komplekse geometrie, alhoewel implisiete tydstappe geneem moet word om bruikbare numeriese stabiliteit te verseker. Vergeleke met FDTD kan hierdie metodes ook konsepsueel maklik uitgebrei word om meer akkurate resultate te lewer. Die algoritmes is egter heelwat ingewikkelder om te implementeer en voer stadiger uit op reëlmatige geometrieë.

In die verlede is FDTD en implisiete FETD gekombineer in hibriede metodes, wat implisiete en eksplisiete benaderings om die beurt volg. Implisiete FETD word gebruik in gebiede van komplekse geometrie, terwyl die FDTD elders toegepas word. Sodoende word die beste eienskappe van elke algoritme benut. Vorige werk bestudeer egter slegs gemengde eerste-orde metodes (metodes met tweede-orde akkuraatheid). Vir stelsels wat groot is in vergelyking met die elektromagnetiese golflengte, of wanneer baie akkurate oplossings verlang word, is hoër-orde benaderings aanloklik. In hierdie proefskrif word 'n nuwe hibriede FETD metode met arbitrêre ruimtelike orde bekendgestel.

'n Hoër-orde eksplisiete FETD-metode is geïmplementeer. Dit maak gebruik van Gauss-Lobatto samevoeging op reelmatige Kartesiese heksahedrons met sentrale verskille in tyd, toegepas op 'n gekoppelde Maxwell-formulering van die Eindige Element-metode. Dit kan geïnterpreteer word as 'n veralgemening van FDTD met 'n hoër ruimtelike orde. 'n Konvolusielose, perfek-aangepaste, laag-metode (PML) is aangepas vanuit FDTD-literatuur vir maas-afsluiting. 'n $\nabla\times$-ooreenstemmende hibriede maas, wat verbindings van arbitrêre orde tussen tetrahedra en heksahedra toelaat sonder die gebruik van piramidale of prismatiese elemente as tussengangers, word voorgedra. 'n Implisiete FETD-metode met onvoorwaardelike stabiliteit, wat gebruik te maak van Newmark-β tydintegrasie en die standaard $\nabla \times \nabla\times$ FEM formulering, is geïmplementeer. Die implisiete/eksplisiete hibriede word gekonstrueer op die hibriede heksahedrale/tetrahedrale maas deur gebruik te maak van die gelykheid tussen die gekoppelde Maxwell-formulering met sentrale verskille en die Newmark-β-metode met $\beta = 0$, asook die element-tot-element implisietheidsmetode. Die akkuraatheid en die doeltreffendheid van hierdie hibriede metode word numeries gedemonstreer aan die hand van verskeie toetsprobleme.

Acknowledgements

I would like to thank my supervisor Prof D. B. Davidson for his patient support and for leading me into an interesting field of research. I also thank my friends who aided the preparation of this manusript: Paul Smeddle, Wendy-Leigh Vandoolhaeghe and Stefan van der Walt.

Furthermore, I would like to thank my parents Alit and Hanlie and grandma Rita and Janneke for their support and understanding, and also Margo for the odd free meal.

I must thank R. Lech for agreeing to let me use his results, and for providing information. I also thank Lisandro Dalcin who aided me immensely and on very short notice to get petsc4py up and running, thereby making the iterative solution results possible.

Much grattitude must also accrue to the Free and Opensource Software (FOSS) community for providing all the software used in the development of this thesis. From the Ubuntu Linux operating sytem, the LaTeX typesetting system, the EMACS text editor, the Gmsh mesher through to the Python programming language and its libraries to numerous to mention. I would however be remiss for not specifically mentioning the SciPy community, not only for making my work much lighter, but also for the pleasure I had in contributing (rather modestly I might add) to SciPy development.

And to anyone that I have overseen, my apologies!

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Full-wave EM solvers such as the one presented here are needed to solve problems that are electrically too large for the use of quasi-static or lumped approximations, and at the same time electrically too small, or geometrically too complicated for the use of asymptotic methods. A large and growing number of practical problems fall into this category. While military radar and communications applications might be seen as the genesis of the category, the more recent explosion of wireless devices and medical applications have caused continuous and rapid growth [2].

The motivation for the novel method boils down to an attempt to improve the computational efficiency of the solution. As workers in Engineering and other disciplines have become more familiar with the capabilities of computational electromagnetic (CEM) tools, they have demanded the ever more accurate solution of ever more challenging EM problems. In part this demand has been met by the rapid decrease in cost and increase in computational power of computers, but recently this trend has slowed down significantly. Conversely, the rapid advances in computational ability have increased the demands even further. As it becomes possible to solve categories of problems that were previously computationally intractable, new applications for CEM arise, calling again for more computational capability.

One approach to increasing computational capability is the parallelisation of existing methods. Parallelisation is a complex topic and beyond the scope of the current work, but in principle improvements to existing methods should eventually lead to improved parallel methods[1]. Computational software advances such as higher order modelling in general, adaptive refinement methods in general, advanced mesh termination schemes such

[1]Of course, some methods are inherently more parallelisable than others. FEM methods are in general neither the easiest nor the hardest to parallelise. The explicit portion of the proposed method should be very easy to parallelise; the implicit portion is more challenging, but not beyond the norm. The implicit/explicit split may also aid the parallelisation of problems with multiple implicit regions, since independent matrix equations result.

as the perfectly matched layer (PML) for all partial differential equations (PDE) methods, the multi-level fast-multipole method (MLFMM) for the method of moments (MOM) and the hybridisation of full-wave methods with asymptotic methods have drastically enhanced the numerical EM modelling capabilities available to analysts. Some problems that would require parallel computing[2]using older computational methods – even assuming current hardware – can be solved comfortably on a high end PC using the appropriate advanced software methods.

The nature of the software advances can be put in two categories, viz. those that can be seen as fundamental improvements to the numerical formulation, and those that are "merely" using existing methods in a new or smarter way. The complication involved in implementing the latter category should in no way be underestimated. Methods such as the MLFMM, the PML and perhaps higher order modelling can be put in the former category; adaptive refinement methods and hybrid methods in the latter. The currently proposed method would fall in the latter category.

Not all improved methods are applicable to all problems. For instance, the MLFMM is based on the MOM; it is well known [2] that the MOM is a very good choice for unbounded problems consisting largely of highly conductive metallic structures, and this remains true of the MLFMM. However, there is an important caveat: the MLFMM is only beneficial when the problem is electrically large. For solving electrically large problems the MLFMM is often orders of magnitude faster than the MOM. On the other hand, the standard MOM is often more efficient at solving small or mid-sized problems very accurately[3]. One may then further categorise software advances into those that benefit only particular problem classes, and those that are almost universally beneficial. The MLFMM and the asymptotic/fullwave hybrid methods can be put in the former category, and adaptive refinement probably in the latter. The currently proposed method could also be argued to fall in the latter category; at the very least it should deliver a clear advantage for any problem that would have been amenable to time domain FEM in the first place.

This thesis presents a novel time domain, hybrid implicit/explicit finite element method (FEM) full-wave electromagnetics (EM) solver that utilises a higher order spatial field discretisation defined on a hybrid mesh. Broken down into component parts, the solver properties comprise of:

[2]It may rightly be pointed out that even single PCs are today inherently parallel systems due to the availability of multi-core CPUs. However, the problem of parallelisation on a single shared memory system is much easier to address than parallel methods intended to scale to many systems with distributed memory systems.

[3]One may ask if a method is any good if its solutions are not "very accurate." This depends heavily on the problem being solved. For far-field observations such as antenna patterns or radar cross sections, small local field errors often have little impact. Other problems, such as determining the circuit parameters of a microwave device may require very accurate local field solutions.

time domain Results are obtained by starting with initial conditions and then simulating the time-evolution of the electromagnetic field in the problem domain in response to both the initial conditions and applied sources at discrete time-steps. This is in contrast to frequency domain solutions were all calculations are done at a fixed frequency.

hybrid implicit/explicit Implicit formulations require the solution of a matrix equation at every time-step, while explicit formulations potentially do not. Explicit formulations commonly involve much less computational work per time-step, but demand a minimum geometry-dependent time-step size (Δt) to ensure stability. Implicit methods can be unconditionally stable, implying that any value of Δt is stable; however Δt is always bound by accuracy constraints. The optimal choice of implicit or explicit methods is problem dependent. In an implicit/explicit hybrid, implicit and explicit methods can be used in different regions of the problem.

finite element method (FEM) A well known numerical method for approximating the solution of PDEs in a discrete setting.

full-wave electromagnetics solver Problems are solved taking into account the full spectrum of classical electromagnetic wave interactions as described by Maxwell's equations. Such solvers are used extensively in the RF and Microwave Engineering fields.

higher order field discretisation As part of any numerical technique, the electromagnetic fields are approximated using a finite (i.e. discrete) set of basis functions. Piece-wise polynomial functions are commonly used; higher order discretisation refers to the use of polynomials of degree higher than 1. Higher order polynomials have better approximation properties, potentially leading to more efficient solutions.

hybrid mesh In the FEM, and indeed many other numerical methods, the problem geometry is defined by a mesh, or tessellation. A mesh approximates the original geometry using a number of similar mesh elements. Many elemental shapes can be used; some elemental shapes ease computation, while other shapes are better at modelling complex geometries. A hybrid mesh can utilise multiple elemental shapes to model a geometry, allowing an optimal choice to be made depending on the local geometry of a particular problem region.

The following section outlines the contributions made in this thesis, and the subsequent sections briefly motivate the particular methodologies employed in the proposed method. This is followed by a chapter summary.

1.1 Contributions

This thesis first presents the necessary theory and prior art needed to implement the proposed method. Time-domain FEM in the electromagnetics field is currently rapidly developing, and the cohesive summary of related topics presented here may be seen as a contribution in its own right. The construction of the hybrid mesh presented here is technically novel, and its extension to arbitrary order field modelling represents a further contribution.

The hybrid implicit/explicit time-stepping scheme presented here is due to Rylander [3], [4], but its description in general terms applicable to arbitrary order field discretisation is novel. The implementation of the PML [5] and also the total field/scattered field formulation [6] for waveguide problems [7] are closely related to their finite difference time domain (FDTD) counterparts. The description of a convolution-free PML method in a mathematically consistent time-domain FEM setting is in the present author's knowledge unique, and also provides additional insight into the FDTD PML implementation. The description of the total field/scattered field formulation in terms of FEM concepts applicable to arbitrary order field discretisation is also novel.

1.2 Motivation for Time Domain Solution

Full-wave solvers can be formulated in the frequency-domain (FD) or the time-domain (TD). Both approaches have their respective merits and disadvantages, but frequency-domain solvers have a longer history in the field of electromagnetic solvers. The first, and by most counts still the primary contender in the time-domain full-wave EM field is the FDTD. In the EM FEM field, the investigation and use of time domain methods only really started to expand in the 21st century. Some advantages often claimed for time-domain over frequency-domain methods are:

- Time domain methods treat impulsive behaviour naturally. Wide band data can be generated in a single simulation.
- Time domain methods are needed to implement explicit methods. Moreover, the matrices that arise in implicit time-domain methods are often better conditioned than those from frequency-domain methods.
- The ability to directly model non-linear and time-dependent materials.

Of course no method is a panacea; for instance the wide-band advantage of TD formulations have been eroded somewhat by the development of adaptive frequency sampling methods, and FD methods tend to fare better with narrow-band resonant structures. Given the success of the FDTD and the relative novelty of time-domain FEM for EM problems, the further investigation of the possibilities presented by the latter is clearly worthwhile.

1.3 Motivation for Higher Order Approximation

One of the main challenges when PDE methods are used to solve Maxwell's equations is the issue of numerical dispersion. Numerical dispersion occurs when the numerical propagation speed differs from the real speed of light in the medium begin considered. Numerical dispersion results in global errors throughout the whole computational domain because of its cumulative nature. What starts out as a small local error grows into a significant error in phase as it travels further away. The result is spurious constructive or destructive interference. On electrically small problems this is not much of an issue, and inaccuracies such as geometrical modelling and local field truncation dominate. As electrically larger problems are solved, the phase error grows without bound. To counter the phase error, it is necessary to refine the mesh as the problem size grows.

The use of higher order basis functions can dramatically improve the phase-accuracy of the FEM. The phase error behaviour of p'th order elements is [8], [9]:

$$err = O(h^{2p}D) \quad \Rightarrow \quad h \propto \sqrt[2p]{\frac{err}{D}}, \tag{1.1}$$

where err is the total phase error, h the linear mesh element size and D the largest physical dimension of the problem. Since we are dealing with a 3-D discretisation, the number of degrees of freedom is proportional to the cube of the problem dimension:

$$n \propto \left(\frac{D}{h}\right)^3. \tag{1.2}$$

Solving for n in terms of the dimension and error, the following proportionality results:

$$n \propto \frac{D^{3+\frac{3}{2p}}}{err^{\frac{3}{2p}}}. \tag{1.3}$$

The value of n in (1.3) is proportional to the memory use of a FEM code. In the time-domain there is also an error contribution from the time-integration. Second order accurate time-integration methods are the most common, leading to the proportionality[4]

$$\Delta t \propto \sqrt{\frac{err}{D}}, \tag{1.4}$$

where Δt is the time-step size. Then we have the number of time-steps

$$n_t \propto \frac{D}{\Delta t} \propto \frac{D^{\frac{3}{2}}}{\sqrt{err}}, \tag{1.5}$$

[4]In the FDTD literature a similar conclusion is usually reached on the basis of stability rather than accuracy requirements. Since the FDTD is second order accurate both in time and space, the accuracy and stability requirements are asymptotically equivalent.

	cost factor			
	$D \longrightarrow 2D$		$err \longrightarrow \frac{err}{2}$	
order p	memory	CPU	memory	CPU
1	22.6	64	2.8	4.0
2	13.5	38	1.7	2.4
3	11.3	32	1.3	1.8

Table 1.1: CPU and memory cost scaling for different orders of elemental approximation. Shown for a doubling in problem size D keeping the phase error err constant, and for the same problem solved with half the phase error.

where the extra D factor is to account for the fact that in a larger problem the EM waves need longer to propagate. This leads to the CPU runtime proportionality

$$runtime \quad \propto \quad n_t\, n \quad \propto \quad \frac{D^{4.5+\frac{3}{2p}}}{err^{\frac{1}{2}+\frac{3}{2p}}} \tag{1.6}$$

A summary of the CPU and memory cost implications as D and err is scaled is shown in Table 1.1. The benefits of high order modelling are readily apparent. Another motivation for higher order elements is to make p- and hp-refinement methods possible.

1.4 Motivation for the Finite Element Approach

As one of the longest established numerical computational techniques in general, the FEM has had the benefit of much mathematical analysis. This has resulted in the existence of an elegant and general mathematical framework within which FEM methods can be described. Aside from the elegance, which some[5] may find appealing in its own right, it also provides valuable insight into the behaviour of solvers. Casting existing techniques into a finite elements framework often makes it possible to generalise them to higher order modelling or different mesh types. In fact, a significant portion of the work presented here was made possible by describing techniques from the FDTD community in finite elements terms before extending them to higher order methods.

Due to the relative immaturity of the FEM in the electromagnetic context, it stands to benefit greatly from the wealth of FEM work that has already been done in the structural dynamics and other fields. However,

[5]Present author included.

one must tread lightly here; indeed the EM FEM's immaturity stems specifically from earlier failures that were the result of the direct application of mechanical FEM techniques to EM problems. Only after the introduction of the correct FEM spaces for EM problems by Nédélec in the 1980's [10, 11] did the FEM become generally useful for EM problems. Such are the vagaries of history that some of the most useful references used for the current work date from the structural dynamics literature of the 1980's [12], [13], and even the 1970's [14].

1.5 Motivation for Hybrid Meshes

Different mesh types provide different trade-offs. Cartesian hexahedral meshes, of which the FDTD implicitly is the strongest proponent, are simple to generate. From the computational perspective it is also easy to create efficient, stable and accurate explicit EM solvers using them. They have one critical shortcoming in their poor geometrical modelling due to the so-called stair-case approximation [15], [16]. For the FDTD this is somewhat of a problem; for higher order elements built on such a mesh it is catastrophic. Unstructured tetrahedral meshes, on the other hand, offer virtually unsurpassed geometrical modelling. In the EM FEM case they can, however, be fairly computationally expensive to solve. The allure of a hybrid mesh is to get the best of both worlds.

The approach taken in the current work was largely inspired by Rylander's FDTD/FEM hybrid [3, 4]. In essence it can be seen as a FEM mesh that is a combination of mixed first order diagonalised[6] Cartesian hexahedra, and a mixed first order tetrahedral mesh. Rylander in fact made use of the FEM description of the FDTD to prove the stability of the hybrid. The currently proposed hybrid mesh uses a scheme different from Rylander's to connect the two mesh types; it will be seen that this is necessary to enable higher order modelling.

By using tetrahedra only where they are absolutely needed, complex geometries can be modelled with only the minimum of additional computational cost. At the same time solution efficiency benefits from explicit operation in the hexahedral region, and also from the availability of FDTD-like mesh termination techniques. To some extent, the best of both worlds may indeed be had with a hybrid mesh.

1.6 Motivation for Implicit/Explicit Hybrids

Explicit methods are usually more computationally efficient than implicit methods, particularly when meshes that allow diagonalisation are used. The

[6] Often referred to as lumped, or, mass lumped due to historical precedent from the computational structural dynamics community.

choice of time-step Δt for explicit methods is made firstly on the grounds of stability requirements and secondly on the grounds of accuracy. With regular meshes, the limits placed on Δt by both stability and accuracy requirements are generally similar[7]. When highly inhomogeneous meshes are called for, for instance to model geometries with complex details at different scales, the stability requirement can quickly become onerous. The analyst is forced to use very small values of Δt with commensurately long run-times without receiving the benefit of increased accuracy.

Implicit methods are more computationally expensive, but those that are unconditionally stable remove the stability requirements on the choice of Δt; the analyst is free to choose the largest value of Δt that ensures sufficient accuracy. Regarding problems with complex geometries, this often results in implicit methods being more efficient.

The root of the problem is that while Δt is a global parameter, the stability requirements are set by the most stringent local part of the mesh. With an implicit/explicit hybrid, the more computationally expensive implicit method can be only used in the vicinity of complex geometrical detail. The global choice of Δt is therefore not affected by the presence of localised inhomogeneity. The motivation for an implicit/explicit hybrid is in essence similar to that of the hybrid mesh, and using tetrahedra in the implicit region is a natural fit.

1.7 Chapter Summary

Chapter 1 The present introduction.

Chapter 2 Literature survey, focusing on work related to time-domain EM FEM implementations. Particular attention is given to hybrid methods.

Chapter 3 The solution of Maxwell's equations are the reason an EM solver exists. Here, Maxwell's equations are presented both in the language of classical Gibbsian vector calculus and using the language of differential forms. The particular generalised problem geometry to be considered is presented, and two particular PDE forms of Maxwell's equations that form the basis of the numerical formulations are defined. These are subsequently referred to as PDE I, based on the coupled Maxwell's equations, and PDE II, based on the vector Helmholtz equation.

Chapter 4 The finite element formulations that are later used to construct the hybrid methods are presented. The discussion in this chapter is

[7]When higher order field discretisation is used with regular meshes the accuracy requirement is often the more stringent of the two.

not specific to a particular mesh or FEM basis type, but aims to shed light on the general characteristics required for successful time-domain finite element discretisation. The weak forms of PDE I&II are derived, followed by the spatial semi-discretisation[8]. Attention is paid to the correct choice of general FEM functional subspaces for the semi-discretisation of PDE I&II. The semi-discretisations are then re-visited from the perspective of differential forms to provide additional insight. The characteristics of time-discretisation schemes suited to Maxwell's equations are discussed and appropriate methods are presented and briefly analysed. Finally, they are applied to the semi-discretisations to yield the fully discrete update equations.

Chapter 5 Complementing the previous chapter, concrete FEM implementations are presented. The practical implications of some general FEM characteristics are discussed, followed by the introduction of the concrete FEM basis sets used to perform the semi-discretisation. This is followed by a discussion of the operational characteristics of the fully discrete methods. The fully explicit method based on PDE I and higher order diagonalised Cartesian hexahedral elements, and the implicit, unconditionally stable methods based on PDE II form the component parts of the subsequent hybrid system. The explicit method may be seen as a higher order generalisation of the FDTD, and is discussed in this light. Finally some aspects of the software implementation of the above FEM methods using the very high level language Python [17] along with the numpy [18, 19] and scipy [20] array, matrix and scientific computing libraries are discussed.

Chapter 6 The core FEM methods presented in the previous chapter are capable of solving Maxwell's equations numerically, but cannot meaningfully solve real problems[9] on their own. Obtaining practical solutions requires source excitations, methods for extracting the results, and in unbounded problems, methods of terminating the computational volume. The excitation of waveguide modes and the extraction of the scattering parameters are performed with a total field/scattered field formulation. The convolution free UPML approach from the FDTD literature is extended to the higher order explicit FEM setting for mesh termination.

Chapter 7 The hybrid implicit/explicit time-stepping scheme is presented. The structuring of the resulting matrix equation in block diagonal form and the various degree of freedom sub-blocks that make up the system are discussed. Finally, the stability of the resulting system is analysed.

[8] This refers to the discretisation in space, but not time. See §4

[9] With the exception of eigen-problems of course.

Chapter 8 The derivation of an $H(curl)$ conforming discretisation on a hexahedral/tetrahedral hybrid mesh is presented. This hybrid is free of pyramidal elements; instead tetrahedral elements of sufficient order are constrained to conform to the hexahedral discretisation, given certain geometric conformance requirements. The method is extended to arbitrary order discretisations and a convenient numerical implementation strategy presented. Finally numerical eigen-results are presented, demonstrating both the correct working of the currently proposed method and the shortcomings of the higher order pyramidal elements currently available in the literature.

Chapter 9 Techniques for numerically validating EM solvers. Some simple problems that test specific subsystems of a code is presented, along with some example results.

Chapter 10 Problems of practical significance that have published reference results available are solved using the proposed method. They are also solved using an FDTD equivalent method. CPU time benchmarking indicate that the proposed method delivers a significant speedup as compared to the FDTD and a previous lower order hybrid. A brief investigation is also made into the iterative solution of the implicit system matrices.

Chapter 2

Literature Survey

In this chapter a literature survey is presented. Sources in general are cited in place throughout the thesis as needed. This chapter concentrates on material that provide background on time-domain finite element methods in EM. Special attention is given to works that deal with hybrid FEM methods.

2.1 General Time-domain FEM literature

Time-domain EM FEM has matured to the extent that a popular EM FEM textbook includes a chapter on time domain methods [21, §12]. The treatment there is limited to the vector Helmholtz equation based formulations described as PDE II subsequently in this text. A well known review of EM time-domain FEM is [22], which follows up on and greatly expands [23]. While it contains a good deal of material, including the PDE II formulation, it reflects the relative immaturity of time-domain FEM when it was written. It includes some formulations that are known now to work well, such as the EBHD method investigated by the present author [24]. The coupled first-order Maxwell's equations method subsequently refered to as PDE I was first alluded to in [10], while the first actual implementation is in [25]. A useful early work is [26]. Both PDE I and II are treated, but only with explicit time-integration. Applications of time-domain FEM to circuit problems are presented in [27].

2.2 Hybrid Time-Doman FEM

Early FEM-FDTD hybrids were attempted, e.g. [28], but they suffered from instabilities due to unsymmetric coupling between the FEM and FDTD regions. A stable FEM-FDTD hybrid was introduced in [3]. By casting the FDTD as a particular FEM method, it was possible to analyse the stability of the hybrid, showing that the coupling between the FEM and FDTD

regions were symmetrical. In [3] the method is a combination of an unconditionally stable tetrahedral region and a conditionally stable explicit FDTD equivalent region. In [4] the method was extended by using a subtlely different coupling between the FDTD and FEM regions, resulting in a slight improvement in accuracy. The new method also allowed a more elegant proof of stability. In [29] an alternative hybrid based on Nitsche's method is presented. It relaxes the strict curl-conformance between the FDTD and FEM regions, while delivering slightly more accurate results. It has the advantage of eliminating the pyramidal elements used in the previous hybrids to connect the hexahedra to the tetrahedra. A survey of several low-order hybrid EM-solvers is in [30].

Chapter 3

Electromagnetics and Maxwell's Equations

In this and the next chapter, the theoretical foundation for the rest of the document is constructed. Most of the content is review: of basic electromagnetic theory and of some specific aspects of finite element method (FEM) theory applicable to microwave EM problems. The purpose is twofold: to create a clear conceptual basis for further work, and to draw the reader's attention to some pertinent aspects of a rather broad field. In this chapter, the classical continuum electromagnetic (i.e. Maxwell's) equations are formulated in the familiar language of Gibbsian vector algebra, and also in the language of differential forms.

The first section considers the time-domain Maxwell's equations in partial differential equation (PDE) form using vector algebra language, and outlines the subset of the general EM problem to be considered subsequently. Attention is paid to electromagnetic continuity at material discontinuities in the second section. The third section defines the subset of possible electromagnetic problems that are considered throughout the rest of this thesis. Two specific forms of the PDE are defined and the conditions for existence and uniqueness of the solution considered. The fourth section provides a brief overview of differential forms, highlighting only features most pertinent to the following discussion. The fifth and last section casts the electromagnetic problem discussed in the previous sections in the language of differential forms.

3.1 Vector Calculus Description

An electromagnetic problem is fully described by four physical quantities, namely the electric ($\vec{E}$) and magnetic ($\vec{H}$) fields and the electric ($\vec{D}$) and magnetic ($\vec{B}$) flux densities. The relationship between the quantities and

electromagnetic sources is described by Maxwell's equations[1] [31]:

$$\nabla \cdot \vec{D} = \rho \tag{3.1}$$

$$\nabla \cdot \vec{B} = 0 \tag{3.2}$$

$$\frac{\partial \vec{B}}{\partial t} = -\nabla \times \vec{E} - \overline{\overline{\sigma}}_* \vec{H} \tag{3.3}$$

$$\frac{\partial \vec{D}}{\partial t} = \nabla \times \vec{H} - \overline{\overline{\sigma}} \vec{E} - \vec{J}, \tag{3.4}$$

where $\vec{J}$ is the impressed source current, ρ the electric charge density, $\overline{\overline{\sigma}}$ the electric conductivity dyad and $\overline{\overline{\sigma}}_*$ the magnetic conductivity dyad. The conductivities are assumed to be symmetric positive-definite tensors. To close the equations, media dependent constitutive relations between the fields and fluxes need to be specified. The general form is:

$$\vec{D} = \vec{f_D}(\vec{E}, \vec{H}) \tag{3.5}$$

$$\vec{B} = \vec{f_B}(\vec{E}, \vec{H}). \tag{3.6}$$

The simplest materials are linear, isotropic and dispersion-less, and have constitutive equations

$$\vec{D} = \epsilon(\vec{r})\vec{E} \tag{3.7}$$

$$\vec{B} = \mu(\vec{r})\vec{H}, \tag{3.8}$$

where ϵ and μ are simple piece-wise continuous scalar functions of position. The most general material type considered subsequently is linear, time-invariant anisotropic dispersive materials. For such materials, the constitutive equations are:

$$\vec{D} = \overline{\overline{\epsilon}}(\vec{r}, t) * \vec{E} \tag{3.9}$$

$$\vec{B} = \overline{\overline{\mu}}(\vec{r}, t) * \vec{H} \tag{3.10}$$

where $\vec{r}$ is the position vector, t time, $\overline{\overline{\epsilon}}$ and $\overline{\overline{\mu}}$ the unit-impulse time response dyad[2] and $*$ is the time-convolution operator. Throughout the rest of the text the time-convolution operator is implied whenever constitutive relations are applied for dispersive material properties (also for the conductivities $\overline{\overline{\sigma}}$

[1] Strictly speaking (3.1) and (3.2) are not part of Maxwell's equations; they follow from the assumption of conservation of charge[31].

[2] I.e. the $\hat{x}\hat{x}$ component of $\overline{\overline{\epsilon}}(\vec{r}, t)$ is the $\vec{D}$ response to a unit $\hat{x}$-directed $\vec{E}$ field.

and $\bar{\bar{\sigma}}_*$) and will mostly be omitted. No free charges are assumed to exist in the problem domain, hence $\rho = 0$ and (3.1) becomes $\nabla \cdot \vec{D} = 0$.

Maxwell's equations can also be cast in integral form. In a source-free region, neglecting the conductivity terms, the integral form is:

$$\oint_{\partial V} \vec{D} \cdot \hat{n} dA = 0 \tag{3.11}$$

$$\oint_{\partial V} \vec{B} \cdot \hat{n} dA = 0 \tag{3.12}$$

$$\frac{\partial}{\partial t} \int_S \vec{B} \cdot \hat{n} dA = - \oint_{\partial S} \vec{E} \cdot d\vec{s} \tag{3.13}$$

$$\frac{\partial}{\partial t} \int_S \vec{D} \cdot \hat{n} dA = \oint_{\partial S} \vec{H} \cdot d\vec{s} \tag{3.14}$$

3.2 Electromagnetic Continuity at Material Discontinuities

In regions where the material parameters vary smoothly with position, all the field quantities vary smoothly. At surfaces with abrupt material discontinuities, the various field quantities have distinct continuity behaviours. The continuity relations can be determined using the integral form of Maxwell's equations (3.11-3.14).

Consider (3.12) evaluated on the pillbox region straddling two regions with a sharp material parameter discontinuity at the interface shown in Fig. 3.1. Assuming Δa is small enough that $\vec{B}$ can be considered constant

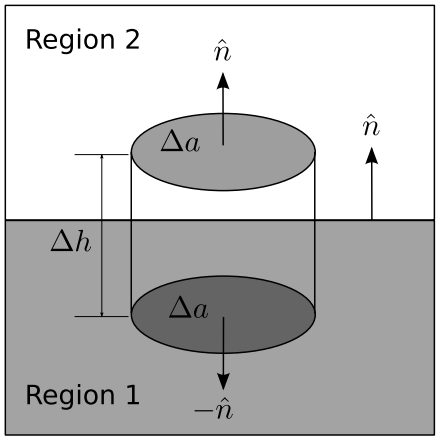


Figure 3.1: Integration volume used to derive the normal continuity condition.

over the end caps, we have

$$\oint_{\partial V} \vec{B} \cdot \hat{n} dA = \hat{n} \cdot (\vec{B}_2 - \vec{B}_1)\Delta a + W = 0, \tag{3.15}$$

where W is the contribution from the sides of the pillbox. Since $W = 0$ in the limit $\Delta h \to 0$, we have normal continuity:

$$\hat{n} \cdot (\vec{B}_2 - \vec{B}_1) = 0, \tag{3.16}$$

with a similar relation holding for $\vec{D}$ by (3.11).

Now consider (3.13) evaluated on the rectangular region straddling to regions with sharp material parameter discontinuity at the interface shown in Fig. 3.2. Assuming Δl is small enough that $\vec{E}$ can be considered constant

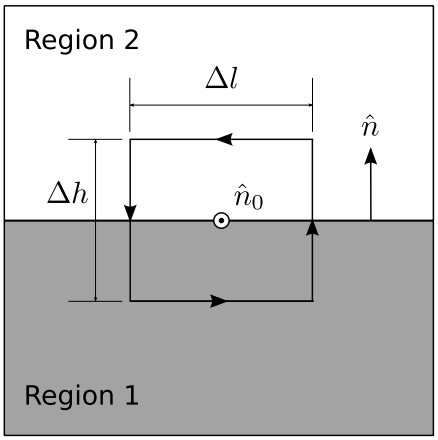


Figure 3.2: Integration volume used to derive the tangential continuity condition.

over the top and bottom integral paths, we have

$$\oint_{\partial S} \vec{E} \cdot d\vec{s} = \hat{n} \times \Delta l(\vec{E}_2 - \vec{E}_1) \times \hat{n}_0 + W \tag{3.17}$$

where W is the contribution from the vertical segments. Since $W = 0$ in the limit $\Delta h \to 0$ and by (3.13), we have

$$\hat{n} \times \Delta l(\vec{E}_2 - \vec{E}_1) \times \hat{n}_0 = -\frac{\partial}{\partial t} \int_S \vec{B} \cdot \hat{n} dA = 0, \tag{3.18}$$

implying tangential continuity

$$\hat{n} \times (\vec{E}_2 - \vec{E}_1) = 0. \tag{3.19}$$

A similar relation holds for $\vec{H}$ using (3.14).

Also of interest are the conditions implied by applying Stokes' theorem to (3.17) and the divergence theorem to (3.15). The former yields

$$\hat{n} \times (\vec{E}_2 - \vec{E}_1) = \lim_{\Delta h \to 0} \Delta h (\nabla \times \vec{E}) = 0 \tag{3.20}$$

and the latter

$$\hat{n} \cdot (\vec{B}_1 - \vec{B}_2) = \lim_{\Delta h \to 0} \Delta h (\nabla \cdot \vec{B}) = 0. \tag{3.21}$$

The relations of (3.20, 3.21) are met if the curl of $\vec{E}$ and the divergence of $\vec{B}$ is "well formed," implying $\vec{E} \in H(curl)$ and $\vec{B} \in H(div)$ [32]. See also the discussion in §4.2.1.

3.3 Problem configuration

The problem is assumed to lie in a finite domain Ω with a boundary $\partial\Omega = \Gamma_1 \cup \Gamma_2 \cup \cdots \cup \Gamma_n$ that is not necessarily simply connected. There may also be interior boundaries of material discontinuity Γ_M. The general problem domain is shown in Fig. 3.3. Note that the problem domain is always assumed

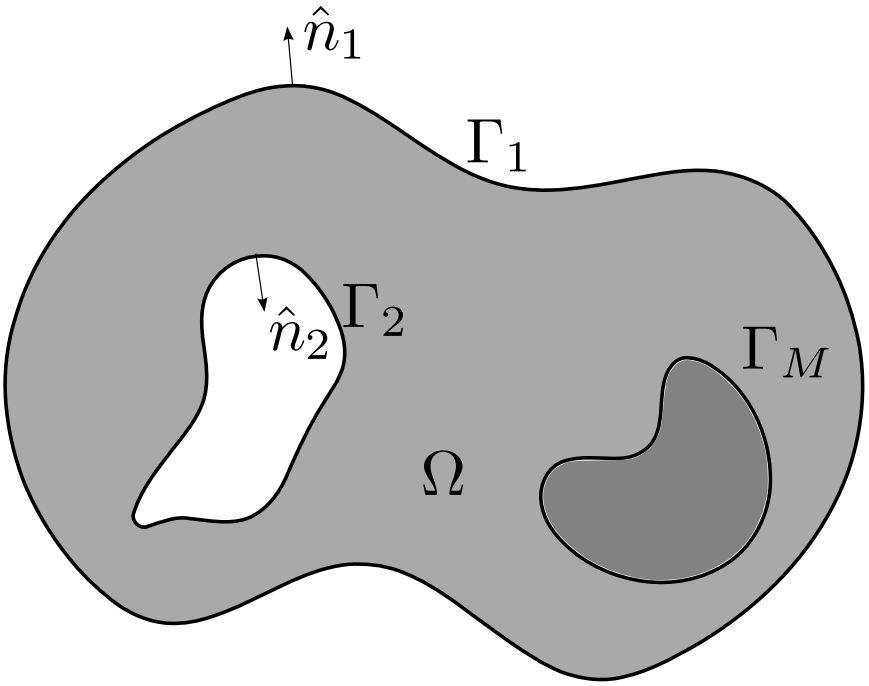


Figure 3.3: A generic inhomogeneous problem domain Ω showing the outward boundary normal orientation and interior boundaries of material discontinuity Γ_M.

to be finite, since the FEM, being a PDE based method, cannot account

for infinite domains naturally. The effect of an infinite domain is usually accounted for by specifying an appropriate boundary condition on Γ_1 or by implementing some well matched numerical absorber close to Γ_1.

The problem is solved for $t > t_0$ subject to initial conditions at $t = t_0$ on $\vec{E}$, $\vec{D}$, $\vec{B}$ and $\vec{H}$. Due to the relations defined by (3.1-3.10) only two quantities or time derivatives thereof can be specified independently. In practice, zero initial values are often specified. See also the subsequent section on existence and uniqueness.

It is not necessary to calculate all four quantities to solve a problem; usually only one or two are solved and the rest derived using the constitutive relations as needed. Subsequently, we consider two forms for solution viz. PDE I and PDE II defined below.

3.3.1 PDE I

$\vec{D}$ and $\vec{H}$ are eliminated using (3.9, 3.10) yielding

$$\frac{\partial \vec{B}}{\partial t} = -\nabla \times \vec{E} - \bar{\bar{\sigma}}_* \bar{\bar{\mu}}^{-1} \vec{B} \text{ in } \Omega \tag{3.22}$$

$$\frac{\partial \bar{\bar{\epsilon}} \vec{E}}{\partial t} = \nabla \times \bar{\bar{\mu}}^{-1} \vec{B} - \bar{\bar{\sigma}} \vec{E} - \vec{J} \text{ in } \Omega. \tag{3.23}$$

Initial conditions are specified as

$$\vec{E}(t = t_0) = E_{ic},\ \vec{B}(t = t_0) = B_{ic} \text{ in } \Omega, \tag{3.24}$$

and boundary conditions

$$\hat{n} \times \vec{E} = \vec{E}_{bc} \text{ on } \Gamma. \tag{3.25}$$

The choice of $\vec{E}$ and $\vec{B}$ or their duals as working variables might seem arbitrary. In the case of discrete solutions, good reasons for this choice exist; see §4.3.3 and also the discussion of *twisted* discrete forms in §4.3.6.

3.3.2 PDE II

By taking the curl of both sides of (3.22, 3.23) and eliminating $\vec{B}$, the Helmholtz vector wave equation results:

$$\nabla \times \bar{\bar{\mu}}^{-1} \nabla \times \vec{E} + (\sigma + \mu^{-1} \sigma_* \epsilon) \frac{\partial \vec{E}}{\partial t} + \mu^{-1} \sigma_* \sigma \vec{E} + \frac{\partial^2 \bar{\bar{\epsilon}} \vec{E}}{\partial t^2} = -\frac{\partial \vec{J}}{\partial t} - \mu^{-1} \sigma_* \vec{J}, \tag{3.26}$$

or its dual in $\vec{H}$. Neglecting magnetic conductivity[3], a much simpler expression results:

$$\nabla \times \bar{\bar{\mu}}^{-1} \nabla \times \vec{E} + \sigma \frac{\partial \vec{E}}{\partial t} + \frac{\partial^2 \bar{\bar{\epsilon}} \vec{E}}{\partial t^2} = -\frac{\partial \vec{J}}{\partial t}. \tag{3.27}$$

Initial conditions are specified as

$$\vec{E}(t = t_0) = \vec{E}_{ic}, \ \frac{\partial}{\partial t}\vec{E}(t = t_0) = \frac{\partial}{\partial t}\vec{E}_{ic} \text{ in } \Omega, \tag{3.28}$$

and boundary conditions

$$\hat{n} \times \vec{E} = \vec{E}_{bc} \text{ on } \Gamma. \tag{3.29}$$

One could formulate (3.26) in terms of $\vec{D}$ or $\vec{B}$, but as for PDE I good reasons exist for the choice of $\vec{E}$ or $\vec{H}$ when dealing with discrete solutions; see the discussion in §4.3.3.

3.3.3 Existence and Uniqueness

PDE I&II are examples of initial boundary value problems (IBVP). An IBVP is well posed if the solution exists, the solution depends continuously on the data and the solution is unique. The existence of solutions to analytic PDEs with analytic initial data is guaranteed by the Cauchy-Kowalewsky theorem [33]. As linear constant coefficient PDEs, PDE I&II above qualify. This also implies that their solution depends continuously on the data.

An electromagnetic field is uniquely determined in a bounded region Ω for $t > t_0$ if the initial values of the $\vec{E}$ and $\vec{H}$ at $t = t_0$ are specified, and the tangential component of one of $\vec{E}$ or $\vec{H}$ are specified on $\partial\Omega$ for $t > t_0$ [31]. Note that the tangential components of exactly one of $\vec{E}$ or $\vec{H}$ must be specified; if none are specified the solution will not be unique, and if both are specified the solution may not exist. In the preceding and subsequent text the tangential $\vec{H}$ field boundary condition is omitted to simplify the discussion. In PDE II it is not possible to directly specify the initial $\vec{H}$ field. However, through (3.2), (3.4), the material constitutive relations and the known continuity behaviour of $\vec{H}$ at material interfaces, the initial value of $\vec{H}$ is implicitly specified through the initial value of $\frac{\partial \vec{E}}{\partial t}$.

3.4 Differential Forms

Differential forms present an alternative mathematical approach to formulating electromagnetics. The language of Gibbsian vector calculus is used

[3] Magnetic conductivity will subsequently be used to implement the Perfectly Matched Layer (PML) mesh termination scheme, and will only be implemented for PDE I. However, even in the context of implementing the PML, the conductivities included in the current formulations are not needed since the losses are incorporated into μ and ϵ.

most prevalently in the electromagnetic engineering literature, but differential forms tend to result in more concise expressions that make certain characteristics of electromagnetics clearer [34]. It may be said that the advantages of differential forms over vector calculus are similar to the advantages of vector calculus over the now-discarded practice of writing everything in Cartesian components [35]. On the scale of level of mathematical abstraction, one might place differential forms between vector calculus and tensor analysis. More precisely, a differential form is a fully covariant, fully antisymmetric tensor. The calculus of differential forms is a self-contained subset of tensor analysis [36]. Since all the basic quantities in electromagnetics are such tensors, no advantage accrues from the additional complexity of general tensor analysis.

A complete discussion of differential forms in electromagnetics is beyond the scope of this text, but the subset presented here is sufficient to express several concepts pertinent to FEM solutions far more clearly than the traditional vector calculus approach. The treatment presented here is intended to be informational rather than mathematically rigorous. For more background on differential forms in general, [37], [38] and [39] are good references, while [34], [35], [36] and [40] deal specifically with differential forms in electromagnetics.

Using differential forms, electromagnetics can be treated as either a 4-D space-time system, or as a 3-D spatial system with separate treatment of the time dimension. While the 4-D treatment results in more concise expressions; the 3-D plus time treatment is more tractable for those mainly familiar with the vector calculus approach and will be used here.

3.4.1 Continuity properties of differential l-forms

As noted in §3.2, different EM quantities obey different continuity conditions. For instance, $\vec{E}$ has tangential continuity and a well-defined curl everywhere, whereas $\vec{D}$ has normal continuity and a well-defined divergence everywhere. Also they behave differently (covariant vs. contravariant, see e.g. [31]) under coordinate transforms. Clearly they are are different kinds of quantities and yet they are both have the same vector field representation. Similarly, the traditional scalar electric potential Φ is a continuous function with a well-defined gradient, while ρ is discontinuous function with no well-defined differential operation. Differential forms make these distinctions explicit.

The distinct differential form types are labeled l-forms, where in n-dimensional space $0 \leq l \leq n$. Each l-form represents a vector space of dimension $\frac{n!}{l!(n-l)!}$. For forms in $\Re^3$, we then have l-forms with l ranging from 0 to 3, with respective dimensions 1, 3, 3, 1. The 0 and 3-forms are dual to scalars, while the 1 and 2-forms are dual to vectors in $\Re^3$. In the following, we will discuss only l-forms in $\Re^3$.

Differential l-forms are characterised by the dimension l of the well-defined oriented integral operator that is associated with them. The line integral

$$\int_s \vec{F} \cdot \hat{s} ds = \int_s F_x dx + F_y dy + F_z dz \tag{3.30}$$

where $\hat{s}$ is tangent to s, is associated with 1-forms. In forms notation, $F = Jdx+Kdy+Ldz$ is the differential form dual to the vector $\vec{F} = J\hat{x}+K\hat{y}+L\hat{z}$. Note how the definition of the differential form is simply the part under the integral sign in (3.30). The surface integral

$$\int_S \vec{G} \cdot \hat{n}\, dA = \int_S G_x dydz + G_y dxdz + G_z dxdy \tag{3.31}$$

where $\hat{n}$ is the surface normal, is associated with 2-forms. In forms notation, $G = Mdydz + Ndzdx + Odxdy$ is dual to the vector $\vec{G} = M\hat{x} + N\hat{y} + O\hat{z}$. Again, the differential form is simply the integrand. The volume integral

$$\int_V H dV = \int_V H dxdydz \tag{3.32}$$

is associated with 3-forms. Following the trend, the differential form $H\, dxdydz$ is dual to the scalar h. Point-wise evaluation as "zero-D integration" is associated with the 0-forms, which are written in forms notation just like a normal scalar.

The requirement that the integrals (3.30-3.32) and the evaluation of a 0-form at arbitrary points are well defined implies certain distinct continuity conditions for each l-form. The 1-form line integral requires tangential continuity along any surface of discontinuity, while the 2-form surface integral requires normal continuity. The 3-form volume integral permits fully discontinuous functions, while the 0-form point-wise evaluation requires full continuity.

Form	Dual	Continuity	Isometric function space
0	scalar	complete	$C^0/H(grad)$
1	vector	tangential	$H(curl)$
2	vector	normal	$H(div)$
3	scalar	none	L^2

Table 3.1: Continuity conditions and function spaces related to various l-forms in $\Re^3$. (See §4.2.1 for a definition of the function spaces.)

When a form is explicitly written out, l is apparent from the multiplicity of the differential, but otherwise there does not seem to be a common convention for notationally designating a quantity as a specific l-form. This is

unlike Gibbsian notation where there is a common convention for denoting quantities as vectors, i.e. boldface or the use of an over-arrow. In practice the nature of the quantity is usually clear from the context. However, we also introduce the notation $F^{(l)}$, where F is a differential l-form. It will be used to make the distinction explicit when needed.

3.4.2 Differential form operations

Treating the full complement of differential form operators is unnecessary for the present discussion. The exterior derivative and the Hodge-star[4] operators are the most important for describing the basic EM relations. The trace and exterior product operators will also be utilised.

The exterior derivative operator d can be seen as the equivalent of a generalised ∇ operator. Depending on the degree of the form that d operates on, it represents the Gibbsian gradient, curl or divergence. An important difference between ∇ and d is that while ∇ depends on the metric (i.e. measure) through some chosen basis system, the d operator is completely metric free [40, pp. 84]. This will be seen to be particularly significant for the discrete case in §4.3.6 where the discrete topological relations can be satisfied exactly, isolating approximations to the metric operators. An l form is mapped to an $l+1$ form through d:

$$dF^{(l)} \rightarrow G^{(l+1)}. \tag{3.33}$$

The equivalence between the various vector differential operators and the exterior derivative is shown in Table 3.2. The l-forms and their isometric function spaces are linked to each other by the exterior derivative, and follows the exact sequence known as the *de Rham complex*

Form	Isometric Space	Exterior Diff. Operator
0	$C^0/H(grad)$	gradient (∇)
1	$H(curl)$	curl ($\nabla\times$)
2	$H(div)$	divergence ($\nabla\cdot$)
3	L^2	none

Table 3.2: Exterior differential operators related to various l-forms in $\Re^3$.

$$\begin{array}{ccccccccc} \text{0-form} & & \text{1-form} & & \text{2-form} & & \text{3-form} & & \\ H(grad) & \xrightarrow[\nabla]{d} & H(curl) & \xrightarrow[\nabla\times]{d} & H(div) & \xrightarrow[\nabla\cdot]{d} & L^2 & \xrightarrow{d} 0. \end{array} \tag{3.34}$$

[4]Subsequently refered to as simply Hodge, or $\star$ in mathematical notation.

The operators listed below the arrows are the equivalent vector differential operators. Another important property of d is

$$d(dF^{(l)}) = 0. \tag{3.35}$$

This is a generalisation of the identities $\nabla \times (\nabla f) = 0$ and $\nabla \cdot (\nabla \times \vec{F}) = 0$.

The Hodge operator depends on the metric of the space. Written as $\star$, it transforms an l-form to an $(n-l)$-form:

$$\star F^{(l)} \rightarrow G^{(n-l)}. \tag{3.36}$$

For forms in $\Re^3$, i.e. $n = 3$ (see e.g. [34, App. F]),

$$\star \star F = F \text{ and } \star^{-1} = \star, \tag{3.37}$$

implying that l- and $(3-l)$-forms are isomorphic in $\Re^3$. This implies

$$F^{(1)} \overset{\star}{\longleftrightarrow} G^{(2)}, \qquad F^{(1)} \cong G^{(2)} \tag{3.38}$$

and

$$F^{(0)} \overset{\star}{\longleftrightarrow} G^{(3)}, \qquad F^{(0)} \cong G^{(3)}, \tag{3.39}$$

where $\cong$ indicates an isomorphism.

The exterior product $\wedge$, often called the wedge product, is defined as

$$F^{(p)} \wedge G^{(q)} \rightarrow H^{(p+q)}, \tag{3.40}$$

and

$$F^{(p)} \wedge G^{(q)} = 0 \text{ if } (p+q) > n. \tag{3.41}$$

The exterior product is super-commutative (see e.g. [36]):

$$F^{(p)} \wedge G^{(q)} = (-1)^{pq} G^{(q)} \wedge F^{(q)}. \tag{3.42}$$

The exterior product takes the place of certain uses of the vector dot and cross products; these uses are summarised in Table 3.3. The basis differentials for 2- and 3-forms are in fact derived from the 1-form basis differentials using the wedge product, e.g. $dydz = dy \wedge dz$; the wedge notation is often used for higher order basis differentials for this reason.

The final operator discussed here is the trace operator. Given an l-form on $\Re^n$, the trace operator defines an l-form on a lower-dimensional $\Re^m$, $m < n$ sub-manifold Γ of $\Re^n$. The new lower-dimensional l-form is defined by some property of the original l-form on Γ. If $m < l$, the trace is zero. The trace of $F^{(l)}$ on Γ is written as $\Upsilon_\Gamma(F^{(l)})$. In $\Re^n$, the trace of a 1-form is defined by its components tangential to a line or surface Γ_l or Γ_s, while the trace of a 2-form is defined by its component normal to a surface Γ_s. The 0-form trace is defined by its values on a line or surface or at a point, while 3-forms have, per definition, no non-zero traces. As an example, specifying $\Upsilon_\Gamma(F^{(1)})$ where Γ_s is a surface is dual to specifying $\hat{n} \times \vec{F}$ on Γ_s, i.e. the tangential component of $\vec{F}$.

Operation		Commutation		Result	Vector Dual
$F^{(0)} \wedge G^{(l)}$	$=$	$G^{(l)} \wedge F^{(0)}$	$\rightarrow$	$H^{(l)}$	scalar mult.
$F^{(1)} \wedge G^{(1)}$	$=$	$-G^{(1)} \wedge F^{(1)}$	$\rightarrow$	$H^{(2)}$	$\vec{F} \times \vec{G}$
$F^{(1)} \wedge G^{(2)}$	$=$	$G^{(2)} \wedge F^{(1)}$	$\rightarrow$	$H^{(3)}$	$\vec{F} \cdot \vec{G}$
$A^{(1)} \wedge B^{(1)} \wedge C^{(1)}$	$=$	$C^{(1)} \wedge A^{(1)} \wedge B^{(1)}$ $B^{(1)} \wedge C^{(1)} \wedge A^{(1)}$ $-A^{(1)} \wedge C^{(1)} \wedge B^{(1)}$ $-B^{(1)} \wedge A^{(1)} \wedge C^{(1)}$	$\rightarrow$	$H^{(3)}$	$\vec{A} \cdot (\vec{B} \times \vec{C})$

Table 3.3: Wedge product operational equivalence with the vector dot and cross products and commutative behaviour.

3.5 Differential Form Problem Description

In this section the Maxwell's equations and systems from §3.1 and §3.3 are cast in the language of differential forms. This description is fully dual to the system described in the standard Gibbsian vector calculus notation. The finite element method derived from the differential forms description in terms of discrete differential forms allows for easier analysis of complex material operators, and will be put to use in later chapters.

3.5.1 EM Quantities

The correct choice of l-forms for representing the electromagnetic quantities can be made by considering the function spaces implied by the electromagnetic continuity conditions summarized in §3.2. Referring to Table 3.1, we can see that $\vec{E}$ and $\vec{H}$ are dual to 1-forms, while $\vec{B}$ and $\vec{D}$ are 2-forms. Following the sequence (3.34), the correct forms for all the other quantities can be determined, and is listed in Table 3.4.

3.5.2 Maxwell's Equations in Differential Forms

Written in differential forms and taking into account the assumed absence of free electrical charge, equations (3.1-3.4) are

$$dD = 0 \tag{3.43}$$

$$dB = 0 \tag{3.44}$$

$$\frac{\partial B}{\partial t} = -dE - \bar{\bar{\sigma}}_* \star H \tag{3.45}$$

Physical Quantity	Units	Vector/Scalar	Differential l-Form
Electric Field Intensity	$V.m^{-1}$	$\vec{E}$	1-form: E
Magnetic Field Intensity	$A.m^{-1}$	$\vec{H}$	1-form: H
Electric Flux Density	$C.m^{-2}$	$\vec{D}$	2-form: D
Magnetic Flux Density	$W.m^{-2}$	$\vec{B}$	2-form: B
Electric Current Density	$A.m^{-2}$	$\vec{J}$	2-form: J
Electric Charge Density	$C.m^{-3}$	ρ	3-form: ρ
Electric Scalar Potential	V	Φ	0-form: Φ

Table 3.4: Physical quantities and their associated l-forms.

$$\frac{\partial D}{\partial t} = dH - \overline{\overline{\sigma}} \star E - J. \tag{3.46}$$

Note how the quantities on the LHS of (3.45) and (3.46) are both 2-forms. The d operator applied to the 1-forms E and H is equivalent to the Gibbsian curl operator, yielding the requisite 2-form result by (3.34). The Hodge $\star$ operator is needed to apply the electric and magnetic conductivity operators to E and H, also yielding 2-forms by (3.38).

The material constitutive relations (3.9, 3.10) can be written as:

$$D = \overline{\overline{\epsilon}}(\vec{r}, t) * \star E \tag{3.47}$$

$$B = \overline{\overline{\mu}}(\vec{r}, t) * \star H, \tag{3.48}$$

but since the $\star$ operator is never used separately from the material operators, they are usually notationally combined into material Hodge operators:

$$D = \star_\epsilon E \tag{3.49}$$

$$B = \star_\mu H. \tag{3.50}$$

The same notation is used for the conductivities, e.g. $\star_\sigma = \overline{\overline{\sigma}}\star$.

3.5.3 PDE I Expressed in Differential Forms

PDE I from §3.3.1 can now be written as:

$$\frac{\partial B}{\partial t} = -dE - \star_{\sigma_*} \star_{\mu^{-1}} B \text{ in } \Omega, \tag{3.51}$$

$$\frac{\partial \star_\epsilon E}{\partial t} = d \star_{\mu^{-1}} B - \star_\sigma E - J \text{ in } \Omega, \tag{3.52}$$

with initial conditions are specified as

$$E(t = t_0) = E_{ic},\ B(t = t_0) = B_{ic} \text{ in } \Omega, \tag{3.53}$$

and boundary conditions

$$\Upsilon_\Gamma(E) = E_{bc}. \tag{3.54}$$

Note that in (3.52) we are using $\star_{\mu^{-1}}$ rather than $\star_\mu^{-1}$. Since $\star_{\mu^{-1}} = \star_\mu^{-1}$, it does not make any difference to the result, but in the case of discrete representations, only $\star_{\mu^{-1}}$ is directly accessible. Using this notation will make the continuum expressions more consistent with the discrete equivalents to be considered later. Note also the repeated Hodge operators for the magnetic conductivity $\star_{\sigma_*}\star_{\mu^{-1}}$. $\star_{\mu^{-1}}$ transforms 2-form B into the isomorphic 1-form H, whereas $\star_{\sigma_*}$ applies the magnetic conductivity material property resulting in a 2-form. Note also the use of the trace operator in (3.54) to specify the tangential electric field at the boundary.

3.5.4 PDE II Expressed in Differential Forms

PDE II (neglecting for simplicity magnetic conductivity) from §3.3.2 can now be written as:

$$d \star_{\mu^{-1}} dE + \star_\sigma \frac{\partial}{\partial t} E + \frac{\partial^2}{\partial t^2} \star_\epsilon E = -\frac{\partial J}{\partial t}. \tag{3.55}$$

Initial conditions are specified as

$$E(t = t_0) = E_{ic},\ \frac{\partial}{\partial t} E(t = t_0) = \frac{\partial}{\partial t} E_{ic} \text{ in } \Omega, \tag{3.56}$$

and boundary conditions

$$\Upsilon_\Gamma(E) = E_{bc}. \tag{3.57}$$

3.6 Conclusion

Maxwell's equations have been reviewed, and some properties of the vector fields that are their solution recounted. Specific forms (PDE I and PDE II) of Maxwell's equations to be considered in subsequent chapters have been defined, and the problem configuration considered for solution has also been defined. The language of differential forms has briefly been described. The application of differential forms to Maxwell's equations has been motivated and described. This chapter has described the problem domain to which the application of FEM methods is described in subsequent chapters.

Chapter 4

FEM Formulations

The aim of this chapter is to put on the table the finite element formulations on which the hybrid method is built in later chapters. To facilitate this, some elements of FEM are very briefly reviewed. Rigorous discussion of these concepts is beyond the scope of this work; more detailed references will be cited judiciously throughout. The FEM is then applied to PDE's I & II as defined in §3.

The first section briefly outlines the Galerkin method of constructing approximate weak projective solutions to PDEs. The second section constructs weak projective forms of PDE's I & II using some of the concepts from the first section. The third section section considers the discretisation of the weak forms in space. The correct choice of approximation space for the various field quantities and their properties are discussed, and the resulting semi-discrete[1] systems presented. Discussion of concrete basis sets for speficic mesh types is however left for the next chapter. The fourth section deals with time discretisation. Appropriate time integration schemes for the two semi-discrete systems are presented, and their properties (including stability) discussed. The fith section briefly considers the sources in error in a FEM solution, and how to optimally balance them.

4.1 Casting PDEs in Projective Weak Form using Galerkin's Method

One might say that from the Engineer's perspective, the projective weak form of problems have the important advantage that they make finite element solutions possible. Fortunately projective solutions are also on a mathematically solid footing. A very readable informal presentation of the basic properties of projective solutions in electromagnetics is given in [41, §3.5].

[1] A semi-discrete system is one where the continuum problem has been spatially discretised but is still continuous in time. This term originates from the structural dynamics literature where space and time discretisations are usually done independently [12].

A more mathematically rigorous treatment is given in the earlier chapters of [32], also in the electromagnetic context. Another useful summary can be found in [42, §2]. The presentation here largely follows [41].

The well known Galerkin's method is subsequently applied to PDE I&II to obtain the corresponding weak forms. Galerkin's method is closely related to variational methods often employed to derive finite element methods – in fact they often lead to identical systems. Galerkin's method is attractive since it can be used even when variational principles for a given problem is not known. Instead of "finding" a variational principle and proving that it is equivalent to a known PDE, Galerkin's method starts with the PDE and derives a weak form. Of course, proving that it satisfies the original PDE is still necessary, but is beyond the scope of this discussion.

4.1.1 Projective Operator Form

To derive a Galerkin representation of a given PDE, it is first written as a linear operator equation

$$\mathcal{L}u = v, \tag{4.1}$$

where $\mathcal{L}$ is a symbolic linear operator and u & v are symbolic representations of the fields; they may be vectors, scalars or combinations thereof. Here u is the unknown solution and v is a known driving function. We can view $\mathcal{L}$ as a transformation rule that maps every u into some v. We define the range of $\mathcal{L}$ as $R_{\mathcal{L}}$, the *linear space* of all functions v that could appear as sources for a given problem. Similarly the domain $D_{\mathcal{L}}$ as the linear space of all possible functions that could be solutions to the problem. The problem statement is now: Find a function u in $D_{\mathcal{L}}$ that satisfies (4.1) for a given $v \in R_{\mathcal{L}}$.

The solutions of the electromagnetic problems considered in this text belong to *inner product spaces*. Inner product spaces are more narrowly defined than linear spaces. An inner product space is a linear space that can be endowed with an *inner product*. Given a linear space $\mathbb{V}$, an inner product of the form $\langle \psi_1, \psi_2 \rangle$ where ψ_1 and ψ_2 are any two functions from $\mathbb{V}$ have the following properties:

$$\langle \psi_i, \psi_i \rangle > 0 \;\forall\; \psi_i \neq\; 0, \quad \langle \psi_i, \psi_i \rangle = 0 \;\text{ iff } \psi_i = 0, \tag{4.2}$$

$$\langle \psi_1, \psi_2 \rangle = \langle \psi_2, \psi_1 \rangle \tag{4.3}$$

$$\langle a\psi_1, \psi_2 \rangle = \langle \psi_1, a\psi_2 \rangle = a\langle \psi_1, \psi_2 \rangle \;\text{ for any scalar } a. \tag{4.4}$$

The inner product norm, often called the natural norm, the L^2 norm or simply the norm, is defined as

$$\|\psi\| = \sqrt{\langle \psi, \psi \rangle}, \tag{4.5}$$

and can be seen as a generalised measure of distance.

Inner products are usually integrals defined over a given domain Ω. The standard inner product for scalar functions is

$$\langle \psi_1, \psi_2 \rangle = \int_\Omega \psi_1 \psi_2 \, dV. \tag{4.6}$$

Note that the definition of this particular inner product requires that all ψ_i functions be square integrable on Ω. Therefore, an inner product space $\mathbb{V}$ endowed with this inner product can contain only square integrable functions, i.e. $\psi_i \in L^2(\Omega) \; \forall \; \psi_i \in \mathbb{V}$.

If $R_\mathcal{L}$ forms an inner product space, the original problem (4.1) can be written as projective equation

$$\langle w, \mathcal{L}u \rangle = \langle w, v \rangle \quad \forall w \in R_\mathcal{L}, \tag{4.7}$$

where w are called the weighting, or testing functions. If we have a set of testing functions $w_i \in \mathbb{V}$ such that they form a spanning set[2] of $R_\mathcal{L}$, and a given u satisfies (4.7) for every $w_i \in \mathbb{V}$, then u also satisfies (4.1). I.e. (4.7) and (4.1) are exactly equivalent.

This statement is a generalisation of vector equality by projection on a linearly independent basis in Euclidean space. The standard inner product for Euclidean vectors is the dot product:

$$\langle \vec{a}, \vec{b} \rangle = \vec{a} \cdot \vec{b}. \tag{4.8}$$

In the same way that two vectors in $\Re^3$ are equal if their projections onto (i.e. inner products with) three independent vectors are equal, $\mathcal{L}u$ and v are equal if their projection onto every $w_i \in \mathbb{V}$ is equal.

As a concrete example, consider the Poisson's equation defined in the region Ω with the boundary $\partial\Omega$ that is the union of a part with homogeneous Dirichlet- (Γ_D) and Neumann (Γ_N) boundary conditions shown Fig. 4.1. We can define the operator as

$$\mathcal{L} = \begin{cases} \nabla^2 u = v & \text{in } \Omega, \\ u = 0 & \text{on } \Gamma_D, \\ \dfrac{\partial u}{\partial n} = 0 & \text{on } \Gamma_N, \end{cases} \tag{4.9}$$

where $\frac{\partial}{\partial n}$ is the directional derivative normal to Γ_N. Using (4.6) as an inner product with (4.7), we can write

$$\int_\Omega w \nabla^2 u \, dV = \int_\Omega w v \, dV. \tag{4.10}$$

[2] A set of functions $b_i \in \mathbb{B}$ is a spanning set of a space $\mathbb{A}$ if they are linearly independent and any function $a \in \mathbb{A}$ can be written as a linear combination of b_i functions, i.e. $a = \sum \alpha_i b_i$. The weights α_i can be seen as the coordinates that represent a using $\mathbb{B}$.

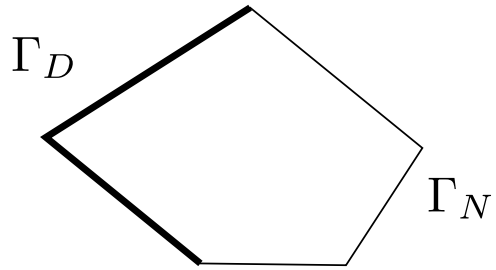


Figure 4.1: Two dimensional region Ω where Poisson's equations hold with part Dirichlet and part Neumann boundary conditions.

Note that on its own, (4.10) is not yet well defined. Firstly, it makes no statement about the behaviour of u on $\partial\Omega$ as is required by $\mathcal{L}$. Secondly we have not completed the definition of the inner product space by specifying the function spaces from which u and v may be chosen. In fact, both issues are interrelated. We fix the boundary conditions by specifying $u \in D_{\mathcal{L}}$, all the twice differentiable scalar functions defined in Ω (as required by the Laplacian in $\mathcal{L}$) that satisfy the specified boundary conditions. Now we can implicitly define $v \in R_{\mathcal{L}}$. Furthermore note that the definition of the inner product requires both $\nabla^2 u$ and v to be square integrable. This is not really a restriction, since solutions that are not square integrable would imply infinite energy in practice.

We say that the projective form made up of (4.10) together with the appropriate definitions of $R_{\mathcal{L}}$ and $D_{\mathcal{L}}$ is strongly equivalent to (4.9), since any solution of the projective form is guaranteed to be a solution to (4.9) as well. While we now have an alternative but exactly equivalent expression of the original problem, it does not really help us to find the solution. More importantly, most real problems do not have exact solutions that can be written down as analytical expressions – the best we can do is find an approximate solution that is in some sense "good enough." The aim here is to use the projective formulation to cast the original problem in a form amenable to approximation by the finite element method. In aid of this, a weak form of the projective formulation is found.

4.1.2 Weak Projective Operator Form

The original, or strong formulation of the problem requires that the solution u and the given boundary have strong regularity properties, i.e. the existence and continuity of derivatives up to the order of the partial derivative operators. Informally, a weak formulation of a problem is one that relaxes the regularity requirements of the space in which solutions are sought[3].

[3]In [41] §3 eq (5.12) a weak form is defined in a different way that, to the present author at least, seems to be at odds with the rest of the literature consulted. In [41]

This is usually done by increasing the regularity requirement on the testing function space and then using integration by parts to transfer one of the differential operators onto the testing functions. A strong solution is always a weak solution too, but the reverse is not necessarily true. A system may or may not have a strong solution, but if it exists the weak solution must be unique and should coincide with the strong solution [42].

Applying Greens first identity (i.e. integration by parts) to (4.10) results in

$$\int_\Omega \nabla w \cdot \nabla u \, dV - \int_{\partial\Omega} w \frac{\partial u}{\partial n} \, dS = \int_\Omega wv \, dV. \tag{4.11}$$

The surface integral term above falls away due to the specification of homogeneous boundary conditions in (4.9):

$$\int_\Omega \nabla w \cdot \nabla u \, dV = \int_\Omega wv \, dV, \tag{4.12}$$

This weak formulation now requires both u and w to be once differentiable and for their first derivatives to be square integrable. Also, w must be square integrable. We label the new domain (solution) and range (testing) spaces respectively $\mathbb{W}_D$ and $\mathbb{W}_R$, i.e. $u \in \mathbb{W}_D$ and $w \in \mathbb{W}_R$. The exact choice of $\mathbb{W}_D$ and $\mathbb{W}_R$ are still left open. Since the original spaces $D_\mathcal{L}$ and $R_\mathcal{L}$ are hard to construct, a common choice is to first enlarge both spaces to that of square integrable functions $L^2(\Omega)$ and then restricting them to the once differentiable subset of $L^2(\Omega)$ that has square integrable first derivatives. This function space is recognisable as the Hilbert space $H(grad, \Omega)$, implying $\mathbb{W}_D = \mathbb{W}_R = H(grad; \Omega)$.

While the weak formulation has simplified the original problem by casting it in terms of simpler function spaces, i.e. $\mathbb{W}_D = \mathbb{W}_R = H(grad, \Omega)$, it still does not provide a method for finding a solution $u \in \mathbb{W}_D$. The space $H(grad, \Omega)$ is infinite dimensional; we can construct a finite dimensional approximation to the true solution by choosing an n-dimensional subset of the domain space $\mathbb{W}_{Dn} \subseteq \mathbb{W}_\mathbb{D}$, and an m-dimensional subset of the range space $\mathbb{W}_{Rm} \subseteq \mathbb{W}_\mathbb{R}$. To solve the approximate weak problem we need spanning sets of functions $\alpha_i \in \mathbb{W}_{Dn}$, $i = 1 \ldots n$ and $\beta_j \in \mathbb{W}_{Rm}$, $j = 1 \ldots m$. The approximate solution $\tilde{u}$ is written as

$$\tilde{u} = \sum_i^n u_i \alpha_i, \tag{4.13}$$

the weak form is formed by enlarging the testing space $\mathbb{W}$ to a Hilbert space such that $\mathbb{W} \supseteq R_\mathcal{L}$. Since $R_\mathcal{L}$ is included in the enlarged $\mathbb{W}$, a projective solution using $\mathbb{W}$ as the testing space must already satisfy the strong form, hence this does not seem to weaken the formulation in any way. It also seems to be at odds with a later part of the section where the conventional integration by parts route is used to form a weak formulation of Poisson's equation. This would not be possible if $\mathbb{W} \supseteq R_\mathcal{L}$, since functions in $R_\mathcal{L}$ are not required to be differentiable. In spite of this, the rest of §3.5 is in the present author's opinion one of the most accessible introductions to the material.

where u_i are the unknown numerical coefficients making up the discrete degrees of freedom (DOFs) of the system. To solve the unknown DOFs, we set

$$\langle \beta_j, \mathcal{L} \sum_i^n u_i \alpha_i \rangle = \langle \beta_j, v \rangle, \quad j = 1 \dots m, \tag{4.14}$$

which by the linearity of $\mathcal{L}$ is equivalent to

$$\sum_i^n \langle \beta_j, \mathcal{L}\alpha_i \rangle u_i = \langle \beta_j, v \rangle, \quad j = 1 \dots m. \tag{4.15}$$

This may be identified as a matrix equation. Solving for u_i, $i = 1 \dots n$ and substituting into (4.13) yields a finite dimensional approximation to (4.1).

In general $\mathbb{W}_{Dn}$ and $\mathbb{W}_{Rm}$ do not have to be equal, yielding an approximate weak projective method. Specifying $\mathbb{W}_{Dn} = \mathbb{W}_{Rm}$ results in the *Galerkin* method. While it is not necessary for the spanning functions α_i and β_i to be identical – they merely need to span the same spaces – it is often convenient to choose them equal.

4.1.3 Outline of Galerkin's Method

In summary, the Galerkin method is applied to a given PDE by

- casting the PDE in operator form $\mathcal{L}u = v$,
- choosing an appropriate inner product $\langle \cdot, \cdot \rangle$,
- converting the operator form into a projective form $\langle w, \mathcal{L}u \rangle = \langle w, v \rangle$,
- converting $\langle w, \mathcal{L}u \rangle = \langle w, v \rangle$ into weak form using integration by parts,
- choosing an appropriate finite dimensional function space to use for both the solution ($\mathbb{W}_{Dn}$) and testing ($\mathbb{W}_{Rm}$) spaces, with $\mathbb{W}_{Rm} = \mathbb{W}_{Dn}$,
- finding spanning sets of functions $\beta_j \in \mathbb{W}_{Rm}$, $j = 1 \dots m$ and $\alpha_i \in \mathbb{W}_{Dn}$, $i = 1 \dots n$,
- finally solving (4.15) as a matrix equation to obtain the coefficients of the approximate solution (4.13).

One important detail has been omitted by considering only homogeneous boundary conditions up till now. In the case where an inhomogeneous Dirichlet condition is prescribed on part or all of Γ_D, the solution must be found in the function set $\mathbb{W}_{Dn}$ that belongs to $H(curl, \Omega)$ and satisfies the inhomogeneous Dirichlet condition, while the testing space $\mathbb{W}_{Rm}$ must satisfy the homogeneous Dirichlet condition on all of Γ_D. While the two spaces are no longer equal, the testing space is a strict subset of solution

space ($\mathbb{W}_{Rm} \subset \mathbb{W}_{Dn}$). Because Dirichlet boundary conditions have to be introduced as direct restrictions on the solution and testing spaces, they are called essential conditions.

An inhomogeneous Neumann condition is prescribed in a weak, projective sense by taking the surface integral term resulting from integration by parts to the RHS of (4.11) and substituting in the desired value of $\frac{\partial u}{\partial n}$. We also see that the surface integral term falls away on Γ_D, since the testing functions are constrained to be zero there.

Following the process from PDE to operator form to projective form, then to weak projective form where the solution and testing spaces are no longer coincident with the original operator's domain and range spaces (although they should intersect around the strong solution if it exists), finally through to a finite dimensional approximation of the weak solution, one may ask if the end result meaningfully solves the original problem. Fortunately the answer is affirmative, backed by an extensive body of mathematical analysis, provided that suitable discrete function spaces are used for $\mathbb{W}_{Dn}$ and $\mathbb{W}_{Rm}$.

Assuming that the spaces are straight forward to obtain can be dangerous; early FEM solutions of Maxwell's equations suffered for not using the right spaces. Indeed, the nature of the appropriate spaces was unknown to many practitioners as recently as the early 1990's and was almost completely unknown to the computational electromagnetics community before Nédélec's seminal paper [10] in 1980. Currently they are quite well known; these spaces, suitable spanning basis functions sets and the expected accuracy of the approximations they provide are discussed in §5.2.

4.2 Weak Forms of the Continuum Maxwell's Equations

In this section the continuum weak forms of PDE I (3.22–3.25) and PDE II (3.26–3.29) are derived in preparation for spatial discretisation. We consider the computational volume $\Omega \subset \Re^3$ shown in Fig. 4.2, which is a modified version of Fig. 3.3, where the numbered Γ_n boundaries are replaced with three boundaries indicating three types to be considered, nl. Γ_D representing a Dirichlet boundary condition, Γ_N representing a Neumann boundary and Γ_M, an interior discontinuous material interface where the continuity conditions from §3.2 are to be enforced. Furthermore Γ_D is split such that $\Gamma_D = \Gamma_{D0} \cup \Gamma_{Dp}$ where Γ_{D0} is the homogeneous- (i.e. zero) and Γ_{Dp} the prescribed Dirichlet boundary conditions. Similarly the Neumann boundary is split into a homogeneous (Γ_{N0}) and prescribed (Γ_{Np}) part. The physical meaning of the various boundary conditions will become clear once the function spaces have been defined. Each boundary type may by made up of multiple disjoint sub-regions. A given problem configuration may con-

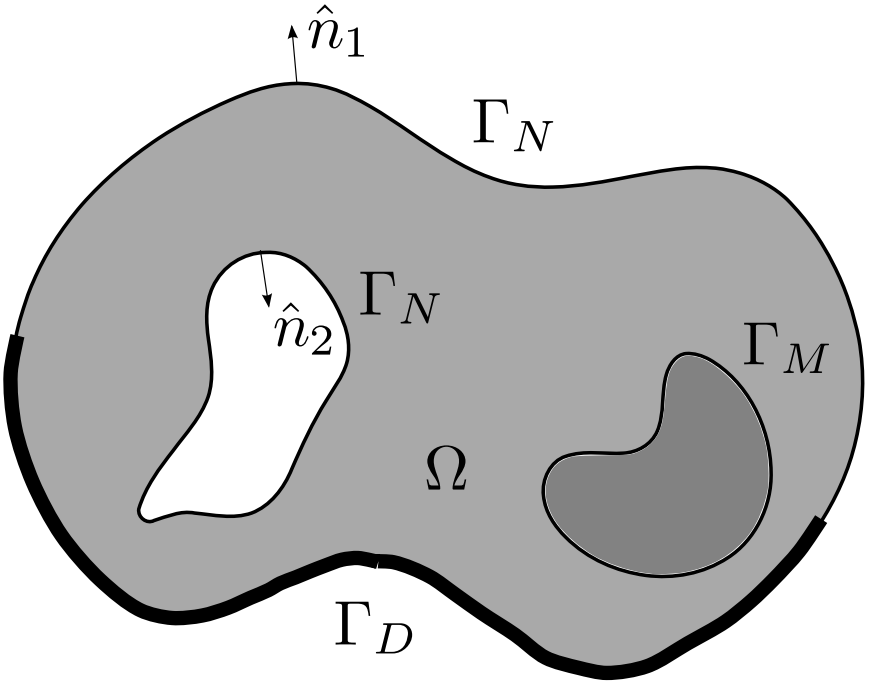


Figure 4.2: A generic inhomogeneous problem domain Ω showing the outward boundary normal orientation and interior boundaries of material discontinuity Γ_M. The Domain is bounded by a combination of Direchlet (Γ_D) and Neumann (Γ_N) boundaries.

tain any combination of these boundary conditions, so long as the exterior problem boundary is covered by some non-overlapping combination of Γ_D or Γ_N.

4.2.1 Inner Product Spaces for Weak Forms

Some inner spaces that will appear below are now defined. Since inner product spaces are a specialised subset of function spaces and all the function spaces dealt with in this text are inner product spaces, the two terms are used interchangeably. The most general space considered, is $(L^2(\Omega)^3)$, the space of vector functions square integrable in $\Omega \subset \Re^3$. This is a Lebesgue[4] space and also a Hilbert space. It is endowed with the inner product:

$$\langle \vec{u} \cdot \vec{v} \rangle = \langle \vec{u} \cdot \vec{v} \rangle_{(L^2(\Omega))^3} = \int_\Omega \vec{u} \cdot \vec{v}\, dV, \tag{4.16}$$

[4]Lebesgue function spaces ($L^p(\Omega)$) contain functions that are p-integrable on Ω, i.e. $\int_\Omega f^p < \infty$ implies $f \in L^p(\Omega)$. $(L^p(\Omega))^n)$ contains n-dimensional p-integrable vector functions. Only L^2 spaces are Hilbert spaces.

leading to the norm

$$\|u\| = \|u\|_{(L^2(\Omega))^3} = \sqrt{\langle \vec{u}, \vec{u} \rangle}. \tag{4.17}$$

The subscripts to $\rangle$ and $\|$ indicate the space that a particular inner product or norm are associated with. When the subscripts are dropped, it refers to the $(L^2(\Omega))^3$ norm. Now we can formally define[5]

$$(L^2(\Omega))^3 = \{\vec{u} : \|\vec{u}\|_{(L^2(\Omega))^3} < \infty\}. \tag{4.18}$$

This space is the basis of the weak formulation, and all the other spaces of vector valued functions considered subsequently are subsets of this space. It is a quite unrestrictive space; functions in $(L^2(\Omega))^3$ are not required to be differentiable, continuous, or even finite[6].

The Hilbert space $H(curl; \Omega)$ of square integrable vector functions with square integrable curl is defined as

$$H(curl; \Omega) = \{\vec{u} : \vec{u} \in (L^2(\Omega))^3 \mid \nabla \times \vec{u} \in (L^2(\Omega))^3\} \tag{4.19}$$

with the norm

$$\|u\|_{H(curl;\Omega)} = \sqrt{\langle \vec{u}, \vec{u} \rangle + \langle \nabla \times \vec{u}, \nabla \times \vec{u} \rangle} = \sqrt{\|\vec{u}\|^2 + \|\nabla \times \vec{u}\|^2}. \tag{4.20}$$

We also define the subset and subspace

$$H(curl; \Omega)_D = \{\vec{u} : \vec{u} \in H(curl; \Omega) \mid \hat{n} \times \vec{u} = \vec{u}_{bc} \text{ on } \Gamma_D\}, \tag{4.21}$$

$$H(curl; \Omega)_0 = \{\vec{u} : \vec{u} \in H(curl; \Omega) \mid \hat{n} \times \vec{u} = 0 \text{ on } \Gamma_D\}. \tag{4.22}$$

Eq. (4.21) is the space of functions restricted to satisfying the given tangential Dirichlet boundary conditions $\vec{u}_{bc}$ on Γ_D, of which a part may be homogeneous, i.e. $\vec{u}_{bc} = 0$ on Γ_{D0}. Eq. (4.22) satisfies the homogeneous condition on all of Γ_D.

The last function space that is needed is the Hilbert space $H(div; \Omega)$ of square integrable vector functions with square integrable divergence, defined as

$$H(div; \Omega) = \{\vec{u} : \vec{u} \in (L^2(\Omega))^3 \mid \nabla \cdot \vec{u} \in (L^2(\Omega))^3\} \tag{4.23}$$

with the norm

$$\|u\|_{H(div;\Omega)} = \sqrt{\langle \vec{u}, \vec{u} \rangle + \langle \nabla \cdot \vec{u}, \nabla \cdot \vec{u} \rangle} = \sqrt{\|\vec{u}\|^2 + \|\nabla \cdot \vec{u}\|^2}. \tag{4.24}$$

We also define the subspace[7]

$$H(div; \Omega)_{D0} = \{\vec{u} : \vec{u} \in H(div; \Omega) \mid \hat{n} \cdot \vec{u} = 0 \text{ on } \Gamma_{D0}\}. \tag{4.25}$$

[5]For those unfamiliar with the formal mathematical notation used to define the space, it is read as "the space $(L^2(\Omega))^3$ contains all the functions $\vec{u}$ such that the $(L^2(\Omega))^3$ norm of $\vec{u}$ is finite."

[6]As an example consider $\vec{f}(\vec{r}) = \frac{1}{|\vec{r}|}\hat{x}$, which is infinite in the limit $|\vec{r}| \to 0$ but is square integrable even in a domain that contains $|\vec{r}| = 0$.

[7]This definition is somewhat subtle. It constrains the normal component of $\vec{u}$ only on the *homogenous* part of the Direchlet boundary; on the prescribed part this space is unconstrained.

4.2.2 Inner Products on Traces

Trace innerproducts are defined by integrals of a dimension lower than that of Ω. They typically operate on the boundaries of the domain. For instance, if Ω is a 3-D volume, a trace inner product is defined on its bounding surface $\partial\Omega$. The trace inner product is written as

$$\langle \vec{u}, \vec{v} \rangle_t = \int_{\partial\Omega} \vec{u} \cdot \vec{v}\, dS, \tag{4.26}$$

where the t subscript indicates that it is a trace inner product.

4.2.3 PDE I in Weak Form

Writing PDE I (3.22–3.25) in operator form we get two coupled operators:

$$\mathcal{L}_a = \left\{ \frac{\partial \bar{\bar{\mu}}^{-1} \vec{B}}{\partial t} + \bar{\bar{\mu}}^{-1} \bar{\bar{\sigma}}_* \bar{\bar{\mu}}^{-1} \vec{B} = -\bar{\bar{\mu}}^{-1} \nabla \times \vec{E} \qquad \text{in } \Omega, \right. \tag{4.27}$$

and

$$\mathcal{L}_b = \begin{cases} \frac{\partial \bar{\bar{\epsilon}} \vec{E}}{\partial t} + \bar{\bar{\sigma}} \vec{E} = \nabla \times \bar{\bar{\mu}}^{-1} \vec{B} - \vec{J} & \text{in } \Omega \\ \hat{n} \times \vec{E} = \vec{E}_{bc} & \text{on } \Gamma_D. \end{cases} \tag{4.28}$$

The projective form of (4.27) using the testing function $\vec{\phi}$ is

$$\frac{\partial}{\partial t} \langle \bar{\bar{\mu}}^{-1} \vec{B}, \vec{\phi} \rangle + \langle \bar{\bar{\mu}}^{-1} \bar{\bar{\sigma}}_* \bar{\bar{\mu}}^{-1} \vec{B}, \vec{\phi} \rangle = -\langle \bar{\bar{\mu}}^{-1} \nabla \times \vec{E}, \vec{\phi} \rangle. \tag{4.29}$$

The projective form of (4.28) using the testing function $\vec{\psi}$ is

$$\frac{\partial}{\partial t} \langle \bar{\bar{\epsilon}} \vec{E}, \vec{\psi} \rangle + \langle \bar{\bar{\sigma}} \vec{E}, \vec{\psi} \rangle = \langle \nabla \times \bar{\bar{\mu}}^{-1} \vec{B}, \vec{\psi} \rangle - \langle \vec{J}, \vec{\psi} \rangle, \tag{4.30}$$

where the boundary condition $\vec{E}_{bc}$ must be satisfied by the choice of function space for $\vec{E}$. The conventional choice of function space for $\vec{B}$ and $\vec{E}$ is

$$\vec{E} \in H(curl; \Omega)_D, \qquad \vec{B} \in H(div; \Omega)_{D0}, \tag{4.31}$$

and for the testing functions:

$$\vec{\psi} \in H(curl; \Omega)_0, \qquad \vec{\phi} \in H(div; \Omega)_{D0}. \tag{4.32}$$

By choosing $\vec{E} \in H(curl; \Omega)$ and $\vec{B} \in H(div; \Omega)$ the correct boundary conditions at internal material discontinuities (Γ_M) are automatically guaranteed. The choice[8] of $H(div; \Omega)_{D0}$ for $\vec{B}$ is motivated by the fact that the normal component of $\vec{B}$ on Γ is fully determined by the time integral of the curl

[8]The distinction between D and $D0$ subscripts is somewhat subtle; see §4.2.1 from around (4.21).

of the tangential $\vec{E}$ component. On Γ_{D0} the tangential $\vec{E}$ component is identically zero, and therefore also the normal component of $\vec{B}$.

The specific subsets for $H(curl;\Omega)$ and $H(div;\Omega)$ chosen for the test and solution spaces of (4.30) warrant further discussion. The choice $\vec{E} \in H(curl;\Omega)_D$ enforces the tangential boundary condition of $\mathcal{L}_b$ as a Dirichlet condition. Note the difference between the solution space $\vec{E} \in H(curl;\Omega)_D$ and the testing space $\vec{E} \in H(curl;\Omega)_0$. The solution space satisfies all the Dirichlet conditions, including the prescribed ones; the testing space satisfies only *homogenous* Direchlet boundary conditions, but over the *whole* Direchlet boundary, including the parts with prescribed values. The choice $\vec{\psi} \in H(curl;\Omega)_0$ prevents the system from being over determined. This can be shown by considering spanning sets of the testing- and solution spaces. We can write the testing space as the spanning set:

$$\bigoplus_i \vec{\psi}_i = H(curl;\Omega)_0. \tag{4.33}$$

Since the spaces are linear and $H(curl;\Omega)_0$ makes no tangential contribution over Γ_D, we can write the solution space as:

$$\vec{\psi}_{bc} \oplus \bigoplus_i \vec{\psi}_i = H(curl;\Omega)_D \tag{4.34}$$

where $\vec{\psi}_{bc} \in H(curl;\Omega)$ is a single function that has a zero tangential component everywhere on Γ, except on Γ_{Dp} where $\hat{n} \times \vec{\psi}_{bc} = \vec{E}_{bc}$.

We know that the solution $\vec{E}$ can be represented by a weighted linear combination of $\vec{\psi}_i$ functions and $\vec{\psi}_{bc}$, but we also know that the weight of $\vec{\psi}_{bc}$ must be 1. The unknown number of degrees of freedom is therefore $\dim(H(curl;\Omega)_0)$. Each function $\vec{\psi}_i$ provides an equation to be satisfied through (4.30) with $\vec{\psi} = \vec{\psi}_i$, implying again $\dim(H(curl;\Omega)_0)$ equations in total. If a larger testing space is used, the problem would be over determined and would have to be solved as e.g. a least-squares problem. While the dimensions of the above spaces are infinite, they are countably so; hence the above argument still holds.

Examining (4.29), it is seen that the integrals implied by the inner products exist with the above choice of function spaces, since both $\vec{B}$ and $\nabla \times \vec{E}$ are indeed square integrable. We also note that by the exact sequence (3.34), $\nabla \times \vec{E} \in H(div;\Omega)$. In (4.30) $\nabla \times \vec{B}$ is required to be square integrable, which is not satisfied by the $H(div;\Omega)$ space. Using integration by parts[9] the resulting weak form is

$$\frac{\partial}{\partial t}\langle \bar{\bar{\epsilon}}\vec{E}, \vec{\psi}\rangle + \langle \bar{\bar{\sigma}}\vec{E}, \vec{\psi}\rangle = \langle \bar{\bar{\mu}}^{-1}\vec{B}, \nabla \times \vec{\psi}\rangle + \langle \hat{n} \times \bar{\bar{\mu}}^{-1}\vec{B}, \vec{\psi}\rangle_t - \langle \vec{J}, \vec{\psi}\rangle. \tag{4.35}$$

[9] A concise summary of the vector Green's identities needed and their proof i.t.o. the Divergence Theorem is in [32, §3.5.1 pp. 50].

The surface integral term falls away on Γ_D since $\hat{n} \times \vec{\psi} = 0$ on Γ_D. The surface term can be used[10] to enforce, in a projective weak sense, conditions on the tangential magnetic field ($\hat{n} \times \vec{H} = \hat{n} \times \overline{\overline{\mu}}^{-1} \vec{B}$) on Γ_N. Setting it to zero (i.e. on Γ_{N0}) implies a perfect magnetic conductor boundary. Since the original statement of PDE I calls only for Dirichlet boundaries, the surface term is omitted subsequently, hence:

$$\frac{\partial}{\partial t}\langle \overline{\overline{\epsilon}}\vec{E}, \vec{\psi}\rangle + \langle \overline{\overline{\sigma}}\vec{E}, \vec{\psi}\rangle = \langle \overline{\overline{\mu}}^{-1}\vec{B}, \nabla \times \vec{\psi}\rangle - \langle \vec{J}, \vec{\psi}\rangle. \tag{4.36}$$

4.2.4 PDE II in Weak Form

Writing PDE II (3.27–3.29) in operator form we get

$$\mathcal{L} = \begin{cases} \nabla \times \overline{\overline{\mu}}^{-1} \nabla \times \vec{E} + \sigma \frac{\partial \vec{E}}{\partial t} + \frac{\partial^2 \overline{\overline{\epsilon}} \vec{E}}{\partial t^2} = -\frac{\partial \vec{J}}{\partial t} & \text{in } \Omega \\ \hat{n} \times \vec{E} = \vec{E}_{bc} & \text{on } \Gamma_D. \end{cases} \tag{4.37}$$

Testing (4.37) with $\vec{\psi}$ leads to the projective form

$$\langle \nabla \times \overline{\overline{\mu}}^{-1} \nabla \times \vec{E}, \vec{\psi}\rangle + \langle \sigma \frac{\partial \vec{E}}{\partial t}, \vec{\psi}\rangle + \langle \frac{\partial^2 \overline{\overline{\epsilon}} \vec{E}}{\partial t^2}, \vec{\psi}\rangle = -\langle \frac{\partial \vec{J}}{\partial t}, \vec{\psi}\rangle. \tag{4.38}$$

The choice of spaces (along with their justification) for $\vec{E}$ and $\vec{\psi}$ are as for PDE I:

$$\vec{E} \in H(curl; \Omega)_D, \qquad \vec{\psi} \in H(curl; \Omega)_0. \tag{4.39}$$

The double-curl operator in (4.38) requires the weak form through integration by parts:

$$\langle \overline{\overline{\mu}}^{-1} \nabla \times \vec{E}, \nabla \times \vec{\psi}\rangle + \langle \hat{n} \times \overline{\overline{\mu}}^{-1} \nabla \times \vec{E}, \vec{\psi}\rangle_t + \langle \sigma \frac{\partial \vec{E}}{\partial t}, \vec{\psi}\rangle + \langle \frac{\partial^2 \overline{\overline{\epsilon}} \vec{E}}{\partial t^2}, \vec{\psi}\rangle = -\langle \frac{\partial \vec{J}}{\partial t}, \vec{\psi}\rangle. \tag{4.40}$$

The surface term allows the imposition of conditions on the tangential component of $\nabla \times \vec{E}$, which by (3.22), in the absence of magnetic conductivity, is the same as proscribing the tangential component of the magnetic field. As for PDE I, we can subsequently omit the surface term since no prescription will be made on the tangential magnetic field on Γ, yielding:

$$\langle \frac{\partial^2 \overline{\overline{\epsilon}} \vec{E}}{\partial t^2}, \vec{\psi}\rangle + \langle \sigma \frac{\partial \vec{E}}{\partial t}, \vec{\psi}\rangle + \langle \overline{\overline{\mu}}^{-1} \nabla \times \vec{E}, \nabla \times \vec{\psi}\rangle = -\langle \frac{\partial \vec{J}}{\partial t}, \vec{\psi}\rangle. \tag{4.41}$$

[10] In the post-processing context, the surface term can also be used to accurately evaluate certain integrals, e.g. for far-field recovery[43], [32, §13.6].

4.3 Finite-Element Semi-Discrete Maxwell's Equations

The continuum weak forms derived in §4.2 are defined in terms of the infinite dimensional spaces $H(curl;\Omega)$ and $H(div;\Omega)$. Obtaining a numeric solution requires the discrete approximation of the continuum spaces through finite-dimensional subspaces. The Galerkin method describes how to turn the continuum PDE problem into a matrix equation once the finite dimensional subspaces have been chosen, but does not indicate how they should be chosen. The finite element method (FEM) is primarily a method for the creation of suitable discrete subspaces.

The finite element method begins by modelling the problem geometry in terms of a tessellation, or mesh. The finite-dimensional subspaces are then defined in terms of the entities making up the mesh, leading to a spatially discrete system. In the case of frequency domain solutions, $\frac{\partial}{\partial t}$ is replaced with $-j\omega$, yielding a system that is ready to solve using matrix linear algebra techniques. In the time domain, the result is a matrix ordinary differential equation (ODE), called the semi-discrete system.

This section describes how to obtain the semi-discrete systems used throughout the rest of this text. The basic characteristics of a finite element mesh is described, followed by a discussion of the discrete functional spaces that are defined on the mesh. The characteristics of these spaces are critical to the accuracy and efficiency of the FEM. Finally the representation of the semi-discrete system as a matrix ODE is shown. Time discretisation is considered in the next section, and will not be discussed here. The concrete expression of the basis functions and geometric definition of the mesh elements will also be dealt with elsewhere; this, and other practical matters make up §5.

4.3.1 Finite Element Meshes

The first step in any finite element discretisation is to decompose the problem geometry into a number of regular geometric elements[11] of finite size. Mesh generation is a fairly complex topic in its own right, and is beyond the current scope. This section will start with the assumption that a valid mesh is available, and only describe its properties. Usually the elements are some sort of polyhedron. Popular choices for 3-D problems are tetrahedra and hexahedra, although several other types that present different trade offs have been used. Tetrahedral and hexahedral finite elements are discussed in some detail in §5.2, and pyramidal elements are also briefly touched upon in §8. Usually only a single mesh element type is utilised when solving a problem, but multiple types may be combined in a hybrid mesh. Constructing a valid

[11] The term "Finite Element Method" indeed derives from this step, and is attributed to Richard Courant. See the historical aside [2, pp. 289].

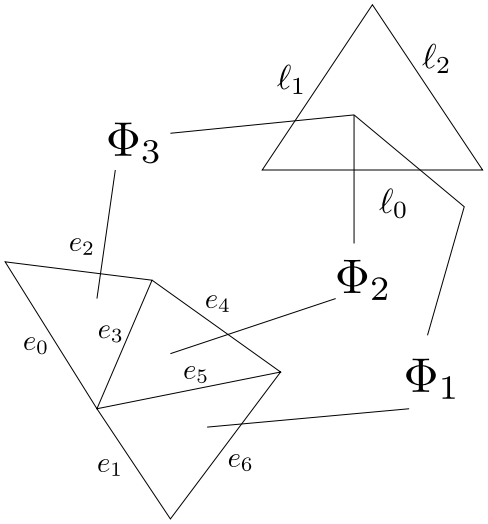


Figure 4.3: Triangular mesh showing how the parent element is transformed to gobal elements through affine transforms Φ_1, Φ_2 and Φ_3.

hybrid mesh involves some additional complexities; indeed, the construction of a valid hybrid mesh for the Maxwell's system is a major focus of this text, and is discussed in detail in §8

The tessellation, or mesh, of a given geometry Ω is denoted by the subscript h to distinguish it from the original domain when confusion may arise, i.e. Ω_h is a finite element tessellation of Ω. For the most part the subscript will be dropped. Assuming for now a single mesh element type, the mesh Ω_h is made up of n_{el} individual elements K_i. The mesh is defined as the union of all the elements:

$$\Omega_h = \bigcup_i^{n_{el}} K_i. \tag{4.42}$$

The K_i elements are defined in terms of a parent element $\hat{K}$. The definition of the parent element greatly simplifies the mathematical construction of the FEM, since the function spaces need only be defined on the parent element; the K_i elements are obtained by affine transforms Φ_i of $\hat{K}$. A triangular example is shown in Fig. 4.3.

Three dimensional meshes are subdivided into four groups of geometrical entities nl. volumes, faces, edges and nodes. Each entity type is numbered separately, i.e. volume i is in no particular way related to face i, edge i or node i. None of the sub-entities can exist in isolation, i.e. each edge, face or node must be connected to at least one element. The entities are defined below:

Volumes A three dimensional entity representing the interior of an element.

Element volumes may not overlap, i.e. if $\text{vol}(K_i)$ denotes the interior

volume of element K_i, $\text{vol}(K_i) \cap \text{vol}(K_j) = \emptyset \; \forall \; i \neq j$. The volume is defined by the nodes making up the element, and the number of nodes depends on the element shape. For instance, tetrahedral elements are defined by four nodes, and hexahedra by 8.

Faces Two dimensional entities that form the boundaries of elements, i.e. ∂K_i is defined by a fixed number of faces, depending on the element shape. For instance, tetrahedral elements are bounded by four triangular faces, and Cartesian hexahedra by six rectangles. Faces are connected either to two elements, or to a signle element and the boundary of Ω_h.

Edges Edges are line segments that form the boundaries of faces. Edges are found where two or more faces meet. Edges are always connected to exactly two nodes and are defined by them, irrespective of the mesh element type. In unstructured meshes, an edge can be connected to any number of faces or volumes. In a structured mesh there is a definite limit; for instance, in a structured hexahedral mesh an edge is connected to at most four elements.

Nodes Nodes represent the lowest level of the mesh. All the other entities are defined topologically in terms of nodes and their connection to other entities; nodes are the only entities defined in terms of spatial coordinates and define the physical dimensions of the geometry.

All the mesh entities are required to conform[12] to each other. Adjacent elements can only join at common faces, edges or nodes; no nodes, edges or faces can exist independently of their connection to higher dimensional entities. No entities are allowed to straddle or cross other entities and no empty space may exist between elements inside Ω. Additionally, no degenerate entities are allowed. For instance, a degenerate tetrahedron arises when the four nodes defining it are co-planar and it collapses into a triangle.

To ensure an accurate solution, the mesh is required to conform to the boundaries of the geometry. This implies that the exterior boundary Γ should be paved with mesh faces, and that no element or mesh entity may straddle Γ. Similarly, the mesh should conform to Γ_M, the internal boundary between material discontinuities, since the FEM specifically deals with material discontinuities on an element-by-element basis.

In practice, not all meshes are able to conform to all geometries. At one extreme, a regular Cartesian hexahedral mesh can only conform to boundaries directed along the Cartesian $\hat{x}$, $\hat{y}$ or $\hat{z}$ base vectors, and only when the

[12]The word "conform" has two distinct meanings frequently encountered when discussing finite elements. The one being discussed here describes the mesh geometry; the other has to do with the properties of the discrete function spaces used to approximate the field and is discussed in the next subsection.

boundaries are coincident with the regular grid. Unstructured tetrahedral meshes are much more flexible and are justly popular under FEM practitioners; they can conform to any polyhedral geometry and permit irregular refinement in regions of greater material complexity. Curved boundaries require curvilinear meshes for exact conformance, see e.g. [44], [45], [46] and [47]. Whenever a mesh does not conform exactly to a problem geometry, an additional geometrical error is incorporated into the semi-discrete system; this is discussed in more depth in §4.5.

4.3.2 Finite Element Function Spaces

Once an acceptable tessellation Ω_h of Ω has been created, discrete function spaces that are subspaces of the continuum spaces may be defined on Ω_h. As for the discrete geometry, the finite dimensional spaces will be denoted by a subscript h when confusion may exist, e.g. $H_h(curl;\Omega) \subset H(curl;\Omega)$, where $H_h(curl;\Omega)$ is the discrete space. For the most part the subscript will be dropped. This section outlines some general requirements for finite element subspaces.

The efficiency of the finite element method depends on the subspaces having spanning basis functions that have *compact support*. Compact support implies that each basis function is non-zero in only a small sub-domain of Ω. The implication is that each solution basis function only has a non-zero inner product interaction with a limited number of testing basis functions that have support in the same vicinity. This results in a large computational saving since, firstly only a small subset of the integrals implied by the inner products have to calculated, and secondly the resulting matrix equation is highly sparse. This sparsity is chiefly responsible for the good computational efficiency of the FEM.

Apart from compact support, the chosen subspaces need to ensure that accurate solutions are obtained. In other words, as the discrete subspace is enlarged, the solution should become progressively more accurate, and in the limit converge to the correct answer. The spaces must poses a spanning set that has degrees of freedom that are uniquely related to a single mesh entity. The degrees of freedom are usually defined in terms of linear functional over entities. As an example, if $\vec{w}_i$ is an "edge" basis function in a spanning set of some discrete space, i.e. $\vec{w}_i \in X_h$ that is used to approximate a function $\vec{f} \in X$, the weight e_i of $\vec{w}_i$ is $\int_{\ell_i} \vec{q}_i \cdot \vec{f}$, where ℓ_i is the edge entity that $\vec{w}_i$ is associated with. This leads to the degree of freedom definition[13] $\Xi_i(\vec{f}) = \int_{\ell_i} \vec{q}_i \cdot \vec{f}$.

The question of degrees of freedom warrants further discussion. As mentioned in §4.3.1, 3-D meshes have node, edge, face and volume entities.

[13] We are assuming a vector-valued function here as the basis functions needed for PDE I&II are vector-valued. The functional $\Xi_i(\vec{f}) = \int q_i f$ would be used for scalar basis functions.

Each degree of freedom must be associated with exactly one mesh entity; with the exception of nodes (which will not be discussed further since nodal basis functions will not be used subsequently), each entity can be associated with several degrees of freedom. The degrees of freedom related to an n-dimensional entity is defined by an n-dimensional functional. We denote ℓ_i, f_j and v_k as respectively the i'th edge, j'th face and k'th volume of Ω_h. Now we can denote the i'th basis function related to edge j'th as $\vec{w}_{i(\ell_j)}$ (with similar bracketed f and v subscripts for face and volume functions), and its associated degree of freedom as $\Xi_{i(\ell_j)}$. Now we can define

$$\Xi_{i(\ell_j)}(\vec{f}) = \int_{\ell_j} \vec{q}_{i(\ell_j)} \cdot \vec{f}\, d\ell, \tag{4.43}$$

$$\Xi_{i(f_j)}(\vec{f}) = \int_{f_j} \vec{q}_{i(f_j)} \cdot \vec{f}\, dS. \tag{4.44}$$

$$\Xi_{i(v_j)}(\vec{f}) = \int_{v_j} \vec{q}_{i(v_j)} \cdot \vec{f}\, dV. \tag{4.45}$$

The degrees of freedom are required to be bi-orthogonal to the basis functions:

$$\Xi_i(\vec{w}_j) = \delta_{ij}. \tag{4.46}$$

Note that here i and j run over all the basis function and degrees of freedom inclusive of edge, face and volume functions. The concrete degree of freedom functionals are seldom used explicitly in a FEM code, but their *existence* in the form described above allows the enforcement of global continuity conditions across elements. They also define the existence of an interpolation operator $\Pi(\vec{f})$. If a given continuum function $\vec{f}$ is to be interpolated by a subspace spanned by basis functions $\vec{w}_i$ with a vector of weights $\{e\}$ such that $\tilde{f} = \sum_i e_i \vec{w}_i \simeq \vec{f}$, then

$$\{e\} = \Pi(\vec{f}) \tag{4.47}$$

and

$$(\Pi(\vec{f}))_i = \Xi_i(\vec{f}). \tag{4.48}$$

A common use for an interpolation operator is the implementation of inhomogeneous Dirichlet boundary conditions; the known field is expanded in terms of a basis function expansion.

4.3.3 Discrete spaces for PDE I&II

This section discusses some of the requirements on finite element discrete spaces used for the discretisation of PDE I&II. The particular spaces that will be used are briefly motivated and some of their salient characteristics discussed.

Piecewise polynomial basis functions have been the mainstay of the FEM more or less for its entire history, and for good reason. Polynomials are well understood, easy to manipulate algebraically and have well studied convergence properties. Furthermore, finding compactly supported set spanning basis functions for piecewise polynomial spaces while satisfying the localised degree of freedom requirements mentioned previously is relatively straight forward.

One of the most prevalent FEM spaces is the piecewise-polynomial subspace of $H^1(\Omega)$, the space of once-differentiable functions in Ω. It is often referred to as "nodal finite elements", since the conformance of $H^1_h(\Omega)$ to $H^1(\Omega)$ is ensured by enforcing continuity at inter-element nodes[14]. Quoting Hiptmair [48] (emphasis mine):

> A finite element space is called *conforming*, if it is a proper subspace of the original continuous function space. This can be ensured through enforcing certain coupling conditions across inter-element boundaries via a *judicious choice of global degrees of freedom*.

For solving vector valued problems, the three dimensional space $(H^1_h(\Omega))^3$ is often used, where it is constructed by assigning an $H^1_h(\Omega)$ space to each of the three orthogonal spatial basis vectors. In the past, it was attempted to use this space to solve Maxwell's equations, but two significant problems surfaced. When solving eigen-problems: spurious modes that did not correspond to physical solutions arose, and for some driven problems the computed result would seem to converge towards an incorrect solution as the mesh was refined. The first problem can solved by noting that the problem statements in PDE I&II are essentially ungauged, i.e. no explicit statement is made about the divergence of $\vec{E}$ or $\vec{B}$. Adding a term to the weak form controlling the divergence (see e.g. [32, §7.4]) effectively removes the spurious modes, but the second problem remains. The second problem is due to the fact that solutions to Maxwell's equations in general[15] are in $H(curl;\Omega)$, not $(H^1(\Omega))^3$ [49], [32, §3.8, §7.4]. Clearly increasingly large subspaces $(H^1_h(\Omega))^3$ can never converge to a solution that lies outside of $(H^1(\Omega))^3$.

The introduction of suitable discrete $H(curl;\Omega)$- and $H(div;\Omega)$ conforming spaces by Nédélec [10], [11] alleviated this difficulty. These spaces

[14] For polynomial bases of order higher than linear this is not strictly true, since edge, face and volume degrees of freedom become necessary in turn as the order is increased. However, interpolatory polynomial bases are often used for $H^1(\Omega)$ elements, resulting in degrees of freedom that can be represented as the value of the field at nodes defined on edges or faces, or interior to the elemental volume. This is discussed in more detail in light of the interpolatory hexahedral elements in §5.2.3.

[15] Many EM problems do have smooth solutions that are in $(H^1(\Omega))^3$, such as the resonant modes of a PEC cavity without re-entrant corners.

are often called respectively curl- and divergence conforming[16] elements. Strictly speaking, the Nédélec spaces do not eliminate the spurious modes from eigen-solutions; rather they model the null space of the curl-curl operator well enough such that they have numerically zero eigenvalues, and are easy to discard automatically. They do not affect driven solutions either, since any legitimate source condition would not excite them[17]. Removing the spurious modes entirely is possible and has certain benefits such as allowing static solutions (see e.g. [50]), but will not be considered here. It was later realised that the Nédélec elements are in fact a special case of discrete differential forms [48], [51], [52], where the $H(curl;\Omega)$ elements are recognised as discrete 1-forms and the $H(div;\Omega)$ as discrete 2-forms.

The salient feature of Nédélec elements is their inter-element continuity conditions. The 1-form elements have degrees of freedom defined in terms of their components *tangential* to an inter-element edge or face. The lowest polynomial order discrete 1-form bases have only edge degrees of freedom, hence discrete 1-forms are often referred to as *edge elements* or *tangential elements*. Higher order bases include face and volume degrees of freedom. Similarly, the 2-form degrees of freedom are defined in terms of components *normal* to inter-element faces, with the lowest order bases having only face degrees of freedom, and are frequently referred to as *face* or *facet elements*. In [32, lemma 5.3] it is proven that curl- or divergence conformance results from respectively tangential or normal continuity. The tangentially defined degrees of freedom are also required to satisfy the tangential Dirichlet boundary conditions on $\vec{E}$ implied by $H(curl;\Omega)_D$.

Another useful feature is the distinction between mixed- [10] and full [11] order bases. For instance, a mixed first order 1-form base is complete only to zero'th order, but contains some linear terms. This is motivated by the fact that the field and its curl are equally important to the accuracy of a Maxwell's solution. By removing the part of the linear polynomial space that are gradients, i.e. have zero curl, the optimal rate of convergence for a given computational effort is obtained. The full order bases include the full polynomial space, and are advantageous for certain problems [53]. The concrete basis sets used for tetrahedral and hexahedral elements are described in §5.2.

If Nédélec type discrete spaces are used, the curl of a discrete 1-form can be represented exactly by a discrete 2-form of sufficient order defined on the same mesh. If $\{a\}$ is a vector of 1-form degrees of freedom, and $\{b\}$ a vector of 2-form degrees of freedom, the curl of the 1-form can be represented by

[16] Strictly speaking, $(H^1(\Omega))^3$ is both div- and curl conforming, since it has both square integrable curl and divergence. What differentiates the Nédélec discrete spaces is that they are convergent to any function in their respective continuum spaces.

[17] Although erroneously exciting them with a malformed source can wreak havoc in the time domain.

the 2-form as

$$\{b\} = [C]\{e\}. \tag{4.49}$$

This matrix is independent of metric. In other words, the physical dimensions of a mesh play no role in determining the entries of $[C]$, only the mesh topology. The construction of $[C]$ for concrete basis sets is discussed in §5.2.4.

We will now introduce the notation subsequently used for basis functions. We denote the i'th basis function of a set spanning a discrete 1-form, i.e. $H_h(curl;\Omega)$, space as $\vec{w}_i^{(1)}$. Similarly, a discrete 2-form, i.e. $H_h(div;\Omega)$, basis function as $\vec{w}_j^{(2)}$. A whole basis set is written in vector form, e.g. $\{\vec{w}^{(1)}\}$ is the vector of all the $\vec{w}_i^{(1)}$ functions making up a 1-form basis. If the need arises to distinguish between different sets of basis functions in the same context, such as when dealing with prescribed inhomogeneous Dirichlet boundary conditions, we may write $\vec{w}_{fi}^{(1)}$ for the i'th free basis function and $\vec{w}_{pj}^{(1)}$ for the j'th prescribed basis function. A discretised function is represented as a weighted sum of the discrete basis functions. For instance, a discrete $\vec{E}$ representation is written as:

$$\vec{E}_h = \sum_i^{n_e} e_i \vec{w}_i^{(1)} = \{\vec{w}^{(1)}\}^T\{e\}. \tag{4.50}$$

4.3.4 Semi-discrete PDE I

Given discrete approximations of the spaces (4.31) and (4.32) their basis functions, we can write

$$\vec{E}_h \in H_h(curl;\Omega)_D, \qquad \vec{E}_h = \sum_i^{n_{eD}} e_i \vec{w}_{Di}^{(1)} = \{\vec{w_D}^{(1)}\}^T\{e\}, \tag{4.51}$$

$$\vec{B}_h \in H_h(div;\Omega)_{D0}, \qquad \vec{B}_h = \sum_i^{n_b} b_i \vec{w}_i^{(2)} = \{\vec{w}^{(2)}\}^T\{b\}. \tag{4.52}$$

Furthermore, we have the $\vec{E}$ testing functions

$$\vec{w}_i^{(1)} \in H(curl;\Omega)_0, \qquad i = 1 \ldots n_e, \tag{4.53}$$

and $\vec{B}$ testing functions

$$\vec{w}_i^{(2)} \in H(div;\Omega)_{D0}. \tag{4.54}$$

Substituting the discretised fields and testing functions into (4.29) and (4.35) and moving all the terms except those with time derivatives to the RHS, we

get:

$$\frac{\partial}{\partial t}\langle\bar{\bar{\mu}}^{-1}\vec{B}_h, \vec{w}_i^{(2)}\rangle = -\langle\bar{\bar{\mu}}^{-1}\nabla\times\vec{E}_h, \vec{w}_i^{(2)}\rangle - \langle\bar{\bar{\mu}}^{-1}\bar{\bar{\sigma}}_*\bar{\bar{\mu}}^{-1}\vec{B}_h, \vec{w}_i^{(2)}\rangle \quad \text{for } i = 1\ldots n_b, \quad (4.55)$$

$$\frac{\partial}{\partial t}\langle\bar{\bar{\epsilon}}\vec{E}_h, \vec{w}_i^{(1)}\rangle = \langle\bar{\bar{\mu}}^{-1}\vec{B}, \nabla\times\vec{w}_i^{(1)}\rangle - \langle\bar{\bar{\sigma}}\vec{E}, \vec{w}_i^{(1)}\rangle - \langle\vec{J}, \vec{w}_i^{(1)}\rangle, \quad \text{for } i = 1\ldots n_e. \quad (4.56)$$

We can now write (4.55) and (4.56) as matrix equations, with each testing function contributing one row to the matrix system:

$$\frac{d}{dt}[M_{\mu^{-1}}]\{b\} = -[P_{\mu^{-1}}]^T\{e\} - [\sigma_*]\{b\}, \quad (4.57)$$

$$\frac{d}{dt}[M_\epsilon]\{e\} = [P_{\mu^{-1}}]\{b\} - [\sigma]\{e\} - \{f_I\}, \quad (4.58)$$

with matrix and vector entries

$$[M_\epsilon]_{ij} = \int_\Omega \vec{w}_i^{(1)}\cdot\bar{\bar{\epsilon}}\vec{w}_{Dj}^{(1)}\,dV, \quad (4.59)$$

$$[M_{\mu^{-1}}]_{ij} = \int_\Omega \vec{w}_i^{(2)}\cdot\bar{\bar{\mu}}^{-1}\vec{w}_j^{(2)}\,dV, \quad (4.60)$$

$$[\sigma_*]_{ij} = \int_\Omega \vec{w}_i^{(2)}\cdot\bar{\bar{\mu}}^{-1}\bar{\bar{\sigma}}_*\bar{\bar{\mu}}^{-1}\vec{w}_j^{(2)}\,dV, \quad (4.61)$$

$$[\sigma]_{ij} = \int_\Omega \vec{w}_i^{(1)}\cdot\bar{\bar{\sigma}}\vec{w}_{Dj}^{(1)}\,dV, \quad (4.62)$$

$$[P_{\mu^{-1}}]_{ij} = \int_\Omega \nabla\times\vec{w}_{Di}^{(1)}\cdot\bar{\bar{\mu}}^{-1}\vec{w}_j^{(2)}\,dV, \quad (4.63)$$

$$\{f_I\}_i = \int_\Omega \vec{J}\cdot\vec{w}_i^{(1)}\,dV. \quad (4.64)$$

Since curl can be represented exactly by the discrete 2-forms, we can rewrite (4.57) using (4.49) as

$$\frac{d}{dt}\{b\} = -[C]\{e\} - [M_{\mu^{-1}}]^{-1}[\sigma_*]\{b\}, \quad (4.65)$$

with[18]

$$[P_{\mu_{-1}}]^T = [M_{\mu^{-1}}][C]. \quad (4.66)$$

[18]This relation holds exactly, since it is known that the discrete 2-form can exactly represent the curl of the discrete 1-form; hence the projective solution implied is exact.

Using (4.66) and the fact that $[M_{\mu^{-1}}]$ is symmetric, we can write

$$[P_{\mu_{-1}}] = [C]^T [M_{\mu^{-1}}]. \tag{4.67}$$

Now (4.58) can be rewritten as

$$\frac{d}{dt}[M_\epsilon]\{e\} = [C]^T [M_{\mu^{-1}}]\{b\} - [\sigma]\{e\} - \{f_I\}. \tag{4.68}$$

Because of the provision for inhomogeneous Dirichlet boundary conditions, (4.68) as written above is not quite ready to be solved. Following the continuum case discussion around (4.34) and noting that

$$H_h(curl;\Omega)_0 \subset H_h(curl;\Omega)_D, \tag{4.69}$$

can split the representation of $\vec{E}_h$ into a free (i.e. unknown) and prescribed part, such that

$$\{e\} = \{e_f\} \oplus \{e_p\}, \tag{4.70}$$

and define the space of the prescribed field and basis functions

$$\bigoplus_i^{n_{ep}} \vec{w_{pi}}^{(1)} = H(curl;\Omega)_p = H(curl;\Omega)_D \setminus H(curl;\Omega)_0. \tag{4.71}$$

The prescribed degrees of freedom $\{e_p\}$ are chosen to interpolate $\vec{E}_{bc}$, i.e.

$$\hat{n} \times \{\vec{w_p}^{(1)}\}^T \{e_p\} \simeq \vec{E}_{bc} \text{ on } \Gamma_D. \tag{4.72}$$

This implies that the matrix on the LHS of (4.68) is rectangular, and that not all of $\{e\}$ is free. By globally numbering free degrees of freedom before prescribed, we can define block matrices

$$[M_\epsilon] = [\ [M_{\epsilon(ff)}]\ \ [M_{\epsilon(fp)}]\ ], \tag{4.73}$$

$$[C_\epsilon] = [\ [C_{(f)}]\ \ [C_{(p)}]\ ], \tag{4.74}$$

and $\{e_p\}$ can be taken to the RHS of (4.68):

$$\frac{d}{dt}[M_{\epsilon(ff)}]\{e_f\} = -\frac{d}{dt}[M_{\epsilon(fp)}]\{e_p\} + [C_f]^T [M_{\mu^{-1}}]\{b\} - [\sigma]\{e\} - \{f_I\}. \tag{4.75}$$

Note that $\{e_p\}$ is not affected by the magnetic field, implying that a Dirichlet boundary condition is a zero-impedance "hard source." Finally we write (4.65) with the same matrix structure for completeness:

$$\frac{d}{dt}\{b\} = -[C_f]\{e_f\} - [C_p]\{e_p\} - [M_{\mu^{-1}}]^{-1}[\sigma_*]\{b\}. \tag{4.76}$$

For the most part, the free/prescribed structure will subsequently not be explicitly stated since confusion should not occur.

4.3.5 Semi-discrete PDE II

The process for discretising PDE II is largely similar to the development presented for PDE I in the previous subsection; in light of this, the development from the previous section will be assumed. We start by discretising $\vec{E}$ as in (4.51) and the corresponding testing space as in (4.53). Substituting the discrete representations in (4.41) and writing it as a matrix equation,

$$\frac{d^2}{dt^2}[M_\epsilon]\{e\} + \frac{d}{dt}[M_\sigma]\{e\} = -[S]\{e\} - \{f_{II}\}. \tag{4.77}$$

The $[M_\epsilon]$ and $[M_\sigma]$ matrix entries are as defined in (4.59) and (4.62) respectively. Furthermore,

$$[S_{ij}] = \int_\Omega \nabla \times \vec{w}_i^{(1)} \cdot \bar{\bar{\mu}}^{-1} \nabla \times \vec{w}_{Dj}^{(1)}\, dV, \tag{4.78}$$

$$\{f_{II}\}_i = \int_\Omega \frac{\partial}{\partial t}\vec{J} \cdot \vec{w}_i^{(1)}\, dV. \tag{4.79}$$

Note that barring the time derivative, the entries of $\{f_I\}$ and $\{f_{II}\}$ are identical. As for PDE I, prescribed Dirichlet boundary conditions can be handled by moving the matrix blocks involving prescribed degrees of freedom to the RHS:

$$\frac{d^2}{dt^2}[M_\epsilon]\{e\} + \frac{d}{dt}[M_\sigma]\{e\} = -[S]\{e\} - \{f_{II}\}. \tag{4.80}$$

$$\begin{aligned}\frac{d^2}{dt^2}[M_{\epsilon(ff)}]\{e_f\} + \frac{d}{dt}[M_{\sigma(ff)}]\{e_f\} = &-\frac{d^2}{dt^2}[M_{\epsilon(fp)}]\{e_p\} - \frac{d}{dt}[M_{\sigma(fp)}]\{e_p\} \\ &- [S_{(ff)}]\{e_f\} - [S_{(fp)}]\{e_p\} \\ &- \{f_{II}\}.\end{aligned} \tag{4.81}$$

4.3.6 Discrete differential forms interpretation

The treatment presented in §4.3.4 and §4.3.5 can be seen as the classic approach. The same numerical equations can equally well be derived by following the discrete differential forms route. While the result may be identical, it is worth considering the discrete differential forms route since it provides additional insight into the structure of the finite elements solution. It also permits a tractable interpretation of the convolution free PML presented in §6.3.

Discrete forms will be represented notationally simply as their degree of freedom vectors, e.g.

$$\{e\} = \Pi_1(E), \tag{4.82}$$

where E is the continuum 1-form dual to the vector $\vec{E}$ field, $\{e\}$ represents the discrete 1-form approximation and $\Pi_1(f)$ the interpolation operator. While the derivation is different, the Nédélec elements used in the previous subsections are equivalent to discrete forms and have identical degrees of freedom. To cast the continuum forms of PDE I (§3.5.3 and PDE II (§3.5.4) in discrete form, discrete equivalents are needed for the material Hodge operators, and for the exterior derivative. The boundary conditions (which would involve the trace operator) and conductivities[19] are neglected for simplicity's sake.

The discrete exterior derivative (i.e. curl) of $\{e\}$ is simply the curl matrix (4.49):

$$[d_1] = [C]. \tag{4.83}$$

This operation is exact; no approximation is made, even in the discrete case. It depends only on the topology of the mesh and is thereby metric free. As noted in §4.3.3, the $[C]$ matrix for low-order discrete forms contains only ± 1 and 0 entries. The operator used to calculate the curl of the discrete magnetic field requires the discrete Hodge operator that is discussed next.

Unlike the discrete curl operator, the discrete Hodge operator permits various implementations. Considering for now only the Galerkin Hodge, the discrete material Hodge operators are simply the mass matrices:

$$[\star_\epsilon] = [M_\epsilon], \tag{4.84}$$

$$[\star_{\mu^{-1}}] = [M_{\mu^{-1}}]. \tag{4.85}$$

While the expressions are simple, their interpretation is a little more subtle. The discrete operation $\{d\} = [\star_\epsilon]\{e\}$ equivalent to the continuum $D = \star_\epsilon E$ describes an isomorphism between $\{e\}$ and $\{d\}$; while the interpretation of $\{e\}$ is quite clear in terms of the discrete 1-form basis functions and degrees of freedom, the meaning of $\{d\}$ is uncertain. The same question arises when considering the discrete magnetic Hodge $\{h\} = [\star_{\mu^{-1}}]\{b\}$.

We will consider as an example discrete $\{e\}$ and $\{b\}$ of the lowest (i.e. mixed first) order. The 1-form $\{e\}$ has only edge basis functions and degrees of freedom associated with edges of the mesh. Similarly, the 2-form $\{b\}$ has only face basis functions and degrees of freedom. An initially plausible interpretation[20] of $\{d\} = [\star_\epsilon]\{e\}$ is that since the continuum $\star_\epsilon$ operator maps a 1-form to a 2-form, $\{d\}$ is a discrete 2-form of the same type as $\{b\}$. The

[19] The conductivities could of course be incorporated in a frequency-dependent material Hodge operator. This in fact is the approach that will be taken for the PML implementation in §6.3.

[20] Numerical formulations based on this interpretation have been proposed in the past. The current author has investigated such a formulation in the process of the current work. It is discussed in [24].

dimension of $\{e\}$ and $\{d\}$ must be the same due to their isomorphism. However, the dimension of $\{e\}$ is the number of mesh edges, while the dimension of $\{b\}$ is the number of mesh faces; they are unlikely to be the same.

The correct interpretation involves the barycentric dual lattice. The $\{e\}$ and $\{b\}$ forms are defined in terms of the primary mesh of the geometry. The dual lattice if formed by connecting all the barycenters of the primary mesh elements together. Hexahedra are special in that their dual lattice is also made up of hexahedra; other mesh types have more complicated duals. The significance of the dual mesh is that it has the same number of faces as the primary mesh has edges and vice versa. Now $\{d\}$ can be defined in terms of face degrees of freedom on the dual mesh and $\{h\}$ by degrees of edge freedom on the dual mesh [54], [55], [56], [57], [58], [59]. These dual-mesh quantities are sometimes referred to as *twisted* differential forms. While explicit expressions for the twisted forms are not generally available, one may implicitly formulate a numeric solution in terms of them through the Hodge operator matrices [60]. Note also that, unlike the continuum case where $\star^{-1} = \star$, the matrices defined in (4.84) and (4.85) have to be inverted to obtain the inverse Hodge, even in the case of unit material parameters.

Now we are in position to consider the curl of the magnetic field as required by (4.75). Since $\{b\}$ is a 2-form, applying the exterior derivative operator to it is equivalent to divergence. Since we want to take the curl, it is converted to a twisted 1-form through $[\star_{\mu^{-1}}]$, and then the exterior derivative operator that takes its curl is

$$[d_1^\dagger] = [C]^T, \tag{4.86}$$

where the dagger indicates that the matrix operates on a twisted form.

By substituting the discrete differential for operators into continuum differential form expressions of PDE I&II, semi-discrete systems may be obtained. For PDE I (3.51), (3.52), omitting conductivity, we obtain

$$\frac{d}{dt}\{b\} = -[C]\{e\} \tag{4.87}$$

$$\frac{d}{dt}[\star_\epsilon]\{e\} = [C]^T[\star_{\mu_{-1}}]\{b\} - \{j\}, \tag{4.88}$$

which is identical to (4.65) and (4.68) except for $\{j\}$ which should clearly be equivalent to $\{f_I\}$. In the discrete differential forms interpretation $\{j\}$ represents the twisted 2-form interpolant of the continuum current density J. This suggests that (4.64) represents an approximation to the twisted 2-form interpolation operator $\Pi_2^\dagger(f)$. The current author is not aware of any specific discussion regarding the interpolation of J in the current literature; in any case, this interpretation is certainly one that is known to provide accurate computational results!

Considering PDE II (3.55), again neglecting conductivity, we obtain

$$[C]^T[\star_{\mu^{-1}}][C]\{e\} + \frac{\partial^2}{\partial t}[\star_\epsilon]\{e\} = \frac{d}{dt}\{j\}. \tag{4.89}$$

This form can be shown to be equivalent to (4.77) by noting that [59]

$$[S] = [C]^T[\star_{\mu_{-1}}][C]. \tag{4.90}$$

The separation of the topological (exterior derivative) and metric (Hodge) operations by the discrete differential forms approach is notable for clarifying the nature of the approximation made in the discrete case. The topological operations are exact, implying that the only source of error is introduced by the Hodge operations. From the discrete perspective, $[\star_\epsilon]$ and $[\star_{\mu_{-1}}]$ are given properties much like ϵ and μ in the continuum case. The analyst's job is to find $[\star_\epsilon]$ and $[\star_{\mu_{-1}}]$ operators that approximate the continuum case accurately. The Galerkin Hodge presented here is a particular choice that recovers that Galerkin FEM approximations. Other choices of discrete Hodge, leading to different computational trade offs are possible. Some references discussing the implementation of Hodge operators are [54], [61], [62], [60].

4.4 Fully Discrete Systems

The systems presented in §4.3 are discrete in space, but not in time. A fully discrete frequency domain phasor formulation can be obtained by replacing the $\frac{d}{dt}$ operations with the phasor domain equivalent $-j\omega$. Obtaining a fully discrete time domain formulation requires discrete approximations to $\frac{d}{dt}$ and $\frac{\partial^2}{\partial t}$ to be found. Time discretisation methods are often referred to as *time integration methods*; the terms will subsequently be used interchangeably. The coupled first order equations of PDE I and the single second order equation of PDE II require separate time discretisation approaches although it will subsequently be shown in §7 that under certain circumstances the fully discrete PDE I&II can be made exactly equivalent; indeed, that is the basis of the implicit-explicit hybrid formulation.

Time discretisation involves "stepping"[21] the discrete system by constant discrete time increments[22] Δt. At each time step the current system state, the values of the driving function and potentially a period of the system history is taken into account to calculate the next system state Δt seconds in the future. In this section two commonly used time discretisation schemes are presented for respectively PDE I and PDE II, viz. leapfrog central differencing and the Newmark-β method. Before the methods are presented, the

[21] Sometimes referred to as the system "marching on in time"

[22] Methods with variable time increments exist but are usually reserved for non-linear problems.

properties required by Maxwell's equations in a time discretisation scheme and their implications are considered. The leapfrog method is applied to derive a fully discretise PDE I, and some of its properties discussed. The same process is then repeated for Newmark-β method and PDE II. The eigensolution of the two semi-discrete systems are derived in aid of the stability analysis which concludes this section.

4.4.1 Time Discretisation and Maxwell's Equations

Some of the properties that a time discretisation method should have to ensure the accurate solution of Maxwell's equations are listed below and will be discussed in more detail presently. The time discretisation method should be:

Conservative Conservative methods do not introduce any extra numeric dissipation beyond that which is present in the original problem.

Stable The solution to a passive problem exited with finite energy should remain finite, and solutions of lossy systems should eventually decay; solutions that grow without bound in time are unstable.

Consistent Any discrete method will introduce some error in time integration; as the time increment $\Delta t \to 0$, so should the error.

Convergent Convergence is related to consistency, but a consistent method can be unstable for some values of Δt. Convergence is implied by simultaneous consistency and stability.

Time integration schemes almost always require a value of Δt that is much smaller than required to resolve the highest frequency of practical interest to be stable and/or accurate. Many common schemes are specifically designed to damp high frequency modes[23]and are therefore not conservative. Of course they are designed to have very little damping in the frequency range of interest, but there is always some damping. Since practical electromagnetics problems are frequently lossless, or nearly so, and often require relatively long simulation periods, schemes that have numerical damping can lead to unacceptable errors [63].

The importance of stability is obvious; a solution that grows without bound will have an error without bound. Stability can be conditional or

[23]This is largely due to the historically greater use of the FEM for transient analysis in mechanics. The numerical damping removes oscillations due to high order modes that are in any case poorly discretised, leading to solutions that tend to the steady state more quickly. In principle these poorly discretised high order modes are undesirable in EM solutions too, but they are easier to tolerate than non-physical damping. By exciting EM systems with band-limited pulses the high order modes should never be excited, and in any case EM results are almost always post processed with the FFT where the high frequency components are trivially removed by the simple expedient of not considering them.

unconditional. Conditional stability implies that Δt must be smaller than some upper bound that depends on the particular semi-discretisation, or the solution will grow without bound, usually exponentially. Unconditionally stable schemes are stable independent of Δt. While it would seem preferable to always use an unconditionally stable method, they are with very few exceptions[24] *implicit*.

Time integration schemes can be *explicit* or *implicit*. The next value calculated by an explicit method depends only on the current and historical state. An implicit method includes a dependency on the next state while it is being calculated. In practice this implies that an implicit method will *always* require a matrix system to be solved at every time-step, while an explicit method can avoid a matrix solve when applied to a suitable semi-discrete system.

The overall accuracy of a time integration method is in general closely linked to its convergence properties. Generally, the time integration error behaves as $O(\Delta t^p)$, where p is the order of convergence.

Both methods subsequently considered are second order accurate in time, provably stable and conservative. The leapfrog method is conditionally stable and explicit, while the Newmark-β method can be either explicit and conditionally stable, implicit and conditionally stable or implicit and unconditionally stable.

Time discretisation will be performed at time-steps Δt apart. A vector function of time $\{x\}(t)$ is evaluated at discrete times by the substitution $t = n\Delta t$ where n is an integer. Now we introduce the notation.

$$\{x\}^n = \{x\}(n\Delta t). \tag{4.91}$$

Values half way between two time-steps are sometimes require and are written simply as

$$\{x\}^{n\pm\frac{1}{2}} = \{x\}((n \pm \tfrac{1}{2})\Delta t). \tag{4.92}$$

The discrete approximation of time derivatives are using the dot notation

$$\{\dot{x}\}^n \simeq \frac{d}{dt}\{x\}(n\Delta t) \tag{4.93}$$

in order to prevent confusion with the continuous time case.

4.4.2 Fully Discrete PDE I

The leapfrog integration is used to fully discretise PDE I. The leapfrog method starts out by discretising quantities and their derivatives half a

[24] The unconditionally stable explicit methods that do exist generally suffer from bad accuracy with values of Δt larger than the stability limit [64]. They may still be useful in cases where there are only a very small number of stiff elements.

time-step apart. Consider a coupled system of 2 variables x_1 and x_2. They are discretised half a time-step apart, such that

$$\begin{aligned} x_1^{n+\frac{1}{2}} &\simeq x_1((n+\tfrac{1}{2})\Delta t) \\ x_2^n &\simeq x_2(n\Delta t). \end{aligned} \tag{4.94}$$

Furthermore their time derivatives are discretised half a time-step removed from their values:

$$\begin{aligned} \dot{x}_1^n &\simeq \frac{d}{dt}x_1(n\Delta t) \\ \dot{x}_2^{n+\frac{1}{2}} &\simeq \frac{d}{dt}x_2((n+\tfrac{1}{2})\Delta t). \end{aligned} \tag{4.95}$$

The leapfrog update equations are

$$\begin{aligned} x_1^{n+\frac{1}{2}} &= x_1^{n-\frac{1}{2}} + \Delta t \dot{x}_1^n \\ x_2^{n+1} &= x_2^n + \Delta t \dot{x}_2^{n+\frac{1}{2}}. \end{aligned} \tag{4.96}$$

The leapfrog method is a central difference method, since the value of velocity (i.e. time derivative) used to update a variable is located in time midway between the current value and the next value to be computed.

By identifying from (4.75), (4.76) (ignoring the free/prescribed split)

$$\begin{aligned} x_1 &= \{b\} & \dot{x}_1 &= -[C]\{e\} - [M_{\mu^{-1}}]^{-1}[\sigma_*]\{b\} \\ x_2 &= \{e\} & \dot{x}_2 &= [M_\epsilon]^{-1}\left([C]^T[M_{\mu^{-1}}]\{b\} - [\sigma]\{e\}\right). \end{aligned} \tag{4.97}$$

The fully discrete update equations can now be written by substituting (4.4.2) into (4.96):

$$\{b\}^{n+\frac{1}{2}} = \{b\}^{n-\frac{1}{2}} - \Delta t\left([C]\{e\}^n + [M_{\mu^{-1}}]^{-1}[\sigma_*]\{b\}^n\right), \tag{4.98}$$

$$\{e\}^{n+1} = \{e\}^n + \Delta t[M_\epsilon]^{-1}\left([C]^T[M_{\mu^{-1}}]\{b\}^{n+\frac{1}{2}} - [\sigma]\{e\}^{n+\frac{1}{2}}\right). \tag{4.99}$$

One difficulty presented by (4.98) and (4.99) is the requirement for $\{b\}^n$ and $\{e\}^{n+\frac{1}{2}}$ when respectively magnetic and electric conductive losses are included. Since discrete representations are not available at those time-steps, they need to be approximated in some way. A straight forward choice is $\{b\}^n \simeq \{b\}^{n-\frac{1}{2}}$ and $\{e\}^{n+\frac{1}{2}} \simeq \{e\}^n$. While this approximation can be used, it is known to reduce the stability of the system. Using the *semi-implicit* [6, eqn. (3.26)] approximations

$$\begin{aligned} \{b\}^n &\simeq \frac{1}{2}(\{b\}^{n-\frac{1}{2}} + \{b\}^{n+\frac{1}{2}}) \\ \{e\}^{n+\frac{1}{2}} &\simeq \frac{1}{2}(\{e\}^n + \{e\}^{n+1}), \end{aligned} \tag{4.100}$$

the update equations are

$$\{b\}^{n+\frac{1}{2}} = \left([M_{\mu^{-1}}] + \frac{\Delta t}{2}[\sigma_*]\right)^{-1} \left[\left([M_{\mu^{-1}}] - \frac{\Delta t}{2}[\sigma_*]\right)\{b\}^{n-\frac{1}{2}} - \Delta t[M_{\mu^{-1}}][C]\{e\}^n\right), \quad (4.101)$$

$$\{e\}^{n+1} = \left([M_\epsilon] + \frac{\Delta t}{2}[\sigma]\right)^{-1} \left[\left([M_\epsilon] - \frac{\Delta t}{2}[\sigma]\right)\{e\}^{n-1} + \Delta t[C]^T[M_{\mu^{-1}}]\{b\}^{n+\frac{1}{2}}\right]. \quad (4.102)$$

Note that in the absence of magnetic conductivity[25], (4.101) is fully explicit:

$$\{b\}^{n+\frac{1}{2}} = \{b\}^{n-\frac{1}{2}} - \Delta t[C]\{e\}^n. \quad (4.103)$$

If a semi-discretisation that has diagonal material matrices are used, the inversion of $([M_{\mu^{-1}}] + \frac{\Delta t}{2}[\sigma_*])$ and $([M_\epsilon] + \frac{\Delta t}{2}[\sigma])$ are trivial, resulting in both the $\{b\}$ and $\{e\}$ updates being fully explicit. The stability properties of this method is discussed in §4.4.5.

4.4.3 Fully Discrete PDE II

The Newmark-β method is used to fully discretise PDE II. The Newmark method was first introduced [66] in a two-parameter (γ and β) version. The choice $\gamma = \frac{1}{2}$ is very frequently desirable[26]; fixing γ at $\frac{1}{2}$ results in the Newmark-β method. The Newmark-β method is generally applicable to equations of the form

$$[M]\frac{d^2}{dt}\{x\} + \frac{d}{dt}[D]\{x\} + [K]\{x\} = -\{f\}, \quad (4.104)$$

[25] Another approach that leads to a potentially simpler $\{b\}$ update can be derived from the discrete differential forms perspective. The matrix $[\sigma_*]$ as defined in (4.61) can be seen as a triple-Hodge. It applies $\star_{\mu^{-1}}$ to transform the B 2-form to the H 1-form, followed by $\star_{\sigma_*}$, resulting in the magnetic conductivity contribution to $\frac{\partial}{\partial t}B$, followed again by $\star_{\mu^{-1}}$ to match the left hand side 1-form $\frac{\partial}{\partial t}H$. By splitting the discrete operation into two steps such that the new $[\star_{\sigma_*}]_{ij} = \int_\Omega \vec{w}_i^{(2)} \cdot \overline{\overline{\sigma}}_* \overline{\overline{\mu}}^{-1} \vec{w}_j^{(2)}\, dV$ and then using $[M_{mu^{-1}}][\star_{\sigma_*}]$ instead of $[\sigma_*]$ as before. The $[M_{\mu^{-1}}]$ terms in (4.101) then cancel, leaving only $(1 \pm [\star_{\sigma_*}])$ matrices. They would be much more sparse than before if it is assumed that most of Ω does not contain magnetically conductive materials. While the two approaches are not exactly equivalent, the alternative has in fact been used successfully [65, §6.5 and §7.5.3] to implement a simplified PML using consistent elements, i.e. a semi-discretisation that has non-diagonal mass matrices. If a diagonalised mass-matrix semi-discretisation is used the two methods coincide, as is the case in the subsequent use of PDE I based formulations in this text.

[26] In fact, other choices would introduce numerical damping making the method unsuitable for electromagnetics. Furthermore, it would result in a first-order rather than second-order accurate method.

where $[M]$ is symmetric positive definite and $[D]$, $[K]$ are symmetric positive semi-definite. With time-step Δt, and solutions at $t = n\Delta t$, we approximate at discrete time-steps

$$[M]\{\ddot{x}\}^{n+1} + [D]\{\dot{x}\}^{n+1} + [K]\{x\}^{n+1} = -\{f\}^{n+1}, \tag{4.105}$$

$$\{x\}^{n+1} = \{x\}^n + \Delta t\{\dot{x}\}^n + \frac{\Delta t}{2}\left[(1-2\beta)\{\ddot{x}\}^n + 2\beta\{\ddot{x}\}^{n+1}\right], \tag{4.106}$$

$$\{\dot{x}\}^{n+1} = \{\dot{x}\}^n + \frac{\Delta t}{2}\left[\{\ddot{x}\}^n + \{\ddot{x}\}^{n+1}\right]. \tag{4.107}$$

Assuming $\{\ddot{x}\}^n$, $\{\dot{x}\}^n$ and $\{x\}^n$ are known, there are three equations and three unknowns in (4.105-4.107), allowing the values at time-step $n+1$ to be solved. The approximate derivatives $\{\ddot{x}\}$ and $\{\dot{x}\}$ can be eliminated[27] [14] by writing (4.106) and (4.106) for the time step $n-1, n$ and (4.105) also for times n and $n-1$. This results in the more familiar and useful form where $\{x\}^{n+1}$ can be solved in terms of only $\{x\}^n$ and $\{x\}^{n-1}$:

$$\begin{aligned}\left[\frac{1}{\Delta t^2}[M] + \frac{1}{2\Delta t}[D] + \beta[K]\right]\{x\}^{n+1} &= \left[\frac{2}{\Delta t^2}[M] - (1-2\beta)[K]\right]\{x\}^n \\ &- \left[\frac{1}{\Delta t^2}[M] - \frac{1}{2\Delta t}[D] + 2\beta[K]\right]\{x\}^{n-1} \\ &- \left[\beta\{f\}^{n+1} + (1-2\beta)\{f\}^n + \beta\{f\}^{n-1}\right].\end{aligned} \tag{4.108}$$

Applying the above method to the semi-discrete PDE II (4.77) is a simple matter of identifying

$$[M] = [M_\epsilon], \qquad [D] = [M_\sigma], \qquad [K] = [S], \quad \text{and} \quad \{f\} = \{f_{II}\}. \tag{4.109}$$

To simplify the notation, we define

$$\begin{aligned}[A] = \frac{1}{(\Delta t)^2}[M_\epsilon] + \frac{1}{\Delta t}[M_\sigma] + \beta[S], \quad [B] = \frac{2}{(\Delta t)^2}[M_\epsilon] - (1 - 2\beta[S]), \\ [R] = \frac{1}{(\Delta t)^2}[M_\epsilon] - \frac{1}{\Delta t}[M_\sigma] + \beta[S].\end{aligned} \tag{4.110}$$

Note that in the absence of electric conductivity, $[A] = [R]$. With these combined matrices, the update equation is

$$\begin{aligned}[A]\{x\}^{n+1} &= [B]\{x\}^n - [R]\{x\}^{n-1} \\ &- \left[\beta\{f_{II}\}^{n+1} + (1-2\beta)\{f_{II}\}^n + \beta\{f_{II}\}^{n-1}\right].\end{aligned} \tag{4.111}$$

The parameter β changes the characteristics of the method. With $\beta = 0$ the method is explicit, containing only $[M_\epsilon]$ in the LHS matrix $\{A\}$. As

[27] Zienkiewicz [14] also derives a rather elegant weighted residual method that can be used to obtain this form, and is extensible to higher order accurate methods

discussed in §4.4.5, the choice $\beta \geq \frac{1}{4}$ results in unconditional stability and is one of the most appealing features of the Newmark-β method. To date this represents the most popular electromagnetics time domain FEM method, and is well known [21, §12] in the electromagnetics community.

4.4.4 Eigen solution

A typical application for eigen-solution is obtaining the resonant modes of a cavity. On the other hand, the eigenvalues of the semi-discrete systems also play a critical role in determining the stability of the discrete time-domain solution. When solving practical problems, the lower order modes are usually of interest; for stability purposes, it is usually the mode with the largest eigenvalue that is important.

To obtain the eigen systems, $\frac{d}{dt}$ is replaced with[28] $-j\omega$ and the losses ignored. In the case of PDE I, $\{b\}$ is eliminated. The resulting eigen system for PDE I is

$$[C]^T[M_{\mu^{-1}}][C]\{e\} = \omega^2[M_\epsilon]\{e\}, \tag{4.112}$$

and for PDE II

$$[S]\{e\} = \omega^2[M_\epsilon]\{e\}. \tag{4.113}$$

As noted before, $[S] = [C]^T[M_{\mu^{-1}}][C]$, implying that PDE I&II in fact have identical eigen solutions. This is quite logical, since both methods model exactly the same semi-discrete spaces. When using exact time integration as implied by $j\omega$ there is no practical difference between PDE I&II.

Due to the large number of spurious zero-frequency modes inherent in the semi-discretisations, solving eigen modes of interest to practical problems are somewhat challenging; solving for the largest eigenvalue on the other hand is quite simple, requiring only matrix-vector products when solved with a library such as ARPACK [67]. A cheaply calculated upper bound to the largest eigenvalue of a system obtained by a finite element assembly procedure is provided by the largest eigenvalue of any individual element [12]. It is argued in [12] that for structural dynamics problems the highest elemental eigenvalue provides a fairly tight upper bound to the largest global eigenvalue. In the present author's experience, that is not the case for electromagnetic problems when using tetrahedral meshes. Since the cost of computing the largest eigenvalue accurately is usually equivalent to only that of a few time steps, it seems worth doing if a conditionally stable method is used with a tetrahedral mesh.

4.4.5 Stability

The stability of the leapfrog method is fairly well studied in the context of the FDTD [6], but the FDTD analysis assumes a regular grid. In [68] it

[28] Assuming the $e^{-j\omega}$ time convention.

is shown that with an arbitrary mesh, stability of the system defined by (4.101) and (4.102) is ensured by

$$\Delta t \leq \frac{2}{\sqrt{\lambda_{\max}}}, \tag{4.114}$$

where $\lambda_{\max}$ is the largest eigenvalue of (4.112). The familiar FDTD Courant-Friedrichs-Levy limit can be recovered by noting that $\lambda_{\max}$ of a regular grid is given by the largest elemental eigenvalue. This eigenvalue can be calculated analytically [4] as

$$\lambda_{\max}^{\text{FDTD}} = \frac{4}{\epsilon\mu}\left(\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2}\right), \tag{4.115}$$

where Δx, Δy and Δz are the geometric grid-step sizes in the x, y and z directions. Substituting (4.115) into (4.114) yields the familiar result:

$$\Delta t^{\text{FDTD}} \leq \frac{1}{c\sqrt{\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2}}}. \tag{4.116}$$

It is recommended to back off slightly from the Courant limit, since running at the limit may result in weakly unstable, i.e. linear growth behaviour [69].

The stability of the Newmark-β method is studied extensively in [12], in the context of structural dynamics. The results are however directly applicable to the electromagnetics case. A useful summary and explanation of the results are in [13, §9]. In the lossless case, the stability of the Newmark-β method is given by[29]

$$\Delta t \leq \frac{\Omega_{\text{crit}}}{\sqrt{\lambda_{\max}}}, \tag{4.117}$$

where

$$\Omega_{\text{crit}} = \frac{1}{\sqrt{\frac{1}{4} - \beta}}. \tag{4.118}$$

When $\beta = 0$, (4.118) coincides with (4.114) for PDE I. Values of $\beta < \frac{1}{4}$ imply increasingly larger values of the maximal Δt for stability, with $\beta = \frac{1}{4}$ implying $\Omega_{\text{crit}} \to \infty$ and thereby unconditional stability. For $\beta > \frac{1}{4}$ (4.118) no longer holds, but the resulting scheme is still unconditionally stable. The addition of electric conductivity losses (equivalent to viscous damping in [12]) relaxes the requirement on Δt; the lossless estimate is therefore a safe choice.

Under some conditions the Newmark-β method may suffer from weak instabilities, i.e. linear growth. This happens when they system amplifica-

[29]Note that the Ω_{crit} notation is borrowed from [12] and represents a normalised frequency. It is not related to the physical domain Ω in any way.

tion matrix[30] has repeated eigenvalues. The amplification matrix of PDE II does in fact contain such eigenvalues, but in practice the linear growth modes are very seldom a problem. See also the discussion in §7.4.2.

Two popular choices are $\beta = 0$ and $\beta = \frac{1}{4}$. With $\beta = 0$ an explicit scheme results when combined with a suitable semi-discretisation as discussed in §4.4.3. With $\beta = \frac{1}{4}$ the method is unconditionally stable, which is particularly useful when dealing with inhomogeneous meshes. Values of $\beta > \frac{1}{4}$ are still unconditionally stable, but tend to be less accurate. Furthermore, the greater weight of the singular $[S]$ matrix in $[A]$ (4.4.3) implies a more poorly conditioned matrix equation. Values $0 \leq \beta < \frac{1}{4}$ are unattractive since they imply the computational cost of an implicit method without the advantage of unconditional stability[31].

4.5 Sources of Solution Error

In a FEM solution, there are several discretisations, leading to several sources of error. First the geometry is discretised using a mesh. As discussed in §4.3.1, meshes do not always conform exactly to geometry. As is seen in §10, using Cartesian hexahedra to discretise circular shapes results in a fairly large error, requiring a highly refined mesh to obtain reasonable results. Next, the continuum fields defined on the mesh is discretised using a finite number of basis functions. The typical FEM discretisations like those discussed in §5.2 use piecewise polynomial approximations. Physical fields are seldom exactly representable using piecewise polynomials, hence another error is made. Finally time-discretisation is applied to the semi-discrete system, adding yet another source of error.

Besides errors induced by the basic discretisation process, peripheral factors also come into play. If a finite domain is trucated by the PML as described in §6.3, another source of error is the imperfect absorption of outgoing waves by the PML. Imperfect numerical modelling of source excitations may also contribute errors.

To obtain accurate results, it is neccesary to balance all the sources of error. It is no use discretising the field using high order bases if the meshing error is relatively larger. Similarly, using a very small value of Δt to minimize time-integration error is of no use if the geometry or field is inaccurately represented. In some sense the hybrid method subsequently developed is an attempt at giving an analyst the tools to find the best balance of error

[30] The amplification matrix of a time-stepping scheme is created by writing the update equation as a single matrix that is multiplied by the current value to calculate the next value. A semi-discrete system with m unknowns has an $2m$x$2m$ amplification matrix when using Newmark-β, since $\{e\}^n$ and $\{e\}^{n-1}$ are treated as a single "current" value.

[31] An interesting choice is $\beta = \frac{1}{12}$ which results in the Fox-Goodwin method. In the absence of losses, it is fourth rather than second order accurate, and is for this reason sometimes referred to as the "royal road method."

among the several sources of error.

4.6 Conclusion

A number of theoretical FEM topics that are neccesary for the subsequent implementation of the higher order hybrid implicit/explicit hybrid mesh FEM have been recounted. Weak forms of the the PDE I and PDE II forms of Maxwell's equations amenable to discretisation using the FEM have been derived. The Galerkin method has been applied to the weak forms of PDE I&II to derive finite element semi-discrete representations of PDE I&II. Particular attention has been given to the precise definition of the function spaces involved. The semi-discrete representations have also been discussed in the light of discrete differtial forms. Fully discrete representations of PDE I&II have been derived by introducing suitable time discretisation schemes, viz. central differences (leap-frog) for PDE I and the Newmark-β method for PDE II. The characteristics of the time discretisation schemes have been discussed. Sources of error in the basic fully discrete formulations have briefly been touched upon, and the balance requisite for efficient and accurate solution discussed. Discussion of the numerical implementation of FEM methods have however been left for §5.

Chapter 5

Numerical FEM Implementation

The previous chapter (§4) laid the theoretical foundation on which the finite element method is built. It lacks, however, much of the concrete detail needed to numerically implement a practical FEM program. In one sense that is deliberate, since the more abstract treatment allows the consistent treatment of a smorgasbord of different numerical schemes that have widely varying practical characteristics. In this chapter the concrete choices that make the hybrid mesh implicit/explicit hybrid formulation described in §7 and §8 possible are described, motivated and analysed in terms of their practical significance.

The first section deals with certain general features of FEM system matrices and their construction. The second section discuses the concrete basis functions used to implement the discrete $H(curl;\Omega)$ and $H(div;\Omega)$ spaces. The third section describes the FDTD from a FEM perspective. The last section discusses some aspects of the software implementation of the above FEM methods using the very high level language Python [17] along with the numpy [18, 19] and scipy [20] array, matrix and scientific computing libraries.

5.1 General FEM System Characteristics

In one sense, the construction of the system matrices is the essence of any finite element method; it is the nature of these matrices that will determine the final accurary and computational efficiency of the method. While the exact nature of a particular FEM implementation's system matrices are determined by the specifics of the implementation, there are certain general characteristics that are determined by the general nature of: the finite element meshes described in §4.3.1; the finite element function spaces described in §4.3.3; and the bilinear forms (4.59) through (4.63) and (4.78).

These characteristics are discussed in this section.

5.1.1 Compact Support

Probably the most practically significant property of finite element basis functions are their support. The support of a function is informally defined as the geometric domain where the function is not zero. The support of a function f is written[1] as $\text{supp}(f)$. As discussed in §4.3.2, basis functions have degrees of freedom related to either a mesh edge, face or volume. Since the basis functions are always defined in terms of a reference element, a basis function's smallest "unit" of support is a single mesh element. For a basis function defined by a degree of freeom on a mesh entity ℓ, where in this instance ℓ can be an edge, face or volume, its support is the union of all the elements K_i such that K_i has a non-zero intersection with ℓ. More formally, given a basis function $\vec{w}_\ell$ defined by its degree of freedom on mesh entity ℓ,

$$\text{supp}(\vec{w}_\ell) = \bigcup_i K_i \qquad \text{for all } i \text{ such that } K_i \cap \ell \neq \emptyset. \tag{5.1}$$

This implies that if ℓ is a volume, the support is exactly one element, i.e. the one containing the volume; if ℓ is a face, the support is the two elements that meet at that face, and if ℓ is an edge, the support is made up of all the elements connected to that edge. The exact number of elements connected to a given edge[2] is mesh dependent, as discussed in §4.3.1.

5.1.2 Elemental vs. Degree of Freedom Interpretation

In the mathematical sense, a given basis function $\vec{w}_i$ is defined by its value over $\text{supp}(\vec{w}_i)$. Furthermore, $\text{supp}(\vec{w})_i$ is defined by the mesh entity ℓ over which the degree of freedom functional $\Xi_i(f)$ is defined. If, for instance, $\vec{w}_i$ has a degree of freedom over the edge ℓ, we can see $\vec{w}_i$ as "belonging" to the edge ℓ and "living" in its support defined by the elements connected to ℓ. This is in contrast to the common "element-centric" description of finite element bases where each function is defined only in terms of its values on a reference element. There are good practical reasons for the element-centric view of finite element functions, but the degree of freedom centered view is important to keep in mind, particularly when precise descriptions of basis function interactions are important such as in §7. The relation between the two views are discussed by way of a simple 1-D example below.

Consider the 1-D geometry shown in Fig. 5.1. Here the mesh "elements" are the line segements between the nodes. The basis functions are simple piecewise linear functions, with degrees of freedom defined at the nodes,

[1] Not to be confused with the notation $\sup(\cdot)$ used for supremum.

[2] The same would also be true of nodes if nodal basis functions were used.

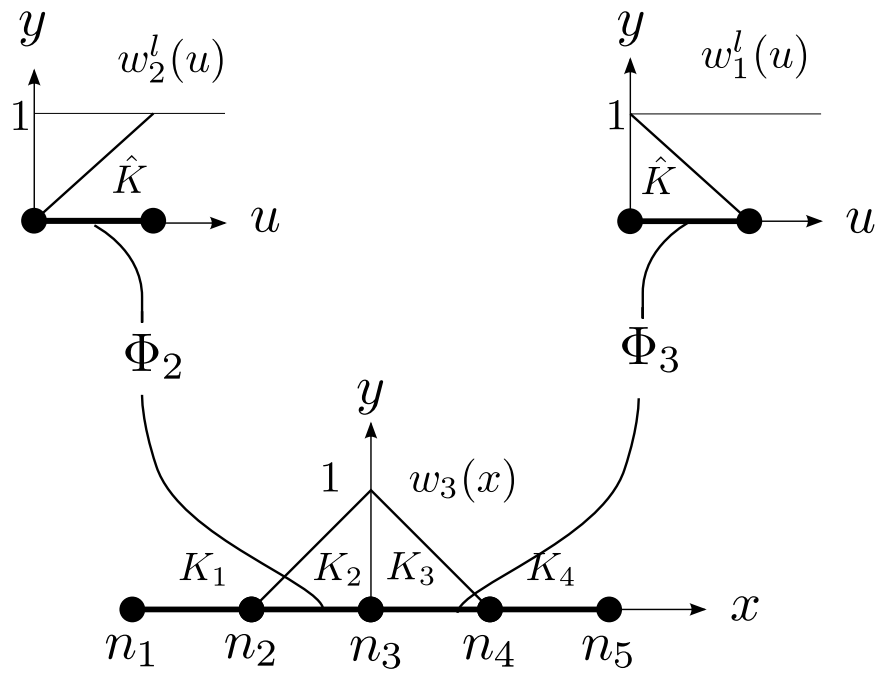


Figure 5.1: Simple 1-D illustration of how the reference element $\hat{K}$ is transformed to define a global basis function.

i.o.w. they are nodal functions. The elemental basis is defined on the reference element $\hat{K}$ with reference coordinate u, and is made up of the two element-local functions $w_1^l(u)$ and $w_2^l(u)$. The reference element is transformed to the global element K_i by the coordinate transform Φ_i. The global basis function $w_3(x)$ is shown on the global x coordinate system with the function value represented by the y axis. It is clear that $\text{supp}(w_3) = K_2 \cup K_3$. From the element-centric perspective, w_3 is equivalent to w_2^l if we are in K_2 and equivalent to w_1^l if we are in K_3. If we are not in K_2 or K_3, w_3 simply does not exist from the elemental perspective. From the degree of freedom perspective, w_3 belongs to the mesh entity n_3, and has support in the elements K_2 and K_3 that intersect n_3. Outside its support, w_3 still exists, but it is defined as 0. We can fully define w_3 as

$$w_3(x) = \begin{cases} w_2^l(\Phi_2^{-1}(x)) & x \in K_2 \\ w_1^l(\Phi_3^{-1}(x)) & x \in K_3 \\ 0 & x \notin K_2 \cup K_3. \end{cases} \tag{5.2}$$

The advantage of the elemental view is that it provides a simple well defined domain over which to define basis functions, and provides a fairly straight forward method of constructing the system matrices element by element. Care must hower be taken to ensure that the basis functions as

defined on the elements result in a mathematically consistent description of the global basis functions. Consider again w_1^l and w_2^l as defined on $\hat{H}$ in Fig. 5.1. The degree of freedom based definition of w_3 requires that both element-local functions attain the same value at n_3. It is fairly simple to ensure for scalar functions by having both w_1^l and w_2^l attain unity at the ends of $\hat{K}$. For the vector valued 3-D $H(curl;\Omega)$ and $H(div;\Omega)$ elements used for Maxwell's equations it is somewhat more complicated[3], but the same principle holds.

5.1.3 Geometric Data Structures

Due to the fact that all the mesh entities are interconnected, FEM codes usually have data structures describing these connections. Suppose a mesh has n_n nodes, n_ℓ edges, n_f faces and n_v volumes/elements and that each element entity type has a unique numbering. That is, nodes are numbered $1 \ldots n_n$, edges $1 \ldots n_\ell$ and similarly for faces and elements. An simple elemental data structure might consist of, per element,

node numbers An array of the node numbers that define the element.

edge numbers An array of the numbers of the edges connected to the element.

face numbers An array of the numbers of the faces connected to the element.

The order of the arrays stored in the elemental data structure is significant. Given e.g. the triangular reference element shown in Fig. 5.2, the local edge numbering is linked to the global edge numbering in order through the elemental "edge numbers" array. These local to global mesh entity numberings are used to construct maps for elementally defined basis functions to the global basis functions; this is discussed next.

Structures defining other interconnections, e.g. node→ element, element→ element, etc. may also be defined. Such structures will subsequently be described as the need arises.

5.1.4 Global Degree of Freedom Numbering

The global degrees of freedom of a FEM discretisation are defined in terms of mesh entities, while in practice most computations are done on an element by element basis. FEM meshes are usually set up in such a way that all the mesh entities have unique numbers. In the case of low order[4] discretisations

[3] For $H(curl;\Omega)$ and $H(div;\Omega)$ elements it is respectively the tangential and normal components at shared mesh entities that need to be continuous.

[4] "Low order" implies the lowest order piece-wise polynomial basis that can model a certain space. For H^1 elements that would be first order (i.e. linear) polynomials, while for the $H(curl)$ and $H(div)$ elements considered here it would be mixed first order.

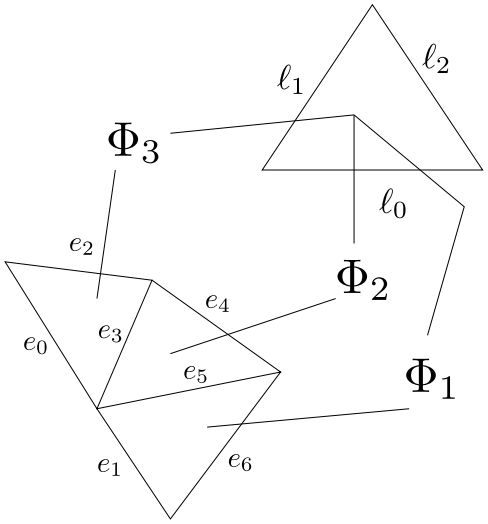


Figure 5.2: Triangular mesh showing element-local edge numbers ℓ_0, ℓ_1 and ℓ_2 related to global edge numbers e_0 trough e_6 via the edge numbers data structure.

the degree of freedom numberings can be defined quite simply in terms of the mesh entity numberings. For instance, when using mixed first order $H(curl;\Omega)$ elements[5], each mesh edge has exactly one degree of freedom associated with it while no other mesh entities have associated degrees of freedom. This implies that the edge numbering inherited from the mesh is also suitable as a global degree of freedom numbering. When using higher order $H(curl;\Omega)$ discretisations, there will usually be more than one degree of freedom defined per mesh edge; additionally there will also be degrees of freedom related to the faces and potentially volumes.

While one may unambigously refer to e.g. $\vec{w}_{i(l_j)}$ as the basis function related to the j'th degree of freedom on the i'th edge, the construction of the system matrices requires a single, contiguous and unambigious numbering scheme where every free degree of freedom defined on the mesh is numbered exactly once. There is no single way of defining global degree of freedom numberings for higher order discretisations, with the method chosen usually depending on practical expediency. An important requirement is that a local numbering defined by the degrees of freedom on the parent element $\hat{K}$ can be transformed to the global numbering using the elemental data structure described in §5.1.3.

The particular global degree of freedom numbering scheme in use is mostly irrelavant, so long as the requirements just sketched are met. The

[5]Here we are ignoring for the moment boundary conditions. Generating numberings based on the edge numbering in the presence of constrained edge DOFs are still quite simple.

particular numbering will not affect the sparsity percentage of the resulting matrix, but will affect the pattern, i.e. exactly where the non-zeros lie. For some matrix storage and/or solution schemes the exact nature of the sparsity pattern is significant. Aside from noting that the storage and solution schemes considered subsequently are mostly agnostic to the particular numbering used, this topic is beyond the current scope.

5.1.5 System Matrix Sparsity Pattern

The sparsity pattern[6] of the FEM system matrices depends on the degree of freedom numbering scheme used and on the support of the basis functions. Assuming a given set of n_{tot} globally numbered basis functions $\vec{w}_i$, $i = 1 \dots n_{\text{tot}}$, we will now determine the potential locations of the non-zero entries of a finite element system matrix $[A]$. While specific finite element basis sets may result in fewer non-zeros, any basis set satisfying the assumptions of §4.3.2 cannot have any non-zeros outside the locations we are about to derive.

System matrix entries generally take the form

$$[A]_{ij} = \int_{\Omega} \vec{w}_i \cdot \vec{w}_j \, dV. \tag{5.3}$$

It is clear from the definition of (5.3) that if either one of $\vec{w}_i$ or $\vec{w}_j$ are zero at given point, the integrand will also be zero. For an entry $[A]_{ij}$ to be non-zero, there must be some sub-domain of Ω with non-zero measure where both $\vec{w}_i$ and $\vec{w}_j$ are non-zero. This sub-domain is simply the intersection the functions' support. More formally,

$$\text{supp}(\vec{w}_i) \cap \text{supp}(\vec{w}_j) = \emptyset \implies [A]_{ij} = 0, \tag{5.4}$$

$$[A]_{ij} \neq 0 \implies \text{supp}(\vec{w}_i) \cap \text{supp}(\vec{w}_j) \neq \emptyset. \tag{5.5}$$

Of course, given $\text{supp}(\vec{w}_i) \cap \text{supp}(\vec{w}_j) \neq \emptyset$ it is still possible for $[A]_{ij}$ to be zero if $\vec{w}_i$ and $\vec{w}_j$ are orthogonal under the inner product implied by (5.3).

5.1.6 Evaluating the Inner Products

Calculating the numerical entries for FEM system matrices requires the numerical evalution the inner products defined in §4.2.1, resulting in integrals of the form (5.3). Since the functions involved are polynomial, exact evaluation of (5.3) is feasible, but as will subsequently be discussed, not always desirable. Because of FEM basis functions having compact support, it is not desirable to evaluate integrals over all of Ω.

[6] The sparsity pattern of a matrix is a "picture" showing the location of all the non-zero entries.

Following the discussion in §5.1.5, it is clear that for a matrix entry $[A]_{ij}$, it is only neccesary to evaluate (5.3) over the domain $\text{supp}(\vec{w}_i) \cap \text{supp}(\vec{w}_j)$, i.e.

$$[A]_{ij} = \int_{\text{supp}(\vec{w}_i)\cap\text{supp}(\vec{w}_j)} \vec{w}_i \cdot \vec{w}_j \, dV. \tag{5.6}$$

Furthermore for any i, j such that $\text{supp}(\vec{w}_i)\cap\text{supp}(\vec{w}_j) = \emptyset$, $[A]_{ij} = 0$, and no integral evaluation is neccesary. Given the definition of any basis function's support by (5.1) as a union of mesh elements and the fact that the basis functions are guaranteed to be continuous within a single element, it makes sense to decompose the integration domain into distinct elements. In this subsection we consider only the evaluation of the integrals on individual elements. The assembly of the element-wise integrals into the global matrix is considered in the following subsection.

Since basis functions are typically defined on a reference element $\hat{K}$, it is generally convenient to evaluate the integrals on the reference element using a change of coordinates. We assume that a set of element local basis functions $\vec{w}_i^l(\mathbf{u})$ are defined on $\hat{K}$ where $\mathbf{u}$ is a coordinate defined on the reference element's coordinate system. For each element K_k there exists a coordinate transform Φ_k (or equivalently Φ_{K_k}) such that $K_k = \Phi_k(\hat{K})$. The coordinate transform Φ_k implies a Jacobian matrix J_k that is required when a change of variables are applied to an integral. Using g_i to refer to the global degree of freedom number of the element local function $\vec{w}_i^l$, an integral defined in global coordinates (r):

$$[A]_{(g_i)(g_j)}(K_k) = \int_{K_k} \vec{w}_{g_i}(\mathbf{r}) \cdot \vec{w}_{g_j}(\mathbf{r}) \, dV \tag{5.7}$$

into

$$[A]_{(g_i)(g_j)}(K_k) = \int_{\hat{K}} \left(\vec{w}_i^l \circ \Phi_k\right)(\mathbf{u}) \cdot \left(\vec{w}_j^l \circ \Phi_k\right)(\mathbf{u}) \, \text{abs}\left(|J_k(\mathbf{u})|\right) \, dV, \tag{5.8}$$

where $\circ$ is functional composition, $|\cdot|$ the determinant and abs() the absolute value. Because volume integrals are *orientable*, the sign of the integral (5.7) depends on the orientation[7] of the element K_k. Since the original global integration calls for the positive orientation, negatively oriented integrals need to be negated to ensure the correct sign; using the absolute value of $|J_k|$ effectively manages this [48, Remark 4].

For certain element classes, such as the $H(curl)$ and $H(div)$ rectilinear tetrahedra and Cartesian hexahedra considered subsequently, it is possible to evaluate (5.8) only once for the reference element. Transforming the integral values to a global element tetrahedral requires only some cheap calculation involving the transformation of the basis-vectors on which the

[7] The orientation of the integral volume is the multi-dimensional equivalent to $\int_a^b f\,dx = -\int_b^a f\,dx$.

vector values basis functions are defined [8]. Through some basic computer algebra, given the form of arbitrary order bases defined in [8] the sytem matrices for arbitrary order bases can even be generated with no additional implementation work [70]. Analytical pre-calculation is somewhat simpler for Cartesian hexahedra since the elemental basis vectors are always aligned with the global Cartesian $\hat{x}$, $\hat{y}$ and $\hat{z}$ basis vectors.

Another popular approach is numerical quadrature [71]. Efficient rules that allow exact integration of up to 8th order polynomials on tetrahedral domain have been published [72]. Quadrature rules for Hexahedra can be constructed quite simply as a Cartesian product of the familiar 1-D Gaussian quadrature rules. Compared to the analytical approach, numerical quadrature involves extra computation work at runtime, but is more flexible. Using numerical quadrature also makes the use of curvilinear elements – which do not allow analytical integration – possible [45], [46], [73]. In spite of the extra computational work required, it generally does not have a significant impact on the total runtime of a FEM solution, particularly if the reference-element function values are cached [74, pp. 21].

While the analytical and quadrature approaches attempt to evaluate (5.8) exactly[8], a judicious combination of basis functions and approximate evaluation of the inner product can result in a system with a diagonal metric matrix. This results in a fully explicit[9] method when used with explicit time integration as described in §4.4. With a suitable reduced integration scheme the asymptotic accuracy of the explicit method will be identical to that of the consistent method while saving much computation effort. Since these methods were intially developed for structural mechanics, they are often refered to as "mass lumping" methods. A good summary is [13, §7.3.2], where amongst other topics the Gauss-Lobatto integration rules that are subsequently used are covered, although the structural mechanics approaches are not always directly transferable to Maxwell's solvers. Diagonalisation schemes for Maxwell's equations are considered in more detail subsequently.

5.1.7 Global System Matrix Construction

Given the concepts previously described in this section, we are now in a position to define an algorithm for the construction of the global matrix. To make the algorithm operational, the following is needed:

- A well defined reference element $\hat{K}$ with local coordinates $\mathbf{u}$.

[8] When the inner products are exact the resulting matrix is said to be "consistent."

[9] Fully explicit implies that no matrix inversion is neccesary to time-step the fully discrete system. Explicit time integration alone does not result in a fully explicit system if the metric matrix of the semi-discretisation is not diagonal; similarly a diagonal metric matrix does not result in a fully explicit method unless it is combined with explicit time integration.

- Basis functions $\vec{w}_i^l$ defined on $\hat{K}$ in a manner consistent with the global degrees of freedom.
- A consistent, contiguous and unambigious global numbering scheme that assigns a number to every gobal degree of freedom.
- A routine that maps basis function degrees of freedom on $\hat{K}$ to the global numbering.
- A method for numerically evaluating the integrals implied by the inner products. Now the global assembly procedure can be described by Algorithm 1.

Algorithm 1 Element-wise assembly of global system matrix $[A]$

for all $K \in \{K_1, K_2, \ldots, K_{n_v}\}$ **do**
  for $i \in 1 \ldots$ number of basis functions on K **do**
    $g_i =$ the global DOF number for $\vec{w}_i^l$
    for $j \in 1 \ldots$ number of basis functions on K **do**
      $g_j =$ the global DOF number for $\vec{w}_j^l$
      $[A]_{(g_i)(g_j)} = [A]_{(g_i)(g_j)}$
        $+ \int_{\hat{K}} \left(\vec{w}_i^l \circ \Phi_K\right)(\mathbf{u}) \cdot \left(\vec{w}_j^l \circ \Phi_K\right)(\mathbf{u}) \operatorname{abs}\left(|J_K(\mathbf{u})|\right) \, dV$

Note that the support of the basis functions is never explicitly considered in Algorithm 1. By looping over elements and then only considering the degrees of freedom connected to each element in turn, the correct integration domain for each basis function is automatically chosen.

5.2 Concrete Finite Element Basis Sets

The treatment of numerical FEM implementation presented in §5.1 is a somewhat general and brief overview; in this section concrete basis sets and mesh types are provided that can be "plugged into" the general description. As alluded to before, finding a particular choice of discrete basis to approximate the continuum $H(curl)$ and $H(div)$ spaces has historically been difficult. Today it is uncontroversial to suggest that the tetrahedral and hexahedral bases about to be presented are the correct[10] ones. In spite of some subtle mathematical questions that remain open, there is a strong concensus based on practical results that these bases do indeed work as advertised. The exact statement of what conditions are neccesary and sufficient to avoid spurious solutions still seems to be open for discussion. While no attempt

[10] More accurately, bases span the correct *spaces*, i.e. those of [10] and [11]. These spaces permit an arbitrary choice of bases, provided of course that the particular bases adhere to the definition of the spaces.

will be made to address this question here, mention is made of a fairly recent paper that reviews the previously proposed conditions and attempts to define sufficient conditions that are not unnecessarily restrictive [75]. In spite of the succesful application of the spaces defined by Nédélec on tetrahedra and hexahedra, things are not so simple for other element shapes. As seen in §8, the higher order $H(curl)$ pyramidal elements currently available in the literature seem to suffer from spurious modes; in the absence of clear criteria it is hard to say apriori if a given discrete space will be free of spurious modes. A fairly extensive description and mathematical treatment of bases on several elemental shapes is [76].

In this section the tetrahedral and hexahedral bases that are subsequently used to implement finite element methods, and also used to construct a hybrid mesh in §8 are described. Before the specific bases are described, some general properties that $H(curl)$ and $H(div)$ bases may posess and their practical significance are considered. The next subsection provides a short overview of tetrahedral bases for electromagnetics use and describes some specific aspects of the tetrahedral bases used for 1-form [8] and 2-form [77] discretisation. The last subsection describes the diagonalisable hexahedral bases used [78]. Since the properties of these elements are central to enabling the implicit/explicit hybrid, the will be consdered in more detail than the tetrahedral elements.

5.2.1 Basis Function Properties

An important point to note is that irrespective of the other properties of the particular basis set used, the final accuracy of a FEM solution depends only on the space coverd by the given basis, and not the specific form of the basis function, see e.g. [79], [80]. The choice of a particular basis is made on the grounds of other properties, some of which are discussed below. Some pertinent basis function properties are:

order The order p of a basis refers to the highest complete order of the polynomial space that the base covers. I.e. a p'th order base includes at least all monomial[11] terms to order p and below.

mixed order Same as for "order" above, but with the part of the polynomial space that does not contribute to respectively the curl or divergence of $H(curl)$ or $H(div)$ elements reduced by one order. As discussed in §4.3.3, mixed order elements often lead to optimal convergence.

interpolatory Interpolatory basis functions are defined such that each basis function attains unity at exactly one location within an element;

[11] A monomial of order p is a polynomial term with a total power of p. For single variable polynomials, the order of the monomial is simply the power, e.g. x^a is a monomial of order a. In the multivariate case, $x^a y^b z^c$ is a monomial of order $a+b+c$.

this point is its interpolation point. Furthermore each basis function is identically zero at all the interpolation points associated with the other basis functions in the set.

hierarchical Hierarchical basis functions are defined such that as the polynomial order of the basis is increased from p to $p+1$, it is only neccessary to add new functions to the original order p base; the original lower order basis functions are never discarded.

diagonalisable Diagonalisable basis functions are such that an approximate inner product as described in §5.1.6 can be found that will result in a semi discretisation with a diagonal metric matrix, without unduely impacting its error convergence.

Interpolatory basis functions are usually constructed with the aid of interpolatory polynomials. For each order p a new interpolatory base needs to be defined, since each interpolation polynomial needs to be of the some order to ensure they have the right number of zeros. For this reason, the interpolatory and hierarchicality properties are mutually exclusive – a basis can be either interpolatory, hierarchical or neither. Since the interpolatory and hierarchicality properties both confer certain advantages, bases are almost always constructed to be the one or the other.

An extensive treatment of interpolatory bases is [81]. For a given elemental shape, all interplatory bases are essentially the same apart from the precise location of the interpolation points. The location of the interpolation points have a significant effect on the conditioning of the FEM matrix equations that need to be solved [82]. An advantage of interplatory bases is that they provide a straight forward interpretation of the degrees of freedom as the values of a function at the interpolation points. This property can be used to construct an interpolation operator with minimal computational cost by sampling the function to be interpolated at the interpolation points; here again the locations of the interpolation points are quite important. For instance, it is well known that the use of interpolatory polynomials to reconstruct a function sampled at regularly spaced interpolation points can lead to an error without bound as the order of the interpolatory polynomials are increased. Furthermore, approaches to constructing diagonalisable bases frequently depend on a particular choice of interpolation points.

Hierarchical bases are primarily useful for adaptive p-refinement methods, e.g. [83], [84] or combined hp-refinement, e.g. [85], since they allow different basis orders to be used throughout the mesh. Since the lower order basis functions remain unchanged as p-refinement is applied, the lower order system matrix and preconditioner entries can be re-used for the refined solution. Most hierarchical bases also allow the selective use of mixed or full order elements with little programming effort [53]. Hierarchical bases also fascilitate certain system matrix solution strategies, e.g. [50], [86], [80].

One disadvantage is that it is somewhat harder to define a computationally efficient interpolation operator for hierarchical bases [87], [88].

Diagonalisable bases are required for fully explicit time domain methods. Since the computational efficiency inherent to a fully explicit method is typically such a compelling advantage, the other properties of a diagonalisable base usually become subservient. While diagonalisable bases are quite common in the computational structural dynamics field [13, §7.3.2], they are quite uncommon in the computational electromagnetics field due to the difficulty involved in creating diagonalisable $H(curl)$ bases. Their application is, however, somewhat less uncommmon than it may seem at first if one takes into account that the FDTD can be seen as the implementation of a particular diagonalised basis set. This is discussed in more detail in §5.3.

A diagonalised system matrix results if

$$[A]_{ij} = \langle \vec{w}_i, \vec{w}_j \rangle\tilde{} = c\delta_{ij}, \tag{5.9}$$

where c is some constant and the tilde indicates the approximate inner product. It is tempting to assume that (5.9) could be satisfied, even using an exact inner product, if the basis is made up of a set of mutually orthogonal polynomials. Unfortunately the inter-element continuity conditions place requirements on the basis that cannot be met by orthogonal polynomials. A common strategy used to satisfy (5.9) is to use an interpolatory basis together with an inner-product based on an approximate numerical quadrature rule where the quadrature points coincide with the interpolation points. The interpolation points should then be chosen to optimise the accuracy of the quadrature rule. The Gauss-Lobatto points are often optimal in this context. They are used with interpolatory hexahedra to define diagonalisable $H(curl)$ and $H(div)$ elements in [78]. Diagonalisable tetrahedra pose a greater difficulty, and are discussed in §5.2.2.

5.2.2 Tetrahedral $H(curl)$ and $H(div)$ Elements

Tetrahedral elements have long been a mainstay of FEM analysis because of unstructured tetrahedral meshes' ability to model complex geometries. In the electromagnetics community tetrahedral elements have a much greater body of literature than just about any other elemental type. This may be because time domain electromagnetic FEM analysis has only recently gained popularity, resulting in the relatively easy diagonalisability of hexahedral elements not having been seen as a great advantage. Another factor might be that the general definition of higher order tetrahedral bases proved to be less straight forward than for hexahedra, resulting in more publications on the topic. As a geometric simplex[12] tetrahedra also offer some basic

[12] A simplex is the simplest non-trivial shape in a given spatial dimension. In 1-D it is a line segment, in 2-D the triangle, and in 3D the tetrahedron.

mathematical conveniences, such as having a convenient simplex coordinate system.

Some of the earliest accounts of using $H(curl)$ elements based on Nédélec's [10] spaces are [89], [90] in the context of eddy-current problems. The first elements higher than mixed first order are the consistently linear elements presented in [91]. One of the first application of $H(curl)$ elements to fullwave electromagnetics is in [92]. Initially approaches to higher order and hierarchical bases were somewhat ad-hoc e.g. [93], [94], [95]. Systematic approaches to higher order interpolatory [81], [96] and hierarchical [8], [97] bases later arose. Later still, the focus shifted to defining bases that result in well conditioned matrix equations. Many of these methods depend on splitting the bases into groups that are gradients and groups that have non-zero curl. Most tetrahedral hierarchical approaches explicity split functions into groups that are gradients (i.e. have zero curl) and those that have non-zero curl. Because of the degree of freedom continuity requirements this is not possible at the local element level for the lowest order basis set. Techniques exist for splitting the two function sets completely [80], [50], and for $H(div)$ elements [77].

The later work on conditioning the matrix equations are less applicable to the time domain FEM. In the frequency domain case, the matrix that must be solved is of the form $[S] - k^2[M_\epsilon]$, where k is the wavenumber and the matrices are as defined in §4.3.2. Since $[M_\epsilon]$ is positive definite and $[S]$ is positive semi-definite, the resulting matrix is indefinite. In the time domain method, the matrix to be solved is $[M_\epsilon]$ alone for PDE I or of the form $[M_\epsilon] + \Delta t^2 \beta [S]$ for PDE II. Both these forms are positive definite, and therefore much easier to solve. For iterative solution the conditioning of $[M_\epsilon]$ is still important, particularly as the matrices have to be solved at every timestep.

Diagonalisable $H(curl)$ tetrahedral elements are a rarity. Early attempts at applying reduced integration to unmodified low order $H(curl)$ elements suffered from potentially singular metric matrices [22]. Diagonalisable tetrahedral bases seem to require the introduction of extra degrees of freedom for first order [98] and second order [99] elements. The latter paper is somewhat inaccesible to the Anglophone community since it is written in French with only a 1 page summary in English. The present author is not aware of any independent use of [99]'s work in the literature. Since the intention of the current work is to use implicit time integration on the tetrahedral mesh, diagonalisable tetrahedral elements were not persued further.

As noted in in §5.2.1 the accuracy of different bases that cover the same space are the same. The bases used subsequently are the $H(curl)$ elements from [8] and the $H(div)$ elements form [77]. These are both hierarchical bases. More importantly, the solenoidal[13] functions defined by [77] were

[13]Solenoidal vecetor functions are one that have zero divergence, i.e. $\nabla \cdot \vec{f} = 0$ if $\vec{f}$ is

derived by taking the curl of the non-gradient functions presented in [8]. This makes the calculation of the curl matrix $[C]$ particularly simple, see §5.2.4. Another advantage of the hierarchical bases is that since we know $\nabla \cdot \vec{B} = 0$, the non-solenoidal $H(div)$ basis functions can be omitted [24].

5.2.3 Hexahedral $H(curl)$ and $H(div)$ Elements

One of the earliest papers that present concrete $H(curl)$ bases on hexahedra is [100]. In spite of its age, it is still fairly up to date, covering higher order as well as curvilinear interpolatory hexahedra; it is a rather brief paper though, only treating 2-D elements directly. The paper [81], which deals with several types of interpolatory $H(curl)$ and $H(div)$ elements is notable for introducing a very flexible face-based coordinate system on the reference element. Whereas tetrahedra, being 3-D simplexes, are endowed with the very convenient simplex[14] coordinate system on the reference element, hexahedral bases are often defined in Cartesian coordinates; sometimes with reference coordinates running from -1 to +1, other times from 0 to 1. The resulting expressions often obscure the symmetry inherent in the basis functions. The face-based scheme results in convenient expressions, and can be extended to almost any element shape; on tetrahedra, it is dual[15] to the standard simplex coordinates[16].

Further work on interpolatory hexahedra in [82] suggested the use of the extended Chebychev interpolation points, rather than the regularly spaced points used before in order to improve the conditioning of the matrix equations. Diagonalisable $H(curl)$ and $H(div)$ hexahedra based on a mixture of the Gauss-Legendre and Gauss-Lobatto[17] interpolation points are presented in [78], along with an approximate inner product based on mixed Gaussian and Gauss-Lobatto numerical quadrature. In [102] the Gauss-Lobatto bases of [78] are applied to non-Cartesian hexahedra; while bases on non-Cartesian hexahedra are no longer diagonalisable, the resulting matrices are much more sparse and somewhat better conditioned than when consistent integration is used. The use of the irregularly spaced interpolation points should also

solenoidal. Any function that can be written as the curl of another function is solenoidal, i.e. $\vec{f} = \nabla \times \vec{g}$ implies that $\vec{f}$ is solenoidal.

[14] Sometimes also called barycentric, homogenous, volume, or in the case of triangles, area coordinates.

[15] Dual in the sense that the usual simplex coordinate ζ_i (see e.g. [41, §4.2.1]) is zero at node 1 and attains unity at face 1 opposite node 1. The face-coordinate would attain unity at node 1, and be zero at face 1.

[16] This seems to suggest that a FEM program dealing with several element types would benefit from using the face-coordinates throughout.

[17] The Gauss-Legendre points and the weights needed for numerical integration are very commonly used; most numerical software include a routine for their calculation. The Gauss-Lobatto rules are less common. A simple Matlab program [101] can be used to calculate them.

improve[18] the quality of approximation if the bases are used to interpolate known fields.

Hierarchical hexahedra are less common that hierarchical tetrahedra. Hierarchical hexahedral elements to arbitrary order are presented in [104], while the same authors suggest the use of Legendre basis polynomials for improved conditioning in [105]. A method for deriving hierarchical bases by specifying the degree of freedom functionals is in [106, 107]. The current literature does not seem to contain any hierarchical hexahedral bases that split the gradient and non-gradient spaces like most of the hierachical tetrahedral bases do.

All the basis sets described by the works cited above are called *tensor product* elements, since the 3-D basis sets are constructed by the tensor product of 1-D polynomial functions in each coordinate. For the purposes of this work, Cohen&Monk's [78] diagonalisable bases are most important and will now be discussed in more detail. The notation we will use is however closer to that of Fisher *et al.* [102] than to Cohen&Monk's.

A p-th order Lagrange interpolatory polynomial is written as

$$L_i^p(x;X) = \prod_{\substack{j=0 \\ j\neq i}}^{p} \frac{x - x_i}{x_i - x_j}, \tag{5.10}$$

where X is a zero-indexed[19] array of $p+1$ interpolation points such that x_i is the i'th member of X. It satisfies the interpolation property:

$$L_i^p(x_j;\mathbf{X}) = \delta_{ij}. \tag{5.11}$$

In other words, $L_i^p(x;\mathbf{X})$ is identically zero at all the interpolation points $x = x_j$, $j \neq i$ and attains unity at $x = x_i$. We define the p'th order Gauss-Lobatto interpolation points as $\mathbf{B}^p$ and the p'th order Gauss-Legendre points as $\mathbf{G}^p$. Now we can define the p'th order Gauss-Lobatto interpolation functions as

$$B_i^p(x) = L_i^p(x;\mathbf{B}^p), \tag{5.12}$$

and the p'th order Gauss-Legendre interpolation functions as

$$G_i^p(x) = L_i^p(x;\mathbf{G}^p). \tag{5.13}$$

[18]The quality of a polynomial interpolation is measured by the Lebesgue constant. A Lebesgue constant of 0 would imply optimal interpolation, i.e. the polynomial interpolation would have a smaller maximum error than any other polynomial interpolation of the same order. The analysis required to find the Lebesgue constant of a given set of interpolation points is rather involved, but some well known results are: 1) No known set of interpolation points have a zero Lebesgue constant. 2) The Lebesgue constant of uniformly spaced points grow exponentially in polynomial order. 3) The Chebychev, extended Chebychev and Gauss-Lobatto points are nearly optimal [103].

[19]Zero indexed arrays have their first element numbered as 0 rather than 1

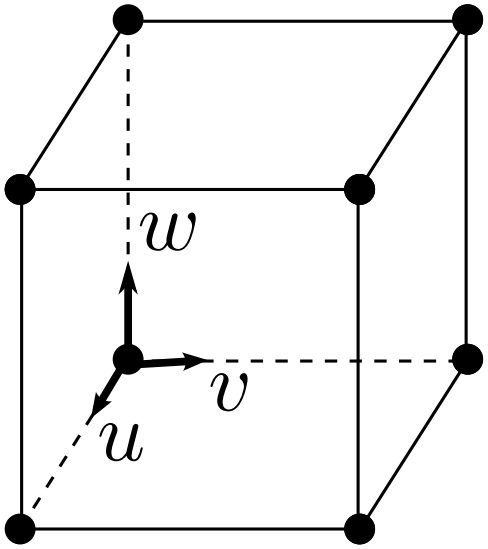


Figure 5.3: Hexahedral reference element.

The Gauss-Lobatto and Gauss-Legendre interpolation points both arise as the sampling points of optimal numerical integration rules. The Gauss-Legendre points form the basis of standard Gaussian quadrature. For a given number of quadrature points n over an interval (a, b), the Gauss-Legendre points together with suitable weights can integrate a polynomial function of order $(2n-1)$ exactly; this order of integration is higher than any other set of n points could manage. All the points in Gauss-Legendre sets are always interior to the domain; that is they never coincide with the ends of the integration domain a or b. The Gauss-Lobatto points, on the other hand, provide the optimal order of integration given that a and b are required to be part of the set. An n point Gauss-Lobatto quadrature rule can exactly integrate a polynomial of order $(2n-3)$. In other words, the price for constraining the quadrature points to include the interval endpoints is a reduction in integration order of 2.

The hexahedral reference element with local coordinates u, v, and w shown in Fig. 5.3. A mixed order interpolatory $H(curl)$ (1-form) basis set from [78] can be written as:

$$\begin{aligned}
\vec{w}^{(1)}_{(u);ijk} &= B^p_i(v)B^p_j(w)G^{(p-1)}_k(u)\nabla u \\
\vec{w}^{(1)}_{(v);ijk} &= B^p_i(w)B^p_j(u)G^{(p-1)}_k(v)\nabla v \\
\vec{w}^{(1)}_{(w);ijk} &= B^p_i(u)B^p_j(v)G^{(p-1)}_k(w)\nabla w \\
&\qquad i,j = 0,\ldots,p; \quad k = 0,\ldots,p-1,
\end{aligned} \tag{5.14}$$

yielding $3p(p+1)^2$ basis functions. Note how the v- and w-directed basis functions can be obtained by cyclically permuting the variables of the

u-directed functions. As with the tetrahedral mixed order elements, the polynomial order of the tangential component is one order lower than the normal components, but without the need for mathematical manipulation to see that. Note also the use of ∇u instead of $\hat{u}$ as a basis vector. Since we are only working with Cartesian elements[20], it would be tempting to simply use the Cartesian basis vectors. Indeed, ∇u is colinear with $\hat{u}$, but scales differently under coordinate transforms. This is neccessary to ensure the proper contravarient transform under the change of variables from the reference element to the global coordinates.

The basis functions in (5.14) can be split into edge, face and volume degree of freedom groups by considering how many of the i, j or k indices are at the endpoints of the interval. Explicit groupings for the 1-form bases in (5.14) and the 2-form bases below (5.15) can be found in [82].

The choice of Gauss-Lobatto and Gauss-Legendre interpolation points migh seem arbitrary. However, the degree of freedom definitions required by conformance to $H(curl)$ as per [10] implies that certain of the interpolation points *must* lie on the endpoints of the interpolation domain. Were it not for this requirement, one may have constructed the whole basis using only Gauss-Legendre interpolation points. Polynomial sets that interpolate at the Gauss-Legendre points are orthogonal. Returning to the topic of diagonalisation, the diagonalisation requirement of (5.9) could then have been satisfied even using exact integration. The same degree of freedom requirements preclude the use of any other sets of orthogonal polynomials to construct diagonalised bases.

Since orthogonal bases cannot be used, (5.9) can instead be satisfied chosing quadrature rules that have evaluation points coinciding with the interpolation points. However, matters are not quite that straight forward; looking again at (5.14) one may note that the interpolation points of the u, v and w basis sets are in fact all different. However, they are also mutually orthogonal because of the orthogonal ∇u, ∇v and ∇w basis vectors. This implies that the approximate inner product only needs to orthogonalise u functions with respect to other u functions and similarly for v and w functions. This can be achieved by using different quadrature rules for the u, v and w functions.

Since the quadrature rules are constructed to coincide with the interpolation points of each basis subset, the u functions will use a tensor product of a Gauss-Legendre quadrature rule in u and Gauss-Lobatto quadrature rules in v and w. Similarly, the v functions will use a Gauss-Legendre quadrature rule in v and Gauss-Lobatto quadrature rules in w and u. We note that for mixed p'th order bases, the Gauss-Lobatto interpolation sets con-

[20]While the discussion up till now has been applicable to any hexahedral mesh, from here on forward all hexahedral meshes are considered to be regular and Cartesian; i.e. all the elements are rectangular bricks with 90 degree internal angles.

tain $p+1$ points and the Gauss-Legendre sets p points. This implies that both the Gauss-Legendre and Gauss-Lobatto quadrature rules implied by the approximate inner product are accurate to order $(2p-1)$.

Since the system matrix entries involve the integration of products of p'th order polynomial basis functions, order $2p$ accurate quadrature would be exact. Hence the mixed Gauss-Legendre and Gauss-Lobatto integration is only one order too low for exact integration. This is motivation of the particular sets of interpolation points. While other interpolation points and quadrature rules that satisfy (5.9) could be used, with one exception[21], any other combination would result in a lower order of accuracy.

$H(div)$ (2-form) elements are also required for PDE I. Mixed pth order 2-form bases using the same scheme as (5.14) can be written as

$$\begin{aligned}
\vec{w}^{(2)}_{(u);ijk} &= B^p_i(u)G^{(p-1)}_j(v)G^{(p-1)}_k(w)(\nabla u)^{-1} \\
\vec{w}^{(2)}_{(v);ijk} &= B^p_i(v)G^{(p-1)}_j(w)G^{(p-1)}_k(u)(\nabla v)^{-1} \\
\vec{w}^{(2)}_{(w);ijk} &= B^p_i(w)G^{(p-1)}_j(u)G^{(p-1)}_k(v)(\nabla w)^{-1} \\
&\qquad i = 0,\ldots,p; \quad j,k = 0,\ldots,p-1,
\end{aligned} \tag{5.15}$$

where $(\nabla u)^{-1}$ is an abuse of notation used to indicate the contravarient basis vectors. The proper expression would be $\frac{\nabla v\times\nabla w}{\nabla u\cdot(\nabla v\times\nabla w)}$ (see e.g. [81, Appendix]), but because we are assuming Cartesian basis vectors we have

$$\begin{aligned}
\nabla u &= (\tfrac{1}{\Delta x}, 0, 0) \\
\nabla v &= (0, \tfrac{1}{\Delta y}, 0) \\
\nabla w &= (0, 0, \tfrac{1}{\Delta z}),
\end{aligned} \tag{5.16}$$

where Δx, Δy and Δz are the mesh grid size. This means we have

$$\frac{\nabla v \times \nabla w}{\nabla u \cdot (\nabla v \times \nabla w)} = (\Delta x, 0, 0) \equiv (\nabla u)^{-1}. \tag{5.17}$$

Since the magnetic flux density $\vec{B}$ in only ever updated by the curl of a mixed p'th order discrete 1-form $\vec{E}$ field, the full space covered by the discrete mixed p'th order 2-form is not needed. With hierarchical tetrahedra, the non-solenoidal $\vec{B}$ basis functions can be removed [24]. Even with non-hierarchical tetrahedra, a lower order $\vec{B}$ base could be used. E.g. the curl of a mixed 2nd order tetrahedral 1-form can be represented exactly by a consistently linear tetrahedral 2-form. This relation does not hold for hexahedral bases; hence, both the $\vec{E}$ and $\vec{B}$ hexahedral bases need to be of the same mixed order.

[21] Using a $p+1$ point Gauss-Lobatto rule in all directions would work too; this is discussed subsequently.

As alluded to before, the mixed Gauss-Lobatto and Gauss-Legendre rules are not the only way to construct a diagonalised matrix. Looking at just the $\vec{w}^{(1)}_{(u);ijk}$ functions of (5.14), we note that the u dependent polynomial is of order $(p-1)$, implying that a quadrature rule that is accurate to order $(2p-2)$ in the u direction will exactly integrate the u dependence. Since the rules used are accurate to order $(2p-1)$, the u dependence is being integrated exactly! A diagonal matrix still results because the Gauss-Legendre interpolation polynomials used for the u dependency are mutually orthogonal.

Changing the integration rule to be Gauss-Lobatto in every direction, the integration is still accurate to order $(2p-1)$. Furthermore, the quadrature points still coincide with the interpolation points of the p'th order functions, while the orthogonal order $p-1$ functions are exactly integrated. In fact, the trapezoidal rule commonly used to diagonalise mixed first order hexahedra (e.g. [4]) can be seen as a 2 point Gauss-Lobatto rule. In the case of non-Cartesian hexahedra as described in [102] the use of the mixed rule would still be required to attain optimal sparsity, since the orthogonality of the Gauss-Legendre interpolation polynomials are likely to be destroyed by the varying Jacobian implied by curvilinear transforms.

The numerical dispersion accuracy of the diagonalised elements is also analysed in [78]. Higher order bases were found to exhibit comensurately higher order convergence. The dispersion relation derived for the lowest order elements were also shown to be identical to the values derived in the FDTD[22] context [6]. The eigen solutions of consistent and diagonalised elements are compared in [102], and were found to be similar. This also implies that higher order convergence is not destroyed by the diagonalisation process. It was also found that consistent elements tended to over-estimate eigenvalues (i.e. the numerical speed of light), while the opposite was true for diagonalised elements. This is similar to the behaviour observed by the computational structural dynamics community [13, §9.1.4]. Results obtained by the present author in the process of developing the current FEM code coroborated those findings.

A difference between the hexahedral and tetrahedral bases that is important in the context of the hybrid mesh construction in §8 are the highest order monomial terms present in the respective bases. A tetrahedral base that is complete to p'th order contains *no* monomials with order higher than p. The hexahedral basis are different because of their tensor product construction. For instance, the tensor product of two linear basis sets is *complete* only to first order, but does contain some 2nd order monomials. A mixed p'th order 1-form tensor product base will contain terms up to order $3p-1$.

[22] The relation between the FDTD and diagonalised FEM will be discussed subsequently.

5.2.4 Construction of the Curl Matrix

The curl matrix (4.49) is the discrete curl operator that takes the curl of a discrete 1-form and expresses it as a discrete 2-form. Given a 1-form bases, it is known that 2-form basis defined on the same mesh of sufficient order can exactly represent the curl of the 1-form. The entries of $[C]$ are however unique to each particular combination of discrete 1-form and 2-from basis.

It is of practical importance to know which 1-form degrees of freedom interact with which 2-form degrees of freedom through $[C]$. One application where this is particularly important is the implementation of the Total Field / Scattered Field formulation §6.1. First recall that discrete 1-forms can have edge, face and volume degrees of freedom while discrete 2-forms can have only face and volume degrees of freedom. In the low-order case the entries of $[C]$ have a very simple physical interpretation in terms of Stoke's theorem as, where the line integral of the 1-form edge degrees of freedom around the face sum to the flux through the face. In this case all the non-zero $[C]$ entries are 1 or -1.

In general, 2-form face degrees of freedom receive curl contributions from the 1-form edge degrees of freedom that make up the face boundary and also from 1-form degrees of freedom on the same face. Similarly, 2-form volume DOFs receive contributions from the 1-form faces that surround it and from 1-form DOFs defined in the same volume. While it would be possible to derive these relations analytically, it is convenient to solve an element local projection problem. Using (4.66), we can solve

$$[C] = [M_{\mu^{-1}}]^{-1}[P_{\mu_{-1}}]^T. \tag{5.18}$$

By the fact that $[C]$ does not depend on the material properties, and the fact that it is metric independent, and therefore exactly the same on any element, we can define the element-local matrices

$$[M_l] = \int_{\hat{K}} \vec{w}_i^{(2)} \cdot \vec{w}_j^{(2)}\, dV \tag{5.19}$$

$$[P_l] = \int_{\hat{K}} \nabla \times \vec{w}_i^{(1)} \cdot \vec{w}_j^{(2)}\, dV, \tag{5.20}$$

where i and j only run over the basis functions defined on the reference element. The resulting local curl matrix

$$[C_l] = [M_l]^{-1}[P_l]^T \tag{5.21}$$

can then be used to trivially construct the global $[C]$.

If hierarchical bases are used, the relations can be somewhat simplified if matched 1-form and 2-form bases are used. The 1-form [8] and 2-form [77] bases used here are such a set. The solenoidal basis functions in the discrete 2-form are all defined as the curl of 1-form basis functions. One thing to note

is that only the lowest order 1-form edge functions are rotational, and their curls are exactly represented by the lowest order 2-form face functions. For all the higher order bases, there are no cross-entity curl contributions, i.e. 1-form face functions interact only with 2-form face functions and similar for volumes. Furtheremore, the solenoidal 2-form functions are defined as the curl of a particular 1-form function, hence that is the only function with which it interacts. For such bases on tetrahedra, each low-order 2-form face function contributes three ± 1 entries to the $[C]$ matrix relating it to the three surrounding low order 1-form edge functions. Further solenoidal 2-form functions contribute a single 1 entry relating it to the 1-form function of which it is the curl.

5.3 Equivalence between FEM and FDTD

The equivalence between the FEM and the FDTD can be shown for both the PDE I and PDE II cases. By noting the relation between Newmark-$\beta = 0$ and leapfrog time-integration shown in §7.3, it is in fact only neccesary to study the equivalence for PDE I. All sources in the literature that the current author has seen proceed by diagonalising the $[M_\epsilon]$ and $[M_{\mu^{-1}}]$ matrices using trapezoidal integration. This essentially entails sampling the basis function values at the nodes of the hexahedral elements. While this approach generates the correct equivalence, it seems somewhat unintuitive, since the fields are not sampled at the same positions as they would have been in the FDTD.

Using instead the mixed Gauss-Lobatto reduced integration described in §5.2.3, the result is more satisfying. As previously noted, the trapezoidal rule is the same as the lowest order Lobatto rule, and applying the mixed rule or the all Lobatto rule yield the same result. If the mixed Gauss-Lobatto rule is used, the numerical integration points fall on the same places where the Yee FDTD cell has its degrees of freedom. The Gauss-Lobatto points of the electric degrees of freedom fall in the middle of the element's edges, and the points of the magnetic degrees of freedom fall in the element's face centers.

5.4 Program Implementation

This section discusses some topics concerning the practical implementation of the subsequently described numerical methods as a computer program. No attempt is made at presenting a detailed description of the code developed, rather the choice of implementation environment and software methodologies and their implications is discussed at a rather high level.

5.4.1 Python for Numerical Computation

The numerical methods described in this thesis was implemented using the interpreted very high level language (VHLL) Python [17]. Python is a modern object orientated VHLL featuring exceptionally clear syntax and powerful language features. It is both easy to learn, and a very productive environment for expert users. Features like automatic memory management, high level data types and extensive, freely available software libraries allow very rapid development, and much shorter code than that of an equivalent program written in e.g. C++ or Fortran. This combined with its free and open source licencing makes Python attractive for most any task. An excelent overview and motivation for the use of Python in scientific computing is [19].

At first glance using an interpreted VHLL language may seem unattractive for numerical computation, since its execution speed may be too slow. However, most VHLLs can interface with compiled libraries for speed critical loops, while some provide additional means of speeding up critical code sections.

Python has always had a C language API, allowing the extension of Python modules such that it is transparent (i.e. appears to be part of the Python environment) to the Python user, at the expense of extra work for the C programmer. More recently, automatic Python wrapper generators such as SWIG [108] and F2PY [109] have made it quite painless to extend Python using new or existing code written in compiled languages such as C, C++ and Fortran.

Python's ability to wrap existing code led to standard computational libraries such as LAPACK [110] and FFTPACK [111] along with fast array and matrix handling becoming available through the Numerical Python [112] package for Python. The Numerical Python work has since been merged and extended [18] by the SciPy[23] [20] project. An active developer and user community has grown around SciPy. The inclusion of iterative and direct sparse matrix solvers libraries in SciPy makes it well suited for FEM codes.

The sections of code that dominate the CPU time in the results described subsequently is

- The solution of the matrix equation arrising out of the implicit system at every time-step implied by (7.8)
- The matrix-vector and vector-vector products at every time-step implied by (7.9).

Both these operations are performed using compiled libraries through the

[23]Building on SciPy, the iPython [113] interactive shell and the Matplotlib [114] Pylab mode make for comfortable interactive use in the style of MATLAB (R), while keeping the full capabilities of a general language and its libraries available.

SciPy interface, and would not benefit much from implementation in a compiled language such as C.

5.4.2 Development Methodology

The purpose of Computational Electromagnetics is the numerical solution of EM problems, but the end product of CEM development itself is software. The software used to generate the subsequent results was developed largely from scratch by the current author through the course of the research presented here. While a large benefit was derived from the wealth of libraries and ease of implementation provided by Python and also the re-use of the tetrahedral mesh-handling routines of the existing FD FEM focused eMAGUS code [115], it nonetheless presented a significant software engineering challenge for a single developer given the time constraints of a PhD. Sound software engineering practice is often overlooked in the persuit of implementing new numerical techniques, but the time initially saved and more is usually lost to debugging obscure problems and dealing with code-tangles. Three techniques that were found beneficial are

- the judicious use of object oriented programming (OOP) [116] techniques, supported by Python's built-in OOP features
- the use of test driven development (TDD) [117] techniques
- the use of a revision control [118] system.

Test Driven Development (TDD) is a relatively young technique that is rapidly gaining popularity. While testing has always been part of software development, it is usually seen an additional step. In TDD, testing is an integral part of the design and coding process.

The basis of TDD is to develop fine grained automated unit tests [119] before writing the actual implementation. Test development thus forms part of the design process, often preventing design mistakes at the earliest possible moment. Implementation code is written and the tests continually re-run until the tests are satisfied. Higher level functional tests can also be added as the need arises.

Tests and new code are usually added in an incremental fashion, interspersed with code and design refactoring. Refactoring is critical to keep code manageable as its complexity grows, but carries the risk of introducing new bugs. Unit tests are invaluable when refactoring since they show up new coding errors almost as soon as they occur.

TDD may seem inefficient since it introduces the extra work of developing tests, but the efficiencies gained from the reduction in subtle bugs, improved design and ease of refactoring handsomely repays the time invested in writing tests. Since test development is part of the design process, the amount of time spent doing traditional design is also reduced.

To be effective, both OOP and TDD depend critically on language or tool-chain support. While it is possible use OOP concepts in a language with no specific support, it would be error prone and involve much extra work. Similarly TDD is infeasible without tools that automate the tests and make it easy to locate test failures; the nosetests [120] testing framework was found most useful in this regard.

Revision control systems have many benefits for software development, even when only a single developer is working on a code. Some benefits that became apparent while undertaking the current work are:

- The ability to see exactly what code has changed in a given development cycle. This both aids understaning and often makes mistakes apparent.
- Accidental edits to unrelated code are easily noticed and reverted.
- The availability of a complete code history helps to track progress, and is also invaluable in tracking down exaclty what change caused a new bug.
- Makes it almost impossible to loose code through accidental deletion or overwiting.
- The ability to easily experiment on a different copy of the code.

The GNU Arch [121] version control system was used, although the newer and easier to use Bazaar-ng [122] system may be more appropriate for new projects.

5.5 Conclusion

Some aspects of the numerical implementation of the more abstract FEM theory presented in §4 have been discussed. General FEM characteristics related to basis function support, DOF defintions, data structures, inner product evaluation and their impact of the global FEM system matrices have been described. The concrete basis function sets used to implement the FEM formulations used in this thesis have been presented, their characteristics described, and the practical implications of their characteristics discussed. The equivalences that exist between the FEM and the FDTD have been discussed. Finally, some issues surrounding the software methodologies and tools used for the practical implementation of FEM methods have been discussed.

Chapter 6

Peripheral FEM Matters

6.1 Total Field/Scattered Field Formulation for Waveguide Analysis

The Total Field/Scattered Field (TF/SF) formulation is most commonly used in FDTD codes with a plane-wave source-condition to perform RADAR cross section (RCS) scattering calculations. This application is described at length in [6]. Application to waveguiding problems are less commonly discussed. In this section, the basic FDTD-like description of the TF/SF formulation for waveguiding problems is given. Subsequently it is cast in a form amenable to operation in a FEM framework based on PDE I, and its practical implementation discussed.

6.1.1 Basic TF/SF Formulation

The basic TF/SF concept is demonstrated with a 1-D example in Fig. 6.1. This example assumes the familiar 1-D FDTD discretisation of a transmission line [6, §3.4]. This configuration has only a single TF/SF interface Γ_{ts}; this is akin to the measurement of a circuit S_{11} parameter where Γ_{ts} defines the port. S_{11} is determined by launching a purely $+\hat{z}$-travelling wave at Γ_{ts} and then recording the $-\hat{z}$ reflected, i.e. scattered wave that returns [123]. In other words, a numerical source is sought that will 1) launch only $+\hat{z}$ travelling waves and 2) be transparent, i.e. matched to returning $-\hat{z}$ travelling waves. Creating such a source is not entirely trivial.

Numerical wave-propagation occurs only through the interaction of neighbouring $\vec{E}$ and $\vec{B}$ degrees of freedom. If an $\vec{E}$ field is simply imposed at $z = z_0 = z_\Gamma$, $\vec{E}_0$ "pushes" the two neighbouring $\vec{H}$ degrees of freedom at $z = z_{\pm\frac{1}{2}}$. They in turn push their neighboring $\vec{E}$ degrees of freedom, etc[1].

[1]Of course this implies that at the half time-step directly after e.g. $\vec{H}_{\frac{1}{2}}$ has pushed $\vec{E}_1$, $\vec{E}_1$ pushes back at $\vec{H}_{\frac{1}{2}}$. This does not cause an $+\hat{z}$ travelling wave since the the push

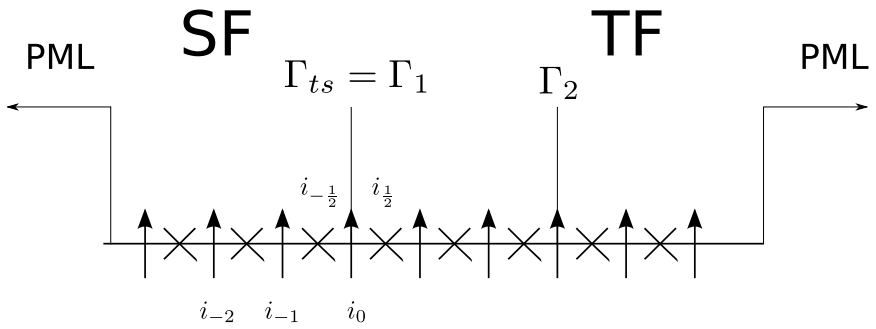


Figure 6.1: One dimensional total field/scattered field geometry. $\vec{E}$ degrees of freedom are represented by arrows and $\vec{H}$ degrees of freedom by crosses.

Not only does imposing $\vec{E}$ at Γ_{ts} result in waves travelling in both directions, it also acts as a short circuit to returning waves.

The TF/SF formulation works by using the linearity of Maxwell's equation to split $\vec{E}$ and $\vec{H}$ fields into incident fields that are assumed to be known, and unknown scattered fields. Then we have

$$\vec{E}_{\text{tot}} = \vec{E}_{\text{inc}} + \vec{E}_{\text{scat}}, \qquad \vec{H}_{\text{tot}} = \vec{H}_{\text{inc}} + \vec{H}_{\text{scat}}, \tag{6.1}$$

where the tot, inc and scat footnotes refer respectively to total, incident and scattered field quantities. Instead of imposing $\vec{E}$ rigidly, the known $\vec{E}_{\text{inc}}$ is *added* to $\vec{E}_{\text{tot},0}$ at Γ_{ts}. The $\vec{H}_{tot,\frac{1}{2}}$ DOF to its immediate right is in the total field region, and is updated as normal using the stored total field $\vec{E}_0$. The $\vec{H}_{\text{scat},-\frac{1}{2}}$ DOF to the left of Γ_{ts} is in the scattered field region and is updated using the both stored $\vec{E}_{\text{tot},0}$ and the known $\vec{E}_{\text{inc},0}$, i.e. by $(\vec{E}_{\text{tot},0} - \vec{E}_{\text{inc},0})$. At the next half time-step, $\vec{E}_{\text{tot},0}$ is updated like normal by $\vec{H}_{tot,\frac{1}{2}}$. The stored value at $\vec{H}_{\text{scat},-\frac{1}{2}}$ is a scattered field value; to update $\vec{E}_{\text{tot},0}$ consistently, the known $\vec{H}_{\text{inc},-\frac{1}{2}}$ value is added, hence $\vec{E}_{\text{tot},0}$ is updated using $(\vec{H}_{\text{scat},-\frac{1}{2}} + \vec{H}_{\text{inc},-\frac{1}{2}})$.

In the absence of any reflections returing from the total field region, the result is a purely $+\hat{z}$ travelling wave in the total field region and a null field in the scattered field region. If a $-\hat{z}$ travelling reflected wave impinges at Γ_{ts}, the stored value $\vec{E}_{\text{tot},0}$ now contains a non-zero scattered value $\vec{E}_{\text{scat},0}$. Since the known $\vec{E}_{\text{inc},0}$ is identically removed from $\vec{H}_{\text{scat},-\frac{1}{2}}$'s perspective, the scattered field region "sees" only $\vec{E}_{\text{scat},0}$. The wave scattered

given by $\vec{E}_1$ is to $\vec{H}_{\frac{1}{2}}$ equal and opposite to the push earlier received by $\vec{H}_{\frac{1}{2}}$ from $\vec{E}_0$.

from an obstacle in the total field region thus enters the scattered field region and propagates in the $-\hat{z}$ direction until it reaches the PML region and is absorbed. This scattered field is exactly the quantity required for an S_{11} measurement for port-1 defined on $\Gamma_1 = \Gamma_{ts}$. If the total field region is also terminated with the PML, it is equivalent to a matched-load, since any $+\hat{z}$-travelling wave impinging on the PML region is absorbed, resulting in no reflected $-\hat{z}$-travelling wave coming back. If port-2 is defined on Γ_2 in the total field region, the measurement made there is the quantity required for an S_{21} measurement [123].

6.1.2 Sourcing the Incident Field

An important requirement for making a TF/SF formulation operational is to have a priori knowledge of the incident field waveform. In the 1-D case described in §6.1.1 it is quite simple to see that $\vec{E}_{\text{inc}}$ needs to be known at $\vec{E}_0$ and $\vec{H}_{-\frac{1}{2}}$. It is the curls of these fields that update each other. In 3-D, Γ_{ts} is constructed such that it coincides with discrete $\vec{E}$ degrees of freedom tangential to it. Therefore, the tangential values of $\vec{E}_{\text{inc}}$ on Γ_{ts} must be known. Furthermore, the values of $\vec{H}_{\text{inc}}$ must be known for all the $\vec{H}$ DOFs in the scattered region that are updated by the curl of the $\vec{E}$ DOFs lying in Γ_{ts}. Since the curl relation between the $\vec{E}$ and $\vec{H}$ DOFs on an FDTD grid are reciprocal, the $\vec{H}$ DOFs updated by $\vec{E}$ lying on Γ_{sc} are also only the $\vec{H}$ DOFs in the scattered field region that update $\vec{E}_{\text{tot}}$.

We have used the $\vec{H}$ field in the discussion this far, but in the PDE I semi-discretisation defined in §4.3.4 the $\vec{B}$ flux density is discretised. Now we define $\{b_{ts}\}$ as the vector of $\vec{B}$ DOFs that are updated by $\{e_{ts}\}$, the vector of $\vec{E}$ DOFs located in Γ_{ts}. Given the $\{e_{ts}\}$ DOFs, it is quite simple to find the required $\{b_{ts}\}$ DOFs in the 3-D FDTD, since only neighbouring DOFs interact with each other. In the general FEM case this is somewhat more complicated. The curl interactions in the FEM is described by the curl matrix $[C]$ defined in (4.49). The specific DOF interactions that $[C]$ make operational are described in §5.2.4. These interactions imply that $\{b_{ts}\}$ is made up of:

- All the $\vec{B}$ face-dofs in the scattered-field region that share an edge with any $\vec{E}$ dof in $\{e_{ts}\}$.
- All the $\vec{B}$ volume-dofs in the scattered-field region that share an element with any of the face-dofs above.

If closed-form time-domain expressions of $\vec{E}_{\text{inc}}$ and $\vec{B}_{\text{inc}}$ are known, obtaining the incident field DOF values at each time-step is a fairly simple question of interpolating the known incident field values using the discrete intepolation operators. For non-TEM guided waves this is not always the case; obtaining the incident values for waveguide modes are considered subsequently.

6.1.3 Generating Incident Field Values for Waveguide Modes

While closed-form expressions for transient waveguide fields are known [124], the numerical evaluation of the transient fields due to general driving waveforms is somewhat involved and in fact quite computationally expensive [24]. Determining the values of $\vec{E}$ on a plane of constant phase is quite easy, but the dispersive propagation properties of waveguide modes make the determination of $\vec{B}$ away from the plane problematic. An alternative approach from the FDTD literature is bootstrapping, see e.g. [7]. A small section of waveguide is simulated independently of the main problem. The $\{e_{ts}\}$ dofs on a waveguide cross-section Γ_{inc} is logged. Informed by the discussion in the previous subsection, the $\{b_{ts}\}$ DOFs in the elements on the source side of Γ_{inc} are also logged.

The only requirement on the source simulation[2] is that the mesh and DOF definitions on Γ_{inc} and in the elements connected to Γ_{inc} on the source side are identical to those on Γ_{ts} and the elements on the scattered-field side of Γ_{ts}. This in fact provides an accuracy advantage over the matching of analytically known fields, since the numerical dispersion of the discrete system is accounted for automatically. The bootstrap system is excited by a Direchlet hard-source at one end and is terminated in a PML at the other. Since the PML prevents reflected $-\hat{z}$ travelling waves, the zero-impedance of the hard-source is immaterial.

The Direchlet source is constructed by interpolating the incident waveguide mode $\vec{E}$ field over the Direclet surface. This can be done by applying an interpolation operator to the analytical incident mode description (if it is know) or by solving a 2-D eigen-problem over the waveguide cross section, e.g. [45, 73]. The tangential trace of the 3-D $\vec{E}$- and the normal trace of the 3-D $\vec{B}$ basis functions are used to construct the 2-D eigen-system. The interpolation approach is used subsequently, simply because it was easier to implement in software. The Direchlet contribution to the RHS in (4.75) is formed by multiplying the interpolated incident DOFs by a time-waveform.

6.1.4 Exciting the Problem Geometry

The problem system and bootstrap system will obviously have different global DOF numberings. This is accounted for by introducing the permutation matrices $[I^e_{ts}]$ and$[I^b_{ts}]$ that permutes the logged $\vec{e_{ts}}$ and $\vec{b_{ts}}$ values to their respective global DOF numbers in the problem system. The update equations can now be written as:

$$\{b\}^{n+\frac{1}{2}} = \{b\}^{n-\frac{1}{2}} - \Delta t\Big([C]\big(\{e\}^n + [I^e_{ts}]\{e_{ts}\}^n\big)\Big), \tag{6.2}$$

[2]Besides it being accurate, of course.

$$\{e\}^{n+1} = \{e\}^{n+1} + \Delta t[M_\epsilon]^{-1}\Big([C]^T[M_{\mu^{-1}}]\big(\{b\}^{n+\frac{1}{2}} + [I^b_{ts}]\{b_{ts}\}^{n+\frac{1}{2}}\big)\Big). \quad (6.3)$$

Simulation of the actual problem geometry can be done fairly simply by setting it up as an homogenous, i.e. source-free system. After each standard $\{b\}$ update, the incident field contribution due to $\{e_{ts}\}$ is added, and similarly the incident $\{b_{ts}\}$ contribution is added after each $\{e\}$ update.

6.2 Extracting Waveguide Results

Usually waveguides are operated at frequencies such that only one dominate waveguide mode is propagating. Higher order modes are generated by inhomogeneities in the waveguiding structure, but since the higher order modes are evanescent they die down away from the inhomogeneities. While waveguide circuit components utilise the higher order modes to perform their functions, only the effects on the dominant propagating mode is to be measured. When extracting waveguide results, the field contributions from evanescent modes have to be discarded.

Due to the orthogonality properties of waveguide modes, the desired mode coefficient can be extracted by integrating over the waveguide cross-section [21]. For a TE mode the integral

$$g = \frac{\int_{\Gamma_p}(\hat{n} \times \vec{E}) \cdot (\hat{n} \times \vec{E}_m)}{\int_{\Gamma_p}(\hat{n} \times \vec{E}_m) \cdot (\hat{n} \times \vec{E}_m)} \quad (6.4)$$

would be used, where g is the mode coefficient, Γ_p the waveguide port surface and $\vec{E}_m$ is the field of the waveguide mode.

6.3 Convolution-free Explicit PML

All PDE based methods such as the FEM considered here inherently deal with finite regions of space only. Whenever an unbounded domain is modelled some means of truncating the computational domain must be used. A good truncation scheme should absorb all outgoing waves without reflection. It should also allow the truncation boundary to be close to the problem geometry to minimise the computational burden. The introduction of the Perfectly Matched Layer (PML) by Berenger [125] caused quite a stir when it was introduced in 1994. It has since been developed to such an extent that for the FDTD, domain truncation is largely a solved problem [5]. The application of the PML to the time-domain FEM has not been as straight forward.

Time-domain implementations of the PML based on PDE II have been presented [126], [127], [128]. Apart from being formidably complicated to

implement, late time instabilities have been reported [126]. While stability can be assured long enough to solve problems of practical interest, this form of the PML is also quite computationally expensive due to being based on an implicit FEM formulation. Implementation for PDE I finite element methods is more straight forward [129], but in general requires the storage of several auxiliary variables to avoid the need for numerical time-convolution.

In this section, a relatively simple implementation of a convolution-free PML for the explicit PDE I is developed along the lines of the popular FDTD uni-axial PML (UPML) implementation [6, §7.8]. For application to FEM discretisations, the FDTD method is cast into a discrete differential forms representation. This representation is analysed and the assumptions that make it operable for the FDTD is found to hold for diagonalised Cartesian hexahedra. This analysis also sheds some light on the need of PML conductivity grading in numerical implementations.

6.3.1 Uni-axial PML Description

The UPML method describes the absorber as a dispersive, electrically and magnetically lossy, anisotropic material with a characteristic impedance that is matched to its surroundings. The frequency-domain Maxwell's equations in a UPML region matched to free-space is written as:

$$\nabla \times \vec{H} = j\omega\mu \overline{\overline{s}} \vec{E}, \tag{6.5}$$

$$\nabla \times \vec{E} = -j\omega\epsilon \overline{\overline{s}} \vec{H}, \tag{6.6}$$

where

$$\overline{\overline{s}} = \begin{bmatrix} \frac{s_y s_z}{s_x} & & \\ & \frac{s_x s_z}{s_y} & \\ & & \frac{s_x s_y}{s_z} \end{bmatrix}, \tag{6.7}$$

and

$$s_x = \kappa_x + \frac{\sigma_x}{j\omega\epsilon_0}, \quad s_y = \kappa_y + \frac{\sigma_y}{j\omega\epsilon_0}, \quad s_z = \kappa_z + \frac{\sigma_z}{j\omega\epsilon_0}. \tag{6.8}$$

κ_{xyz} are stretching factors that makes the medium electrically longer for values larger than 1[3]. σ_{xyz} are electric or magnetic conductivities[4] that cause the fields inside the PML region to attenuate. Setting $\sigma_x > 0$ causes the attenuation of waves travelling in the $\hat{x}$ direction and similarly for σ_y

[3] The numerical results in §10 used $\kappa_{xyz} = 1$.

[4] Applied to the $\vec{E}$ field, σ_{xyz} is an electric conductivity and magnetic when applied to $\vec{H}$.

and σ_z. Note that the same $\overline{\overline{s}}$ is used for both (6.5) and (6.6) in order to match the medium to free-space. We define for subsequent use:

$$\overline{\overline{s}}_x = \begin{bmatrix} s_x & & \\ & s_y & \\ & & s_z \end{bmatrix}, \quad \overline{\overline{s}}_y = \begin{bmatrix} s_y & & \\ & s_z & \\ & & s_x \end{bmatrix}, \quad \overline{\overline{s}}_x = \begin{bmatrix} s_z & & \\ & s_x & \\ & & s_y \end{bmatrix}. \tag{6.9}$$

Since all the matrices are diagonal, we can write:

$$\overline{\overline{s}} = \overline{\overline{s}}_x^{-1}\overline{\overline{s}}_y\overline{\overline{s}}_z = \overline{\overline{s}}_y\overline{\overline{s}}_x^{-1}\overline{\overline{s}}_z = \overline{\overline{s}}_y\overline{\overline{s}}_z\overline{\overline{s}}_x^{-1} \ldots \tag{6.10}$$

Note that while s_x describes the behaviour of the PML in the $\hat{x}$ direction, $\overline{\overline{s}}_x$ has no particular physical meaning. It will however prove useful in the subsequent analysis. We also define

$$\overline{\overline{\sigma}}_x = \begin{bmatrix} \sigma_x & & \\ & \sigma_y & \\ & & \sigma_z \end{bmatrix}, \quad \overline{\overline{\kappa}}_x = \begin{bmatrix} \kappa_x & & \\ & \kappa_y & \\ & & \kappa_z \end{bmatrix}, \tag{6.11}$$

and similar for σ_y, σ_z, κ_y and κ_z such that

$$\overline{\overline{s}}_x = \overline{\overline{\kappa}}_x + \frac{\overline{\overline{\sigma}}_x}{j\omega\epsilon_0}, \quad \overline{\overline{s}}_y = \overline{\overline{\kappa}}_y + \frac{\overline{\overline{\sigma}}_y}{j\omega\epsilon_0}, \quad \overline{\overline{s}}_z = \overline{\overline{\kappa}}_z + \frac{\overline{\overline{\sigma}}_z}{j\omega\epsilon_0}. \tag{6.12}$$

6.3.2 Two-step Material Operator Application

The FDTD convolution-free implementation now introduces two auxiliary fields. In the FDTD literature the physical fields are given as $\vec{E}$ and $\vec{H}$ and the auxiliary fields as $\vec{D}$ and $\vec{B}$. This is not quite accurate, since they are not the physical $\vec{D}$ and $\vec{B}$ fluxes; instead they are written $\tilde{D}$ and $\tilde{B}$ here, the tilde indicating their non-physical nature. They are defined as

$$\tilde{D} = \epsilon\overline{\overline{s}}_x^{-1}\overline{\overline{s}}_z\vec{E}, \qquad \tilde{B} = \mu\overline{\overline{s}}_x^{-1}\overline{\overline{s}}_z\vec{H}, \tag{6.13}$$

allowing the application of the material operator $\overline{\overline{s}}$ to be split into two steps. The FDTD analysis proceeds to directly substitute these relations into the Cartesian component form, leading to expressions that are trivial to transform to the time domain and have an obvious finite differences discretisation. Unfortunately it does not provide any insight into the FEM discretisation of the UPML.

A different strategy is followed here. The two step material property application allows (6.5) and (6.6) to be written as

$$j\omega\overline{\overline{s}}_y\tilde{D} = \nabla \times \vec{H}, \tag{6.14}$$

$$j\omega \overline{\overline{s}}_y \tilde{B} = -\nabla \times \vec{E}. \tag{6.15}$$

This implies

$$\vec{D} = \overline{\overline{s}}_y \tilde{D}, \quad \vec{B} = \overline{\overline{s}}_y \tilde{B}. \tag{6.16}$$

A suitable FEM discretisation can be obtained by taking a differential forms approach. Analysing (6.14), we recognise the RHS as a 2-form due to the curl operator. Furthermore, $\overline{\overline{s}}_y$ appears to be a material Hodge operator, implying that $\tilde{D}$ must be a 1-form, and similarly from (6.15) we deduce that $\tilde{B}$ must also be a 1-form. Looking at (6.13), $\overline{\overline{s}}_x^{-1}$ is taken to be a material Hodge. Furthermore, ϵ and μ have to be combined with $\overline{\overline{s}}_z$ in order to ensure that both sides of the equations are 1-form. Hence, we consider $\epsilon\overline{\overline{s}}_z$ and $\mu\overline{\overline{s}}_z$ as combined Hodge operators. Writing (6.13-6.16) as differential forms, we get:

$$\tilde{D} = \star_{s_x^{-1}} \star_{\epsilon s_z} E, \qquad \tilde{B} = \star_{s_x^{-1}} \star_{\mu s_z} B, \tag{6.17}$$

$$j\omega \star_{s_y} \tilde{D} = dH, \tag{6.18}$$

$$j\omega \star_{s_y} \tilde{B} = -dE, \tag{6.19}$$

$$D = \star_{s_y} \tilde{D} = \star_{s_y} \star_{s_x}^{-1} \star_{\epsilon s_z} = \star_{\epsilon s} E \tag{6.20}$$

$$B = \star_{s_y} \tilde{B} = \star_{s_y} \star_{s_x}^{-1} \star_{\mu s_z} \tilde{B} = \star_{\mu s} H. \tag{6.21}$$

6.3.3 Frequency-domain Discrete UPML

The standard PDE I semi-discretisation (§4.3.4) discretises $\vec{E}$ as 'n discrete 1-form and $\vec{B}$ as a discrete 2-form. With reference to §4.3.6, we note that both these quantities are defined on the primary grid. It was also noted that the discrete forms implicitly defines twisted form discretisations of $\vec{D}$ and $\vec{H}$ on the dual grid. Recall also the following:

- The discrete exterior derivative operator d operating on a primary grid 1-form (i.e. equivalent to $\nabla\times$) results in a primary grid 2-form.
- d operating on a twisted 1-form results in a twisted 2-form.
- Hodge $\star$ operators define an isomorphism between between primary 1-forms and twisted 2-forms or twisted 1-forms and primary 2-forms. In other words, it maps between primary grid and dual grid forms.

Considering the choice of $\vec{E}$ and $\vec{B}$ as the primary grid working variables, the relations in (6.17-6.21) imply that the discrete $\tilde{D}$ must be a primary grid

1-form and $\tilde{B}$ a twisted 1-form. Written in discrete forms notation, (6.17-6.21) become:

$$\{\tilde{d}\} = [\star_{s_x}]^{-1}[\star_{\epsilon s_z}]\{e\}, \tag{6.22}$$

$$\{\tilde{b}^\dagger\} = [\star^\dagger_{s_x}]^{-1}[\star^\dagger_{\mu s_z}]\{h^\dagger\}, \tag{6.23}$$

$$j\omega[\star_{s_y}]\{\tilde{d}\} = [C]^T\{h^\dagger\}, \tag{6.24}$$

$$j\omega[\star^\dagger_{s_y}]\{\tilde{b^\dagger}\} = -[C]\{e\}, \tag{6.25}$$

$$\{d^\dagger\} = [\star_{s_y}]\{\tilde{d}\}, \quad \{b\} = [\star^\dagger_{s_y}]\{\tilde{b}\}, \tag{6.26}$$

where the dagger superscript indicates twisted form quantities. Note also that in the continuum case that the same Hodges, e.g. $\star_{s_x}$, operate on both electric and magnetic quantities. However, in the discrete case there are separate Hodges defined for either primary grid or twisted forms.

Calculating the entries of the primary grid discrete Hodge operators is quite straight forward using the normal Galerkin Hodge process:

$$[\star_{\epsilon s_z}]_{ij} = \int_\Omega \vec{w}^{(1)}_i \cdot \epsilon \bar{\bar{s}}_z \vec{w}^{(1)}_j \, d\Omega, \tag{6.27}$$

and similarly for $[\star_{s_y}]$ and $[\star_z]$, where $\vec{w}^{(1)}_i$ are the primary grid 1-form basis functions used to expand the $\vec{E}$ field. Calculating the entries for the twisted form Hodge operators is not obvious, since basis function expansions of the twisted forms are unknown. They can however be defined quite simply by using Galerkin duality [59, 60]:

$$[\star^\dagger_{\mu s_z}] = [\star_{\mu^{-1} s_z^{-1}}]^{-1}, \tag{6.28}$$

with

$$[\star_{\mu^{-1} s_z^{-1}}]_{ij} = \int_\Omega \vec{w}^{(2)}_i \cdot \mu^{-1} \bar{\bar{s}}_z^{-1} \vec{w}^{(2)}_j \, d\Omega, \tag{6.29}$$

where $\vec{w}^{(2)}_i$ are the primary grid 2-form basis functions used to expand $\vec{B}$. Now we can rewrite (6.23) as

$$\{\tilde{b}^\dagger\} = [\star_{s_x^{-1}}][\star_{\mu^{-1} s_z^{-1}}]^{-1}\{h^\dagger\}, \tag{6.30}$$

and similarly for (6.25).

6.3.4 Analysis of Discrete Frequency-domain UPML

The standard frequency-domain FEM discretisation of (6.20) and (6.21), is

$$\{d^\dagger\} = [\star_{\epsilon s}]\{e\} \tag{6.31}$$

$$\{b\} = [\star_{\mu^{-1}s^{-1}}]^{-1}\{h^\dagger\}, \tag{6.32}$$

where

$$[\star_{\epsilon s}]_{ij} = \int_\Omega \vec{w}_i^{(1)} \cdot \epsilon \overline{\overline{s}} \vec{w}_j^{(1)}\, d\Omega, \tag{6.33}$$

$$[\star_{\mu^{-1}s^{-1}}]_{ij} = \int_\Omega \vec{w}_i^{(2)} \cdot \mu^{-1} \overline{\overline{s}}^{-1} \vec{w}_j^{(2)}\, d\Omega. \tag{6.34}$$

The proposed two-step discretisation implies instead:

$$\{d^\dagger\} = [\star_{s_y}][\star_{s_x}]^{-1}[\star_{\epsilon s_z}]\{e\} \tag{6.35}$$

$$\{b\} = [\star_{s_y^{-1}}]^{-1}[\star_{s_x^{-1}}][\star_{\mu^{-1}s_z^{-1}}]^{-1}\{h^\dagger\}. \tag{6.36}$$

It is not immediately clear if (6.35) and (6.36) represents a reasonable approach. Instead it will be shown that (6.35) and (6.36) are equivalent to (6.31) and (6.32) when the diagonalised Cartesian hexahedra described in §5.2.3 are used.

Assuming initially that $\overline{\overline{s}}$, ϵ and are constant, we can write[5]

$$[\star_{\epsilon s}] = \frac{\epsilon s_y s_z}{s_x}[L_x^{(1)}] + \frac{\epsilon s_z s_x}{s_y}[L_y^{(1)}] + \frac{\epsilon s_x s_y}{s_z}[L_z^{(1)}] \tag{6.37}$$

and

$$[\star_{\mu^{-1}s^{-1}}] = \frac{s_x}{\mu s_y s_z}[L_x^{(2)}] + \frac{s_y}{\mu s_z s_x}[L_y^{(2)}] + \frac{s_z}{\mu s_x s_y}[L_z^{(2)}], \tag{6.38}$$

where the $[L_u]$ matrices represent the metric of the basis functions in the $\hat{u}$ direction. Their entries are

$$[L_u^{(l)}]_{ij} = \int_\Omega (\hat{u} \cdot \vec{w}_i^{(l)}) \cdot (\hat{u} \cdot \vec{w}_j^{(l)})\, d\Omega. \tag{6.39}$$

The separate sub-operator matrices defined by (6.27-6.29) can be similarly decomposed by using the $[L_u]$ matrices. It is then quite clear that

$$[\star_{s_y}][\star_{s_x}]^{-1}[\star_{\epsilon s_z}] \neq [\star_{\epsilon s}] \tag{6.40}$$

and

$$[\star_{s_y^{-1}}]^{-1}[\star_{s_x^{-1}}][\star_{\mu^{-1}s_z^{-1}}]^{-1} \neq [\star_{\mu^{-1}s^{-1}}]^{-1} \tag{6.41}$$

even if diagonalised $[L_u]$ matrices are used.

[5]Note, here s_x, s_y and s_z are the quantities in (6.8), not to be confused with $\star_{s_x}$, etc. that are the Hodge operators associated with the sub-operators in (6.9).

However, the $[L_u]$ matrices for Cartesian hexahedra have additional structure since the basis functions are defined in separate $\hat{x}$, $\hat{y}$ and $\hat{z}$ directed groups. Assuming the global degrees of freedom are numbered such that the $\hat{x}$, $\hat{y}$ and $\hat{z}$ directed groups are numbered in order, we have

$$[L_x^l] = \begin{bmatrix} [L_{xx}^l] & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; [L_y^l] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & [L_{yy}^l] & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; [L_z^l] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & [L_{zz}^l] \end{bmatrix}. \tag{6.42}$$

The sub-operators can now be written as

$$[\star_{\epsilon s_z}] = \epsilon s_z [L_{xx}^{(1)}] \oplus \epsilon s_x [L_{yy}^{(1)}] \oplus \epsilon s_z [L_{zz}^{(1)}], \tag{6.43}$$

$$[\star_{\mu^{-1} s_z^{-1}}] = \mu^{-1} s_z^{-1} [L_{xx}^{(2)}] \oplus \mu^{-1} s_x^{-1} [L_{yy}^{(2)}] \oplus \mu^{-1} s_z^{-1} [L_{zz}^{(2)}], \tag{6.44}$$

and similarly for the other sub-operators.

Using (6.43), we can write

$$\begin{aligned} [\star_{s_z}][\star_{s_x}]^{-1}[\star_{\epsilon s_z}] = & \left(s_y [L_{xx}^{(1)}] \frac{1}{s_x} [L_{xx}^{(1)}]^{-1} \epsilon s_z [L_{xx}^{(1)}] \right) \oplus \\ & \left(s_z [L_{yy}^{(1)}] \frac{1}{s_y} [L_{yy}^{(1)}]^{-1} \epsilon s_x [L_{yy}^{(1)}] \right) \oplus \\ & \left(s_x [L_{zz}^{(1)}] \frac{1}{s_z} [L_{zz}^{(1)}]^{-1} \epsilon s_y [L_{zz}^{(1)}] \right) \\ = & \frac{s_y s_z}{s_x} [L_{xx}^{(1)}] \oplus \frac{s_z s_x}{s_y} [L_{yy}^{(1)}] \oplus \frac{s_x s_z}{s_z} [L_{zz}^{(1)}] \\ = & [\star_{\epsilon s}]. \end{aligned} \tag{6.45}$$

Similarly, we can show that

$$[\star_{s_y^{-1}}]^{-1}[\star_{s_x^{-1}}][\star_{\mu^{-1} s_z^{-1}}]^{-1} = [\star_{\mu^{-1} s^{-1}}]^{-1}. \tag{6.46}$$

So far, we have only needed the block-diagonal property of Cartesian hexehedral elements' $[L_u]$ matrices, but we have also had to assume constant material properties. The derivation of (6.45) and (6.46) depends on the material properties being scalars outside the matrices, allowing commutative multiplication. As soon as material material properties vary with position they must be included in the matrices, implying that (6.45) and (6.46) do not hold anymore. If the $[L_u]$ matrices are diagonal, their multiplication is also commutative. In other words, when using lumped Cartesian hexehedra, *the two-step material application is equivalent to the standard frequency-domain FEM.*

6.3.5 Time Domain Discrete UPML

To make the UPML operational, time-domain expressions are required. If the time-domain expressions involve only first-derivatives of time, the standard central difference leap-frog time-discretisation can be used without the need for numerical convolution. Subsequently we assume diagonalised Cartesian hexahedra are used for the spatial discretisation. We consider the case where the material properties are element-wise constant as is normal for the FEM. Because of the block-diagonal nature of the matrices, we can analyse the $\hat{x}$, $\hat{y}$ and $\hat{z}$ basis sets independently; subsequently we will *only work with the $\hat{x}$ components*, since the analysis of the other components are similar. We also introduce for convenience the dyadic notation for $\overline{\overline{s}}$ such that

$$\overline{\overline{s}} = s_{xx}\hat{x}\hat{x} + s_{yy}\hat{y}\hat{y} + s_{zz}\hat{z}\hat{z} \tag{6.47}$$

with

$$s_{xx} = \frac{s_y s_z}{s_x}, \quad s_{yy} = \frac{s_x s_z}{s_y}, \quad s_{zz} = \frac{s_x s_y}{s_z}. \tag{6.48}$$

Assuming diagonal $[L_u]$ matrices and that we are dealing only with $\hat{x}$ directed basis functions, we can write

$$[\star_{\epsilon s}]_{ii} = \sum_{K}^{\mathrm{supp}(\vec{w}_i^{(1)})} \int_K \epsilon(K) s_{xx}(K)\, \vec{w}_i^{(1)} \cdot \vec{w}_i^{(1)}, \tag{6.49}$$

where $\epsilon(K)$ and $s_{xx}(K)$ are the material parameters in element K. Since the elements K are all identical in our regular Cartesian mesh, we can simply take the average, i.e.

$$[\star_{\epsilon s}]_{ii} = \frac{\sum_{K}^{\mathrm{supp}(\vec{w}_i^{(1)})} \epsilon(K) s_{xx}(K)}{\mathrm{len}(\mathrm{supp}(\vec{w}_i^{(1)}))} [L_{xx}^{(1)}]_{ii}, \tag{6.50}$$

where $\mathrm{len}(\cdot)$ gives the number of elements. We can then define the diagonal material matrix

$$[\epsilon\overline{\overline{s}}]_{ii} = \frac{\sum_{K}^{\mathrm{supp}(\vec{w}_i^{(1)})} \epsilon(K) s_{xx}(K)}{\mathrm{len}(\mathrm{supp}(\vec{w}_i^{(1)}))}, \tag{6.51}$$

letting us write

$$[\star_{\epsilon s}] = [\epsilon\overline{\overline{s}}][L_{xx}^{(1)}]. \tag{6.52}$$

Similarly we have

$$[\mu^{-1}\overline{\overline{s}}^{-1}]_{ii} = \frac{\sum_{K}^{\mathrm{supp}(\vec{w}_i^{(2)})} \epsilon(K)^{-1} s_{xx}(K)^{-1}}{\mathrm{len}(\mathrm{supp}(\vec{w}_i^{(2)}))} \tag{6.53}$$

and

$$[\star_{\mu^{-1}s^{-1}}] = [\mu^{-1}\bar{\bar{s}}^{-1}][L^{(2)}_{xx}]. \tag{6.54}$$

Similar matrices are also defined for other material parameters, e.g.

$$[s_x]_{ii} = \frac{\sum_{K}^{\text{supp}(\vec{w}_i^{(1)})} s_x(K)}{\text{len}(\text{supp}(\vec{w}_i^{(1)}))}, \quad \text{and} \tag{6.55}$$

$$[\sigma_y]_{ii} = \frac{\sum_{K}^{\text{supp}(\vec{w}_i^{(1)})} \sigma_y(K)}{\text{len}(\text{supp}(\vec{w}_i^{(1)}))}. \tag{6.56}$$

We start with (6.30), since it poses the greatest difficulty. Repeating (6.30) for convenience, we have:

$$\{\tilde{b}^\dagger\} = [\star_{s_x^{-1}}][\star_{\mu^{-1}s_z^{-1}}]^{-1}\{h^\dagger\}. \tag{6.57}$$

Using the material matrices defined above, we have

$$\{\tilde{b}^\dagger\} = [s_x^{-1}][L^{(2)}_{xx}]\left([\mu^{-1}s_z^{-1}][L^{(2)}_{xx}]\right)^{-1}\{h^\dagger\}, \tag{6.58}$$

but since all the matrices are diagonal, we solve for $\{h^\dagger\}$ and get

$$\{h^\dagger\} = [s_x^{-1}]^{-1}[\mu^{-1}s_z^{-1}]\{\tilde{b}^\dagger\}. \tag{6.59}$$

Now we consider only a single equation defined for $\{h^\dagger\}_i$ and $\{\tilde{b}^\dagger\}$. Using the fact that discrete 2-forms can only have face or volume degrees of freedom, a given $\vec{w}_i^{(2)}$ can have support in at most 2 elements. We label the elements K_a and K_b. Now we have

$$\{h^\dagger\}_i = \frac{\frac{1}{s_x(K_a)} + \frac{1}{s_x K_b}}{\frac{1}{\mu K_a s_z(K_a)} + \frac{1}{\mu(K_b)s_z(K_b)}}\{\tilde{b}^\dagger\}. \tag{6.60}$$

This results in an expression that requires time-convolution to cast into the time-domain. If we make the following approximation:

$$\frac{1}{s(K_a)} + \frac{1}{s(K_b)} \simeq \frac{1}{(s(K_a) + s(K_b))/2}, \tag{6.61}$$

the much simpler form

$$\bar{\mu}\bar{s_z}\{h^\dagger\}_i = \bar{s_x}\{b^\dagger\}_i \tag{6.62}$$

results, where the overbar $\bar{x}$ indicates the average value of x in K_a and K_b. Now we can write

$$\bar{\mu}(j\omega\bar{\kappa_z} + \bar{\sigma_z})\{h^\dagger\}_i = (j\omega\bar{\kappa_x} + \bar{\sigma_x}\{b^\dagger\}_i. \tag{6.63}$$

This is in fact the same form that the FDTD equations take [6, §7.8.1]. The approximation implied by (6.61) goes some ways to explaining the need for graded FDTD profiles. If the material parameters are very smoothly varying, (6.61) will be a resonable approximation, but abrupt inter-element jumps in material properties would imply a larger error. In practice, the standard cubic material grading described in [6, §7.6.2] yielded good results. Transforming to the time-domain, we get

$$\bar{\mu}(\frac{\partial}{\partial t}\bar{\kappa_z} + \bar{\sigma_z})\{h^\dagger\}_i = (\frac{\partial}{\partial t}\bar{\kappa_x} + \bar{\sigma_x})\{b^\dagger\}_i. \tag{6.64}$$

This form contains only first derivatives, and straight-forward time discretisation is possible using standard leap-frog central differencing and the semi-implicit treatment of loss terms discussed in §4.4.2.

The other equations (6.22), (6.24), (6.25) and (6.26) and are straight forward to transform to the time-domain. This results in expressions similar to (6.64), but without having to make any approximations. It is not necessary to discretise (6.26), since those relations are not involved in the time-stepping.

To make the UPML operational, the following procedure is followed. At whole time-steps (6.24) is solved for $\{\tilde{d}\}$ followed by solving (6.22) for $\{e\}$. At half time-steps, (6.25) is solved for $\{b^\dagger\}$, followed by solving (6.23) for $\{h^\dagger\}$. In the above equation references, we are assuming they have been converted to their discrete time-domain equivalents using the process previously described in this subsection.

6.4 Conclusion

Some peripheral formulations required for obtaining useful results with a time-domain FEM formulation have been discussed. The Total Field/Scattered Field (TF/SF) formulation for waveguide analysis commonly used in the FDTD context has been analysed using methods from the FEM. Using the FEM analysis, its implementation in a higher order explicit FEM setting has been described. The extraction of waveguide results from a FEM analysis has been discussed. The convolution-free explicit FDTD implementation of the PML has been analysed in a FEM framework. This analysis sheds light on some of the approximations made in the FDTD implementation, and has been used to derive a convolution free explict PML implementation in the context of a higher order explicit FEM method.

Chapter 7

Implicit-Explicit Formulation

In this chapter the hybrid implicit-explicit formulation is derived; the implicit partition is updated using an unconditionally stable Newmark-β scheme, while the explicit part is updated using leapfrog central differencing. The derivation will proceed as follows: An implicit-explicit system is first formulated for a globally Newmark-β system using the element-wise implicitness method [4]. The resulting system is then block-structured such that the implicit and explicit partitions can be solved separately. The equivalence between Newmark-β=0 and leap-frog central differencing when using suitably conforming semi-discretisations is shown, allowing the explicit partition to be updated using leap-frog central differencing rather than Newmark-β. A stability proof based on the globally Newmark-β formulation is shown. It is applicable to the Newmark-β/leap-frog hybrid through the equivalence between it and the globally Newmark-β formulation. The basic premise of this formulation is due to Rylander [4], but the explicit definition of the DOF groupings in a form suitable for arbitrary order field discretisation is novel.

7.1 Newmark-β Implicit-explicit System

The standard Newmark-β method for a lossless region (as recalled from §4.4.3) is

$$\begin{aligned}[A]\{e\}^{(n+1)} = [B]\{e\}^{n} - [A]\{e\}^{(n-1)} \\ - \{\tilde{f}\}^{n},\end{aligned} \tag{7.1}$$

where

$$[A] = \frac{1}{(\Delta t)^2}[M] + \beta[S], \quad [B] = \frac{2}{(\Delta t)^2}[M] - (1 - 2\beta[S]),$$
$$f_i = \int_\Omega \frac{\partial}{\partial t}\vec{J} \cdot \vec{w_i}^{(1)}, \tag{7.2}$$
$$\{\tilde{f}\}^n = \left[\beta\{f\}^{(n+1)} + (1-2\beta)\{f\}^n + \beta\{f\}^{(n-1)}\right],$$

$[M] = [M_\epsilon]$, and β is the implicitness parameter. With β=0 the method is explicit and involves only the mass matrix on the right hand side. With $\beta > 0$ the method is implicit, and with $\beta \geq 0.25$ it is unconditionally stable. In (7.2) β is a global parameter. The implicitness parameter can be adjusted on a per-element basis by rewriting (7.2) [4] as

$$[A] = \sum_{k=1}^{n_e}(\frac{1}{\Delta t^2}[M_k] + \beta_k[S_k]), \qquad [B] = \sum_{k=1}^{n_e}(\frac{2}{\Delta t^2}[M_k] - (1-2\beta_k)[S]_k), \tag{7.3}$$

$$\{\tilde{f}\}^n = \sum_{k=1}^{n_e} \beta_k\{f\}_k^{n+1} + (1-2\beta_k)\{f\}_k^n + \beta_k\{f\}_k^{n-1}, \tag{7.4}$$

where n_e is the number of elements, β_k element k's implicitness parameter and $[M_k]$ and $[S_k]$ the element-local, globally numbered mass and stiffness matrices.

7.2 Block-structured System

The domain Ω is split such that $\Omega = \Omega_{imp} \cup \Omega_{exp}$ where Ω_{imp} contains only implicit elements (i.e. $\beta_k = \frac{1}{4}\ \forall\ k \in \Omega_{imp}$) and Ω_{exp} contains only explicit elements (i.e. $\beta_k = 0\ \forall\ k \in \Omega_{exp}$). Five categories of DOF can be defined based on their relation to Ω_{imp} and Ω_{exp}:

a implicit dofs that are not connected[1] to any elements connected to any dofs that are on the hybrid boundary

b dofs in implicit elements where the element is connected[2] to explicit element(s) but where the dofs themselves are fully contained in the implicit region, i.e. they are not connected to the explicit element.

c dofs on the interface between an implicit and an explicit element. They are updated in a mixed implicit/explicit fashion, i.e. the global system

[1] Connection of a DOF to an element implies that the DOF's basis function is non-zero in said element. Any two DOFs connected to a common element represent a potential non-zero entry in a system matrix.

[2] Connection between elements imply that they share a DOF.

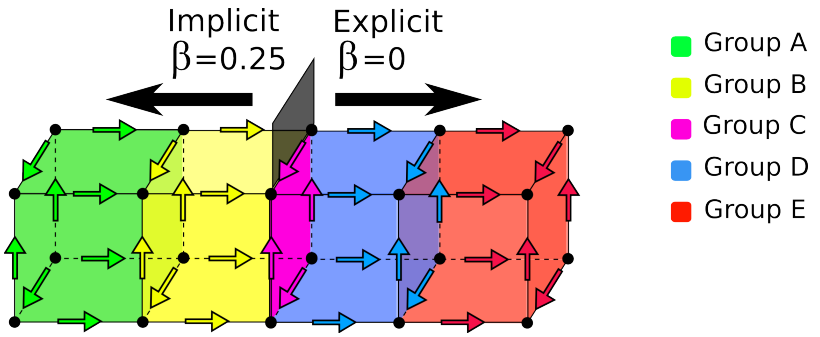


Figure 7.1: Implicit and explicit hybrid degree of freedom groupings.

matrix entries for these dofs are the sum of a part that is calculated using $\beta = \frac{1}{4}$ in the implicit element and another using β=0 in the explicit element.

d dofs in explicit elements where the element is connected to implicit element(s) but where the dofs themselves are fully contained in the explicit region.

e explicit dofs that are not connected to any elements connected to any dofs that are on the hybrid boundary

Groups e) and d) are the mirror images of groups a) and b). The groups are illustrated in Fig. 7.1.

While the time-integration in Ω_{exp} is explicit, the mass matrix still needs to be inverted; Gauss-Lobatto lumping [78] is used to diagonalise the explicit elements' mass as described in §5.2.3. By first numbering the DOFs in group a), then b), etc. the following block structured system is obtained:

$$[A] = \begin{bmatrix} [A_{aa}] & [A_{ab}] & 0 & 0 & 0 \\ [A_{ba}] & [A_{bb}] & [A_{bc}] & 0 & 0 \\ 0 & [A_{cb}] & [A_{cc}] & 0 & 0 \\ 0 & 0 & 0 & [A_{dd}] & 0 \\ 0 & 0 & 0 & 0 & [A_{ee}] \end{bmatrix}, \tag{7.5}$$

$$[B] = \begin{bmatrix} [B_{aa}] & [B_{ab}] & 0 & 0 & 0 \\ [B_{ba}] & [B_{bb}] & [B_{bc}] & 0 & 0 \\ 0 & [B_{cb}] & [B_{cc}] & [B_{cd}] & 0 \\ 0 & 0 & [B_{dc}] & [B_{dd}] & [B_{de}] \\ 0 & 0 & 0 & [B_{ed}] & [B_{ee}] \end{bmatrix}, \tag{7.6}$$

and

$$\{e\} = \{e_a\} \oplus \{e_b\} \oplus \{e_c\} \oplus \{e_d\} \oplus \{e_e\}. \tag{7.7}$$

Taking advantage of (7.5) being block-diagonal, the solution can be partitioned as

$$\begin{aligned}[A_{imp}](\{e_a\} \oplus \{e_b\} \oplus \{e_c\})^{n+1} = \\ [B_{imp}](\{e_a\} \oplus \{e_b\} \oplus \{e_c\} \oplus \{e_d\})^n + \\ [A_{imp}](\{e_a\} \oplus \{e_b\} \oplus \{e_c\})^{n-1} + \tilde{\mathbf{f}}^n_{imp}\end{aligned} \tag{7.8}$$

$$\begin{aligned}[A_{exp}](\{e_d\} \oplus \{e_e\})^{n+1} = \\ [B_{exp}](\{e_c\} \oplus \{e_d\} \oplus \{e_e\})^n + \\ [A_{exp}](\{e_d\} \oplus \{e_e\})^{n-1} + \tilde{\mathbf{f}}^n_{exp},\end{aligned} \tag{7.9}$$

with

$$[A_{imp}] = \begin{bmatrix} [A_{aa}] & [A_{ab}] & 0 \\ [A_{ba}] & [A_{bb}] & [A_{bc}] \\ 0 & [A_{cb}] & [A_{cc}] \end{bmatrix}, \tag{7.10}$$

$$[A_{exp}] = \begin{bmatrix} [A_{dd}] & 0 \\ 0 & [A_{ee}] \end{bmatrix}, \tag{7.11}$$

$$[B_{imp}] = \begin{bmatrix} [B_{aa}] & [B_{ab}] & 0 & 0 \\ [B_{ba}] & [B_{bb}] & [B_{bc}] & 0 \\ 0 & [B_{cb}] & [B_{cc}] & [B_{cd}] \end{bmatrix}, \tag{7.12}$$

$$[B_{exp}] = \begin{bmatrix} [B_{cc}] & [B_{cd}] & 0 \\ [B_{dc}] & [B_{dd}] & [B_{de}] \\ 0 & [B_{ed}] & [B_{ee}] \end{bmatrix}. \tag{7.13}$$

7.3 Newmark-β=0 and Leapfrog Equivalence

Equivalence between the two systems depends on the fact that, when using conforming discrete 1- and 2-forms for respectively $\vec{E}$ and $\vec{B}$,

$$[S_{\mu^{-1}}] = [C]^T [M_{\mu^{-1}}][C] \tag{7.14}$$

holds[3] [59]. Writing out the source-free Newmark-β update equation for $\{e\}^{n+1}$ with $\beta = 0$ and multiplying by Δt^2, we get

$$[M_\epsilon]\{e\}^{n+1} = (2[M_\epsilon] - \Delta t^2 [S_{\mu^{-1}}])\{e\}^n - [M_\epsilon]\{e\}^{n-1}. \tag{7.15}$$

[3] See also the discussion in §4.3.6.

The source free form of the leapfrog update equations (4.101, 4.102) can be written as

$$\{b\}^{n+1/2} = \{b\}^{n-1/2} - \Delta t[C]\{e\}^n \tag{7.16}$$

and

$$[M_\epsilon]\{e\}^{n+1} = [M_\epsilon]\{e\}^n + \Delta t[C]^T[M_{\mu^{-1}}]\{b\}^{n+1/2}. \tag{7.17}$$

Writing (7.17) for the previous timestep

$$[M_\epsilon]\{e\}^n = [M_\epsilon]\{e\}^{n-1} + \Delta t[C]^T[M_{\mu^{-1}}]\{b\}^{n-1/2} \tag{7.18}$$

and subtracting,

$$[M_\epsilon](\{e\}^{n+1}-\{e\}^n) = [M_\epsilon](\{e\}^n-\{e\}^{n-1})+\Delta t[C]^T[M_{\mu^{-1}}](\{b\}^{n+1/2}-\{b\}^{n-1/2}) \tag{7.19}$$

results. Using (7.16),

$$\{b\}^{n+1/2} - \{b\}^{n-1/2} = -\Delta t[C]\{e\}^n \tag{7.20}$$

and substituting into (7.19) we get

$$[M_\epsilon](\{e\}^{n+1}-\{e\}^n) = [M_\epsilon](\{e\}^n-\{e\}^{n-1})-\Delta t^2[C]^T[M_{\mu^{-1}}][C]\{e\}^n. \tag{7.21}$$

Finally, substituting (7.14) and arranging terms, (7.15) again emerges, proving the equivalence of the source free fully discrete Newmark-β=0 and leapfrog formulations.

7.4 Stability

In this section the stability of the implicit-explicit Newmark-β formulation is proved. The stability of the Newmark-β/leapfrog formulation follows by its equivalence to the globally Newmark-β formulation. The proof proceeds by first proving stability in the von Neumann sense and then discussing caveats not covered by the von Neumann stability.

7.4.1 Von Neumann Stability

The von Neumann criterion requires that all the eigenvalues of the fully discrete system have a magnitude less than or equal to 1. The proof follows along the lines of [4]. Rewrite (7.1) in source-free form in terms of a summation over the elements using (7.4):

$$\sum_k^{n_e} \Bigg([S_k][\beta_k\{e\}^{n+1} - (2\beta_k - 1)\{e\}^n + \beta_k\{e\}^{n-1}] + \frac{1}{\Delta t^2}[M_k][\{e\}^{n+1} - 2\{e\}^n + \{e\}^{n-1}] \Bigg) = 0 \tag{7.22}$$

Let $\{\tilde{e}\}$ be a complex eigen mode of (7.22) with an assumed growth factor ρ such that at timestep n $\{e\}^n = \rho^n\{\tilde{e}\}$. Substituting into (7.22),

$$\sum_k^{n_e} \left([S_k][\beta_k\rho^2 - (2\beta_k - 1)\rho + \beta_k] + \frac{1}{\Delta t^2}[M_k][\rho^2 - 2\rho + 1] \right) \{\tilde{e}\} = 0. \tag{7.23}$$

A necessary condition for stability is that

$$|\rho| \leq 1 \text{ for all modes } \{\tilde{e}\}. \tag{7.24}$$

With the substitution $\rho = (1+\xi)/(1-\xi)$ (7.23) becomes

$$\sum_k^{n_e} \{\tilde{e}\}^H [S_k]\{\tilde{e}\} = -\xi^2 \sum_k^{n_e} \{\tilde{e}\}^H \left[\frac{4}{\Delta t^2}[M_k] + [S_k](4\beta_k - 1) \right] \{\tilde{e}\}, \tag{7.25}$$

and the necessary stability condition (7.24) becomes

$$\Re(\xi) \leq 0. \tag{7.26}$$

Since the matrices in (7.25) are all Hermitian, the quadratic forms yield real results and consequently ξ^2 is real. The LHS of (7.25) is non-negative since $[S_k]$ is positive-semidefinite. Given (7.26) and (7.25), (7.23) is satisfied if

$$\sum_k^{n_e} \{\tilde{e}\}^H \left[\frac{4}{\Delta t^2}[M_k] + [S_k](4\beta_k - 1) \right] \{\tilde{e}\} \geq 0 \text{ for all modes } \{\tilde{e}\}. \tag{7.27}$$

The summation in (7.27) can be split into respective sums over the implicit and explicit elements. Over the implicit elements, $\beta_k \geq 0.25$ and (7.27) is always satisfied since both the $[M_k]$ and $[S_k]$ terms have positive weights.

For the explicit elements with $\beta_k = 0$, we first need to consider the eigen-system of the individual explicit elements

$$[S_k]\{e\} = \lambda[M_k]\{e\}. \tag{7.28}$$

With positive semi-definite element matrices, the inequality

$$\{e\}^H[S_k]\{e\} \leq \lambda_{max(k)}\{e\}^H[M_k]\{e\} \tag{7.29}$$

holds for any complex vector $\{e\}$.

A sufficient condition for (7.27) to hold is if the contribution from each element is non negative. With $\beta = 0$ this implies

$$\frac{4}{\Delta t^2}\{e\}^H[M_k]\{e\} \geq \{e\}^H[S_k]\{e\}, \tag{7.30}$$

and by (7.29)

$$\frac{4}{\Delta t^2}\{e\}^H[M_k]\{e\} \geq \lambda_{max(k)}\{e\}^H[M_k]\{e\} \tag{7.31}$$

yielding

$$\Delta t \leq \frac{2}{\lambda_{max(k)}}. \tag{7.32}$$

As discussed in §5.3, lumped mixed first order bricks in the explicit region is equivalent to the FDTD. For such elements, it can be shown that $\lambda_{max(k)} = 1/c^2(1/\Delta x^2 + 1/\Delta y^2 + 1/\Delta z^2)$ where $\Delta x, y, z$ are the grid spacings. This yields the familiar FDTD CFL limit.

7.4.2 Newmark-β Late-time Stability

A fully discrete system can be described in terms of an amplification matrix $[A]$ such that

$$\{u\}^{n+1} = [A]\{u\}^n. \tag{7.33}$$

For the Newmark-β system, $\{u\}^n = \{e\}^{n-1} \oplus \{e\}^n$ and $\{u\}^{n+1} = \{e\}^n \oplus \{e\}^{n+1}$. The von Neumann stability criterion proved in §7.4.1 which implies that the magnitudes of the eigen values of $[A]$ is smaller than or equal to 1, is necessary but not sufficient for late time stability; while it precludes exponentially growing modes, polynomially growing modes are still possible for eigenvalues with unit magnitude if the system amplification matrix is non-diagonalisable [130]. In [130] it is shown that leap-frog systems where the discrete curl of $\vec{E}$ and curl of $\vec{H}$ operators are transposes of each other have diagonalisable amplification matrices, i.e. the leapfrog system of §4.4.2 is fully stable. Stability of the Newmark-β system is analysed in [131], and the existence of linear growth modes shown.

The unstable modes are non-physical. If only stable, physical modes are initially excited, the unstable modes should never manifest, since multiplication by $[A]$ is closed[4] for the stable electrodynamic and electrostatic spaces [131]. Assuming exact arithmetic, a properly set-up problem should therefore, in theory, never suffer from linear growth. However, numerical solutions suffer from errors due to numeric precision and also residual error if iterative matrix solution is used. This inevitably leads to the eventual excitation of unstable modes.

In many practical simulations, the contribution from the unstable modes would not be appreciable. The instability is most apparent when very long simulation times are needed, e.g. if closely spaced resonances of high-Q systems need to be resolved, or if very small values of Δt is called for. In some of the numerical results calculated in §9 late-time finite linear growth was noticed only with closed, lossless problems, but it was never large enough to compromise the results in the frequency range of interest. Results that were

[4]Closed in this sense means that if the initial state of the discrete system contains only stable electrodynamic and electrostatic fields, the result of multiplication by $[A]$ is guaranteed to be stable too. Put another way, the unstable modes can only manifest themselves if they are explicity introduced.

calcuted with a PML termination did not exhibit any growth. A method for identifying and explicitly removing unstable modes from the numerical solution at intervals is presented in [131]. Although not pursued further, it should be applicable to the Newmark-β partition of the hybrid system.

One might rightly ask why the linearly growing modes exist for the Newmark-β=0 and not for the leapfrog formulation when they were shown to be equivalent in §7.3. However, exact equivalence only holds in source-free regions, while the growing modes of the Newmark-β formulation are excited by numerical noise acting as de-facto sources.

7.5 Conclusion

An approach to the formulation of a higher order implicit-explicit hybrid system has been presented. The theoretical basis of the hybrid presented in [4] has been extended to take into account higher order field discretisation. The hybrid formulation's stability and some remaining caveats related to long-time stability have been discussed. The newly extended higher order implicit-explicit formulation in conjuction with the higher order hybrid mesh discussed in §8 allow the construction of higher order adaptations of existing [4] implicit/explicit hybrid mesh methods.

Chapter 8

Hybrid Mesh

In this chapter, a novel method for arbitrary order discrete 1-form ($H(curl)$), field representation on a hexahedral/tetrahedral hybrid-mesh is presented. This chapter is an expanded revision of [132] which has been published in the IET Electronics Letters. The numerical solution of full-wave or eddy current electromagnetic problems often call for 1-form ($H(curl)$) or 2-form ($H(div)$) conforming discrete representations. The method presented here is directly applicable to the discretisation of 1-forms on hybrid meshes, and can be almost trivially extended to deal with 2-form discretisation. Conforming representations have been defined on many element shapes, allowing the analysist to choose the shape best suited to the problem at hand. As discussed in §4.3.1, tetrahedra and hexahedra represent two largely complementary shapes. Unstructured tetrahedra excel at modelling complex geometries, while structured Cartesian hexahedra (i.e. bricks) facilitate the efficient representation of regular geometries; they also allow relatively straight forward metric matrix diagonalisation as discussed in §5.2.3. Using a hybrid mesh discretisation allows the best of both worlds by only using tetrahedra where they are needed to represent complex geometrical features.

Hybrid meshes utilising mixed 1st order $H(curl)$ conforming pyramidal elements [133, 134, 135] to connect the tetrahedral and hexahedral regions have been presented for hybrid FEM/FDTD PDE methods [4] and integral methods [136]. When dealing with electrically large problems or if very high accuracy is required, the use of higher order field representation is desirable, as discussed in §1. While higher order pyramidal representations of up to second [133] and mixed third [134] order have been published, they seem to suffer from spurious modes as shown below. The conforming mesh scheme developed in this chapter takes a different approach altogether, avoiding the use of pyramidal elements. Instead, suitably constrained higher order tetrahedral basisfunctions are used to continue the hexahedral degrees of freedom (DOFs) on the hex/tet hybrid boundary into the tetrahedral mesh.

The first section shows how to construct a mixed first order tetrahedral

discretisation that can conform (in the functional sense) to a hexahedral discretisation, provided the two meshes conform geometrically. The second section extends this construction to arbitrary order discretisations and somewhat relaxed geometrical conformance between the two mesh types. This is followed by a brief numerical validation of this scheme, showing that spurious modes are avoided and that higher order convergence is achieved. At the same time, the avoidance of a third element class (i.e. pyramids) leads to simpler program construction. More complete numerical evaluations are presented in §9 and §10.

8.1 Constructing hexahedral-conforming tetrahedra

The aim is to construct $H(curl)$ conforming basis functions defined on a tetrahedral mesh that can conform to $H(curl)$ conforming functions defined on an adjoining hexahedral mesh. What is meant by conform in this instance is:

1. The basis functions defined on the tetrahedral mesh should share degrees of freedom with the adjoining hexahedral basis functions.
2. There should be no additional tetrahedral degrees of freedom that are not shared with the hexahedra.
3. The global basis function defined by the union of the tetrahedral and hexahedral function should be globally curl conforming, i.e. on the union of the tetrahedral and hexahedral domains.

To see why the tetrahedral functions need to conform to the hexahedral functions and not vice versa, assume the interface between the tetrahedra and hexahedra are geometrically conforming in the sense that every hex-face intersects with exactly two tet-faces as shown in Fig. 8.1. Because of the diagonal edge introduced by the tetrahedra on Γ, there are more tetrahedral degrees of freedom on Γ than hexahedral. Hence, the tetrahedra have the freedom to conform to the hexahedra, and not the other way around.

The conditions required for curl-confromance of a function at the intersection of two domains [32, Lemma 5.3] are now shown. Suppose two non-overlapping domains K_1 and K_2 meet at a common surface Γ such that $K_1 \cap K_2 = \Gamma$. Suppose also that there are curl conforming functions defined in each domain

$$\vec{f_1} \in H(curl; K_1) \text{ and } \vec{f_2} \in H(curl; K_2). \tag{8.1}$$

We define a new function as

$$\vec{f} = \begin{cases} \vec{f_1} & \text{in } K_1 \\ \vec{f_2} & \text{in } K_2. \end{cases} \tag{8.2}$$

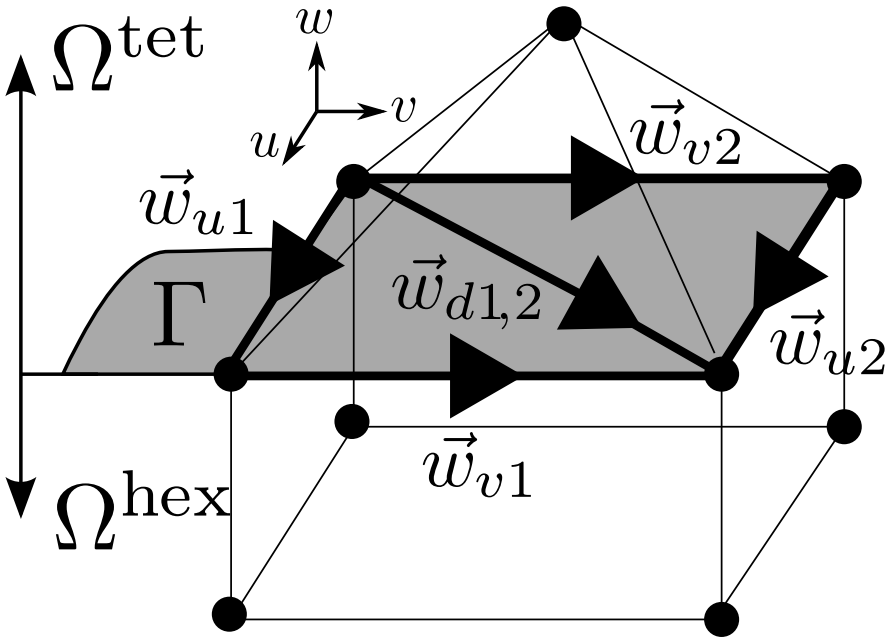


Figure 8.1: Interface between between hexahedral and tetrahedral mesh with degrees of freedom required for mixed 1st order hybrid shown. The subscript d indicates basis functions related to the diagonal edge.

If

$$\hat{n} \times \vec{f}_1|_\Gamma = \hat{n} \times \vec{f}_2|_\Gamma \tag{8.3}$$

holds where $\hat{n}$ is a surface normal, then

$$\vec{f} \in H(curl; K_1 \cup K_2). \tag{8.4}$$

We identify K_1 as the hexahedral cell below Γ in Fig. 8.1, and K_2 as the union of the two tetrahedra above Γ. Assume that K_1 has a discrete space $H(curl; K_1)_h$ with basis $\{\vec{w}^{\text{hex}}\}$ of the type described in §5.2.3 and that K_2 has a discrete space $H(curl; K_2)_h$ with basis $\{\vec{w}^{\text{tet}}\}$ of the type described in §5.2.2 defined on it. By definition, any $\vec{f}_1 \in H(curl; K_1)_h$ and $\vec{f}_2 \in H(curl; K_2)_h$ satisfies (8.1). For simplicity assume that $H(curl; K_1)_h$ is a mixed first order discretisation. The hex-face in Fig. 8.1 contains four edge basis functions $\vec{w}^{\text{hex}}_{u1}$, $\vec{w}^{\text{hex}}_{u2}$, $\vec{w}^{\text{hex}}_{v1}$ and $\vec{w}^{\text{hex}}_{v2}$.

The goal is to find a function $\vec{w}^{\text{hct}}_{u1} \in$ such that

$$\hat{n} \times \vec{w}^{\text{hct}}_{u1} = \hat{n} \times \vec{w}^{\text{hex}}_{u1} \text{ on } \Gamma, \tag{8.5}$$

where the superscript "hct" refers to *hex compatible tet*, and similar for the other three hexahedral basis functions. Obviously, the standard mixed first order tetrahedral $H(curl)$ basis functions will *not* satisfy (8.5), since the tetrahedral basis functions related to non-diagonal edge degrees of freedom

will be identically zero on one of the two triangular faces[1], while all four hex edge functions are nonzero over the whole hex-face. Instead $\hat{n} \times \vec{w}_{u1}^{\text{hct}}$ needs to be written as a linear combination of basis functions that have support on either face. For such a combination to satisfy (8.5), it is required[2] that

$$\Upsilon_\Gamma(H(curl, K_2))_h \supseteq \Upsilon_\Gamma(H(curl, K_1))_h, \tag{8.6}$$

in other words, the function space of components tangential to Γ modeled by the tetrahedral basis in K_2 must at least cover the tangential space modeled by the hexahedral basis in K_1.

Mixed p'th order Cartesian product $H(curl)$ hexahedral(hex) basis functions of the type considered here take the form

$$\begin{aligned}
\vec{w}_{(u)}^{\text{hex}} &= P_v^p(v)P_w^p(w)P_u^{p-1}(u)\hat{u}, \\
\vec{w}_{(v)}^{\text{hex}} &= P_w^p(v)P_u^p(w)P_v^{p-1}(u)\hat{v}, \\
\vec{w}_{(w)}^{\text{hex}} &= P_u^p(v)P_v^p(w)P_w^{p-1}(u)\hat{w},
\end{aligned} \tag{8.7}$$

where u, v, w are the element-local coordinates, $\hat{u}, \hat{v}, \hat{w}$ the covariant component basis vectors and P^p is a p'th order univariate polynomial. On any hexahedral face the normally varying coordinate is constant, hence the mixed p'th order basis functions described in (8.7) will contain multivariate polynomial terms of at most order $2p-1$. Hence, tetrahedral $H(curl)$ elements complete to order $2p-1$ that conform to the hex-face geometry (as in Fig. 8.1) can tangentially match the hexahedral basis functions exactly. In this case we therefore need $H(curl; K_2)_h$ to be consistently linear, rather than mixed first order. A new tetrahedral basis that conforms to the hexahedral basis can now be constructed using $\vec{w}_{u1}^{\text{tet}}, \vec{w}_{u2}^{\text{tet}}, \vec{w}_{v1}^{\text{tet}}, \vec{w}_{v2}^{\text{tet}}, \vec{w}_{d1}^{\text{tet}}$ of the form $\zeta_i\nabla\zeta_j - \zeta_j\nabla\zeta_i$ and $\vec{w}_{d2}^{\text{tet}}$ of the form $\zeta_i\nabla\zeta_j + \zeta_j\nabla\zeta_i$, where ζ_i is the simplex coordinate related to node i.

Applying (8.5) and solving for the four $\vec{w}_i^{\text{hct}}$ functions,

$$\begin{aligned}
\vec{w}_{u1}^{\text{hct}} &= \vec{w}_{u1}^{\text{tet}} + 0.5\vec{w}_{d1}^{\text{tet}} + 0.5\vec{w}_{d2}^{\text{tet}}, \\
\vec{w}_{u2}^{\text{hct}} &= \vec{w}_{u2}^{\text{tet}} + 0.5\vec{w}_{d1}^{\text{tet}} - 0.5\vec{w}_{d2}^{\text{tet}}, \\
\vec{w}_{v1}^{\text{hct}} &= \vec{w}_{v1}^{\text{tet}} + 0.5\vec{w}_{d1}^{\text{tet}} - 0.5\vec{w}_{d2}^{\text{tet}}, \\
\vec{w}_{v2}^{\text{hct}} &= \vec{w}_{v2}^{\text{tet}} + 0.5\vec{w}_{d1}^{\text{tet}} + 0.5\vec{w}_{d2}^{\text{tet}}
\end{aligned} \tag{8.8}$$

results[3]. The sign of the $\vec{w}_{d2}^{\text{tet}}$ term is negative if the edge is connected to the ending node of the diagonal. While one could implement this basis

[1] Recall the support of tetrahedral edge functions from §5.1.1.

[2] Recall Υ_Γ, the trace operator operator from §3.4.2. Since we are dealing with 1-forms, it represents the tangential trace here.

[3] By historical precedent, many workers in the EM FEM field use a varient of the tetrahedral first order basis functions that are normalised by the length of the edge that the function is associated with. In practice the normalisation makes no computational difference, but makes expressions like (8.8) and also the software implementation much more complicated. However, popular texts such as [21] use the edge length normalised form, so one must be aware of the difference.

directly, a more general implementation is described in the next section. A similar procedure using a normal continuity condition in place of (8.5) and $H(div)$ bases can also be used to construct an $H(div)$ conforming hybrid mesh discretisation.

8.2 Implementation to arbitrary order

The approach developed in the previous section could, in principle be extended to analytically construct higher order conforming hybrid mesh discrete representations. While possible, it would be intractable to construct the higher order hexahedral-conforming tetrahedral basis in this way. Besides the analytical difficulty, the software implementation in a FEM code is also complicated by the fact the special basis should only be used for degrees of freedom on the boundary Γ. Furthermore, a boundary tetrahedra may have between one and three faces on Γ. Instead, a numerically exact solution is obtained by solving a local projection problem on each hexahedral hybrid boundary face Γ_K.

The mixed p'th order hexahedral basis on the single hexahedral face Γ_K is written as

$$\{\vec{w}_K^{\text{hex}}\}^T = \{\vec{w}_1^{\text{hex}}, \vec{w}_2^{\text{hex}}, \ldots, \vec{w}_{n_{\text{hex}}}^{\text{hex}}\}, \tag{8.9}$$

where n_{hex} is the number of hexahedral basis functions with a non-zero trace on Γ_K. Similarly, the full[4] order $2p-1$ tetrahedral basis on Γ_K is

$$\{\vec{w}_K^{\text{tet}}\}^T = \{\vec{w}_1^{\text{tet}}, \vec{w}_2^{\text{tet}}, \ldots, \vec{w}_{n_{\text{tet}}}^{\text{tet}}\}, \tag{8.10}$$

where n_{tet} is the number of tetrahedral basis functions with a non-zero trace on Γ_K. Now we need to find a local $n_{\text{hex}} \times n_{\text{tet}}$ transform matrix $[{}^{\text{hc}}T_K]$ such that

$$\{w_K^{\text{hct}}\} = [{}^{\text{hc}}T_K]\{\vec{w}_K^{\text{tet}}\} \tag{8.11}$$

where $\{w_K^{\text{hct}}\}$ is the local hexahedral-conforming tetrahedral basis

$$\{\vec{w}_K^{\text{hct}}\}^T = \{\vec{w}_1^{\text{hct}}, \vec{w}_2^{\text{hct}}, \ldots, \vec{w}_{n_{\text{hex}}}^{\text{hct}}\}. \tag{8.12}$$

Consider the construction of the i'th member of $\{\vec{w}_K^{\text{hct}}\}$:

$$[{}^{\text{hc}}T_K]_i\{\vec{w}_K^{\text{tet}}\} = \vec{w}_i^{\text{hex}}. \tag{8.13}$$

By defining the trace inner product

$$\langle \vec{a}, \vec{b} \rangle_{\Gamma_K} = \int_{\Gamma_K} (\hat{n} \times \vec{a}) \cdot (\hat{n} \times \vec{b})\, dA, \tag{8.14}$$

[4] In other words not mixed order.

and testing both sides of (8.13) with the tetrahedral basis functions, it is cast as a projection problem:

$$[{}^{\mathrm{hc}}T_K]_i[M_K] = \{P_K^i\}, \tag{8.15}$$

where

$$[M_K]_{ij} = \langle \vec{w}_j^{\mathrm{tet}}, \vec{w}_i^{\mathrm{tet}} \rangle \qquad i,j = 1 \ldots n_{\mathrm{tet}} \tag{8.16}$$

and $\{P_K^i\}$ is a column vector with entries

$$\{P_K^i\}_j = \langle \vec{w}_j^{\mathrm{tet}}, \vec{w}_i^{\mathrm{hex}} \rangle \qquad j = 1 \ldots n_{\mathrm{tet}}. \tag{8.17}$$

Since the tetrahedral basis has specfically been chosen to be a superset of the hexahedral basis on Γ_K, a solution to (8.15) is an exact solution to (8.13). Considering (8.15) for every $\vec{w}_i^{\mathrm{hex}} \in \{\vec{w}_K^{\mathrm{hex}}\}$, a matrix equation results with the solution

$$[{}^{\mathrm{hc}}T_K] = [P_K][M_K]^{-1}, \tag{8.18}$$

where $[P_K]$ is the matrix with entries

$$[P_K]_{ij} = \langle \vec{w}_j^{\mathrm{tet}}, \vec{w}_i^{\mathrm{hex}} \rangle \qquad i = 1 \ldots n_{\mathrm{hex}},\ j = 1 \ldots n_{\mathrm{tet}}. \tag{8.19}$$

To implement the hybrid mesh globally, the local $[{}^{\mathrm{hc}}T_K]$ matrices are assembled into a global $n_{\mathrm{tot}} \times n_{\mathrm{tet}}$ transform matrix $[{}^{\mathrm{hc}}T]$, where n_{tot} is the total number of DOFs and n_{tet} is now the number of global tetrahedral DOFs. Order $2p-1$ tetrahedral bases are used on the hybrid boundary and mixed p'th order throughout the rest of the tetrahedral mesh. Hybrid system matrices can be constructed as

$$[A_{\mathrm{hyb}}] = [A_{\mathrm{hex}}] + [{}^{\mathrm{hc}}T][A_{\mathrm{tet}}][{}^{\mathrm{hc}}T]^T, \tag{8.20}$$

where $[A_{\mathrm{hex}}]$ is the system matrix resulting from the hexahedral elements, and has been zero extended to $n_{\mathrm{tot}} \times n_{\mathrm{tot}}$, and $[A_{\mathrm{tet}}]$ is the system matrix due to the tetrahedral elements.

This method might seem somewhat indirect, but it has several important practical advantages:

- Any non-trivial EM FEM code will need routines to calculate matrices like $[P_K]$ and $[M_K]$ in any case.
- Once this method has been coded, hybrid-mesh support for different basis sets require no extra work. Used with arbitrary order tetrahedral (e.g. [8]) and hexahedral (e.g. [78]) basis sets, straightforward implementation of arbitrary order hybrid meshes is possible.
- Many sources of human programming error are eliminated.

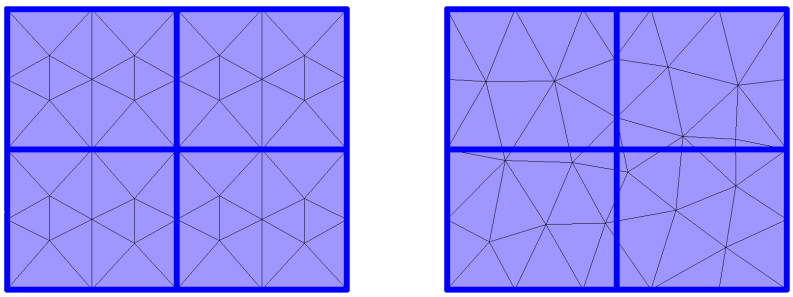


Figure 8.2: Aceptable and unacceptable discretisations of the hybrid mesh interface. The hexahedral faces are indicated with blue lines. No triangles in the acceptable mesh on the left cross any of the blue lines. Some triangles in the unacceptable mesh on the left straddle more than one hexahedral face, and have edges that cross the blue lines.

- The numerical solution of $[^{\text{hc}}T_K]$ is a purely local problem, and therefore the matrices involved are small. They are also calculated only once during the matrix filling step; in practice the time taken to calculate $[^{\text{hc}}T]$ is negligible.
- The geometric conformity requirement between the hexahedral and tetrahedral elements are relaxed somewhat as discussed below.

The only requirement for (8.11) to have an exact solution is that the tetrahedral basis is at least of order $2p-1$ on Γ and that the tet-faces on Γ conform to the hexahedral face edges. It is therefore perfectly acceptable to use more than two tet-faces per hex-face. A suitable and an unsuitable multi-tet interface mesh is shown in Fig. 8.2. This feature allows, for instance, *hp* refinement in the tet region independent of the hexahedral mesh.

8.3 Numerical Verification

To verify the proposed hybrid mesh, the eigen-solution of the vector Helmholtz equation,

$$\nabla \times \nabla \times \vec{E} - k^2 \vec{E} = 0, \tag{8.21}$$

in a 19x23x29 m PEC cavity (speed of light normalised to 1 m/s) is obtained, yielding the cavity mode wavenumbers. Numerical results using hexahedral, hybrid hexahedral-tetrahedral and two different sets of pyramidal elements are compared in Table 8.1. The hexahedral mesh has a cell-size of 29/4 m. The pyramidal mesh is formed by splitting each hex-element into six pyramids. The hybrid mesh utilises half of the hexahedral mesh; the other half is meshed with unstructured tetrahedra that conform to the hex faces on

Solution type	Mode eigen-value k_0^2 $(\text{rad.s}^{-1})^2$				RMS error %	total DOFs
	I	II	III	IV		
Analytic	0.030393	0.039075	0.045997	0.057732	—	—
Mixed 1st order discrete						
Hexahedra	0.028862	0.036075	0.042648	0.053793	6.78	75
Pyramids	0.030821	0.040152	0.046792	0.058448	1.88	459
Hybrid	0.029659	0.037784	0.043875	0.055175	3.80	234
Mixed 2nd order discrete						
Hexahedra	0.030384	0.039048	0.045968	0.057700	0.056	854
Pyramids 1	0.007825	0.007877	0.007970	0.010426	—	2774
Pyramids 2	0.000619	0.000679	0.000683	0.000779	—	3062
Pyramids 3	0.030402	0.039100	0.046025	0.057765	0.055	3062
Hybrid	0.030394	0.039078	0.046002	0.057767	0.031	1596

Table 8.1: First four eigenvalues for a rectangular cavity

Γ. The hexahedral elements apply Gauss-Lobatto mass lumping as described in §5.2.3, making them suitable for explicit time-domain FEM methods.

Both pyramidal element sets are identical at mixed 1st order, and show no spurious modes. For mixed 2nd order, the Coulomb elements [133] ("Pyramids 1" in Table 8.1) suffer from spurious modes throughout the spectrum, while the Graglia [134] pyramids exhibit a limited number of spurious modes at frequencies ranging from about $\frac{1}{50}$ to $\frac{1}{4}$ of the lowest physical cavity mode eigenvalue. In Table 8.1, "Pyramids 2" show the first four spurious eigenvalues of the Graglia elements. "Pyramids 3" show the physical eigenvalues calculated using Graglia's elements after discarding the spurious values. In light of the spurious modes exhibited by the pyramids, a hybrid mesh using hexahedra, tetrahedra and pyramids was not constructed. The hybrid mesh developed here does not suffer from spurious modes, and delivers accurate results.

The convergence of the hexahedral and hybrid solution eigenvalues as the total number of DOFs increases is compared in Fig. 8.3. The rate of convergence observed for the hybrid mesh discretisation is the same as that of the individual hexahedral and tetrahedral discretisations. The accuracy per degree of freedom of the pure hexahedral and hybrid meshes are also similar.

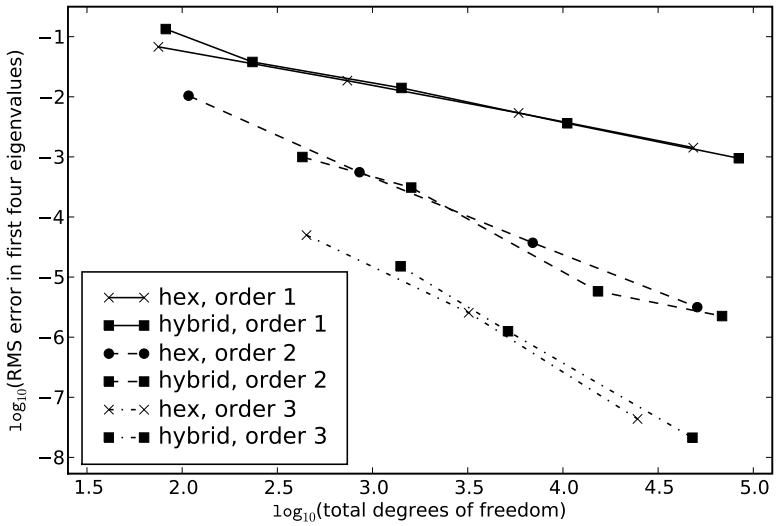


Figure 8.3: Convergence of hexahedral and hybrid mesh eigenvalues for mixed order 1 through 3 discretisation.

8.4 Conclusion

A new approach for constructing $H(curl)$ and $H(div)$ conforming hexahedral / tetrahedral hybrid meshes that enables the straightforward implementation of arbitrary order hybrid discretisations has been proposed. The correct functioning of $H(curl)$ hybrids have been demonstrated using the finite element method eigen-solution of a rectangular cavity. The higher order hybrid solution errors converge at the expected rate while being free of spurious modes. Compared to existing hybrids using pyramidal elements, the complication of an additional element type is removed, while avoiding the problem of spurious modes that currently available higher order pyramidal elements have been found to suffer from. The new hybrid mesh approach allows the construction of higher order adaptations of existing [4, 136] hybrid mesh methods.

Chapter 9

Validation

When developing a new computational method, it is critical to validate its proper working. Even if convergence proofs are available for every part of the method you implement, it is the nature of software to contain bugs. A good way of verifying a code is to see how it converges as the mesh is h- and p-refined. If the convergence matches the theoretical expectation, it is a good sign that that the code is working right, although by no means does it guarantee that it is correct. In this chapter some simple validation methodologies and results are presented. Some of these methods are described in the paper [24] that was prepared in the process of the current PhD work.

9.1 Cavity Eigen Solution

The eigen solution of cavities is a good test of a new code. In one sense eigen problems are the simplest possible to solve; it need no sources, no termination schemes, not even time discretisation. The eigen problem is essentially a test of the semi-discretisations. It can be applied several ways:

- As a test of 1-form semi-discretisations and their curls, the standard Helmholtz vector-wave eigen problem (4.113) can be solved.
- As a test of 1-form and 2-form semi-discretisations and the $[C]$ discrete curl operator, (4.112) can be sovled.
- As a test of 2-form discretisations and their divergence, the linearized acoustic vector wave equation [77] can be solved.

Only the first two solutions are strictly neccesary for EM problems, since the divergence of the basis functions are never needed in EM codes. The last test is still useful as an independent verification of the 2-form basis functions if the second problem does not yield acceptable results. Cavity problems have smooth solutions and hence discrete p'th order bases should

converge as $O(h^{2p})$ [8]. Under p refinement, exponential convergence should generally be seen.

Another very important characteristic of eigen solutions is that they expose spurious modes that are present in the semi-discretisation. Driven problems can sometimes mask the presence of spurious modes, see e.g. the EBHD waveguide result in [24]. Eigein solutions were also fruitful in discovering the presence of spurious modes in higher order pyramidal elements as described in §8. While eigen solutions are useful during development, the presentation of every permutation of eigen test and semi-discretisation performed will not be presented here. Apart from the semi-discretisations that were found to suffer from spurious modes and are reported in the text, the results would essentially comprise an endless procession of downward-sloping error curves.

9.2 Pulsed Cavity

A validation test closely related to the eigen solution test is to inject a time domain pulse into a cavity [24], [26]. The excitation is designed to excite every mode in the cavity. A measurement is made at a point in the cavity that will pick up all the modes. After performing an FFT on the logged data, resonant peaks centered around the cavity eigen-mode frequencies should appear. This is a good test to perform when a new time-integration scheme is applied to a semi-discretisation that has previously been verified by the eigen-solution. Getting extremely high-resolution frequency-domain information requires careful signal-processing. This, along with the initial verification of the formulations developed in the course of the current research is presented in [24]. The pulsed cavity test is also a good test of stability and conservation; since the cavity is lossless, the resonances should never die down. Furthermore, if a given fully discrete scheme suffers from weak instabilities it show up clearly over an extended cavity run. It also allows the convergence of the time integration to be studied in isolation, since in the limit the resonant peaks should converge to the eigen-frequencies of the semi-discretisation. As for the eigen solutions, no further cavity results will be presented.

9.3 Short-time Waveguide Pulse

Obtaining time domain solutions with analytical solutions that are not trivial but at the same time do not depend on the implementation of peripheral formulations such as domain termination schemes can be a challenge. Using the analytical time-domain transient response derived for waveguide modes in [124], a short-time waveguide pulse simulation can be compared to an analytical solution. Since the field evolution only needs to be measured for a

short time, a length of empty waveguide after the measurement port prevents any reflections from reaching the measurement port within the simulation timeframe. Implementing a sensible numerical experiment is somewhat involved; the reader is referred to the description in [24] for more detail. That same experiment as described in [24] was repeated with a large number of the possible formulation permutations arising out of the current research. Some permutations included:

- All tetrahedral mesh with implicit time integration and various basis orders
- All hexahedral, using consistent $[M]$ matrix integration and various basis orders
- All hexahedral, using diagonalisation
- Implicit/Explicit hybrid using only hexahedral elements
- Implicit/Explicit hybrid with tetrahdra in the implicit region and the waveguide mode launched in the implicit region
- Implicit/Explicit hybrid with tetrahdra in the implicit region and the waveguide mode launched in the explicit region.
- Implicit/Explicit hybrid with tetrahdra in the implicit region and the waveguide mode launched in the implicit region with a PML termination directly after the measurement port.

As for the previous sections, it was not considered worthwhile to include a full result set in this thesis.

9.4 Pulsed Electric Dipole

The time-domain expressions of the fields due to a pulsed electric dipole are fairly simple to obtain, see e.g. [137]. This is a basic verification test for domain truncation schemes. The dipole source is simple to implement, since it can be seen as an infintesimally small current segment; evaluating the integrals in (4.64) and (4.79) with $\vec{J}$ as a Dirac-delta at the dipole location is sufficient. This problem tests that the fully discrete system in combination with domain truncation schemes are working properly. A spherical wavefront should propagate out from the dipole location, and leave the computational domain through the truncation boundary withou reflection.

An example comparing the PML discretised with diagonalised hexahedra to the 1st order ABC discretised with tetrahdra is shown in Fig. 9.2. The analytical solution is shown in Fig. 9.1; a time-derivative Gaussian pulse with a centre frequency of 1 Hz is used as the driving-current time-waveform.

The respective meshes are fairly coarse, with both having a nominal edge length of about $\frac{\lambda}{5}$ at the center-frequency of the pulse. The hexahedral mesh is 2x2x2 m with the dipole at the center. The tetrahdral mesh is a 2m diameter sphere, also with the dipole in the center.

The low-order hexahedral result in Fig. 9.2a shows a fairly poor transient response due to numerical dispersion. However, no spurious $\hat{x}$ and $\hat{y}$ field components are seen due to the exact alignment of the Cartesian hexahedral elements' basis functions with the desired field. The low order tetrahdral mesh result in Fig. 9.2b shows a better transient response, but has significant spurious $\hat{x}$ and $\hat{y}$ field components. As p-refinement is applied, the transient responses improve rapidly, and the the spurious field components in the tetrahdral results are also rapidly reduced.

The main difference between the higher order hexahedral and tetrahedral results are due to the respective PML and 1st-order absorbing boundary condition (ABC) mesh termination schemes used. The PML (as described in §6.3) is in theory capable of absorbing any outgoing wave completely, and its performance is largely determined by discretisation errors. The 1st-order ABC can only absorb only certain outgoing wave-components, implying that even with perfect discretisation some reflection will occur at the mesh boundary. For the lowest order case the absorption is clearly limited by the accuracy of the field discretisation, with both the PML and the 1st-order ABC performing similarly. Both termination schemes improve significantly as the order is increased from mixed 1st to mixed 2nd, but only the PML absorption continues to improve as the order is increased to mixed 3rd. In fact the ABC reflection in the mixed 3rd order case is almost identical to that of the mixed 2nd.

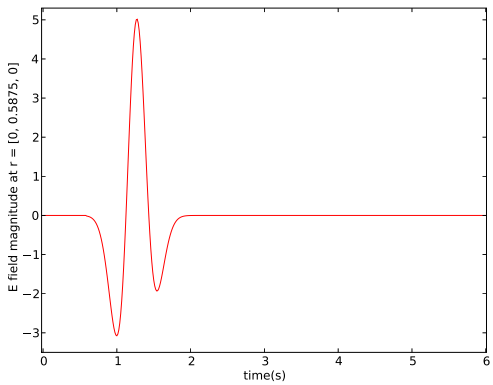
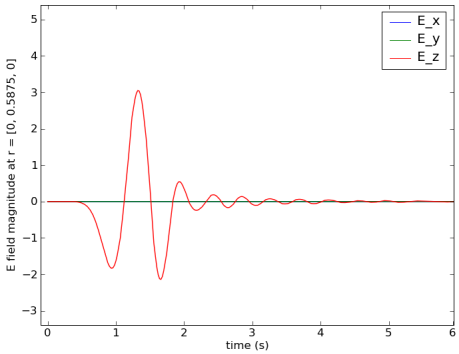
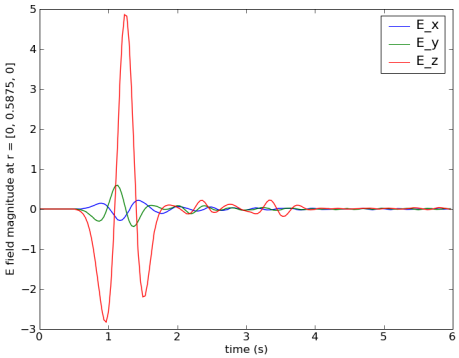


Figure 9.1: Analytical $\hat{z}$-directed pulsed dipole result, z component. The $\hat{x}$ and $\hat{y}$ components are identically zero at the location shown.

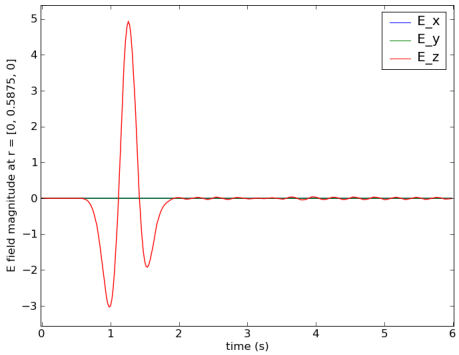
(a) 1st order diagonalised hexahedra, 5 cell PML.



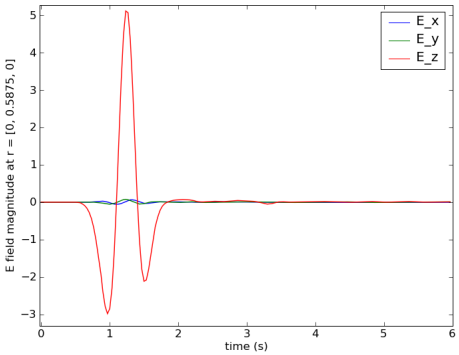
(b) 1st order tetrahedra, 1st order ABC.



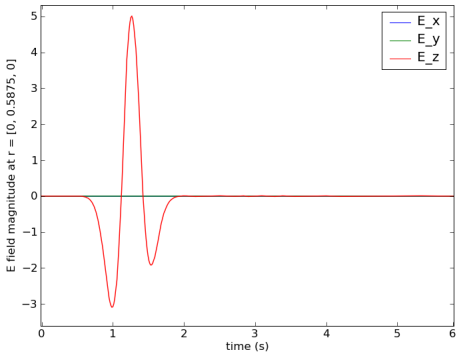
(c) 2nd order diagonalised hexahedra, 5 cell PML.



(d) 2nd order tetrahedra, 1st order ABC.



(e) 3rd order diagonalised hexahedra, 5 cell PML.



(f) 3rd order tetrahedra, 1st order ABC.

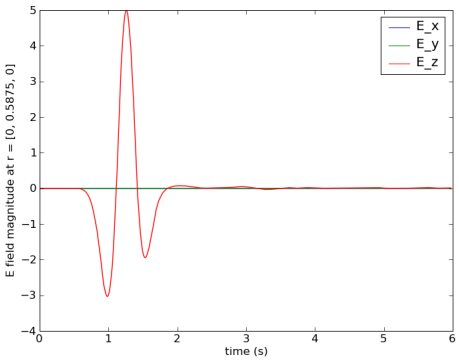


Figure 9.2: Comparison of PML and 1st order ABC as domain truncation using a pulsed electric dipole source.

9.5 Conclusion

Several simple problems that can be used to test the correct functioning of specific aspects of a new numerical method or code have been presented. The problems all have in common that they posses simple analytical solutions, providing accurate and easily calculated reference results.

Chapter 10

Results

In this chapter, problems of practical significance that benefit from the unique characteristics of the higher order hybrid method are solved. The computational efficiency of the new method is compared to that of the classic Yee FDTD and the first order hybrid method. Waveguide problems involving irregular inductive PEC posts are considered.

In the first section, the problem geometries are described in detail. The computational challenges inherent in these geometries and how they may benefit from the features of the higher order hybrid are discussed. In the second section the single conducting half-cylinder computational problem is investigated in detail. The performance characteristics of the numerical methods versus mesh density and approximation order are discussed in detail. The third section compares the performance of the higher order hybrid to published results with two other waveguide configurations using half-cylinder shorting posts; one utilising two posts and another utilising four. The fourth section investigates the performance of several iterative solution schemes for solving the implicit region system matrices. Finally, the conclusions are presented.

10.1 Problem Geometries

All the problem geometries subsequently considered utilise half-cylindrical inductive shorting-posts in WR90 (22.86 x 10.16 mm) waveguide. These posts are proposed [138] as a waveguide design element that is tuneable, avoiding the use of additional tuning components. This element is attractive as a test for the hybrid method since it represents a geometry that would be hard to model using an FDTD grid. The waveguide problem can make good use of the features of the hybrid method, since most of the wavguide is empty with isolated regions of higher geometrical complexity. Unfortunately the paper [138] is very brief, and as noted in [1], it does not seem to include adequate information to repeat the results. Fortunately well speci-

fied geometries and results using the same type of half-cylinder element are presented in [1].

10.1.1 Single Half-Cylindrical Shorting Pin

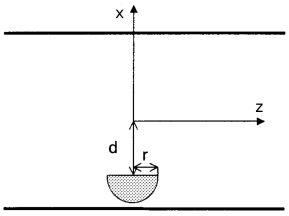


Figure 10.1: Geometric parameters of single half-cylinder pin geometry. Based on [1, Fig. 5], ©[2003] IEEE.

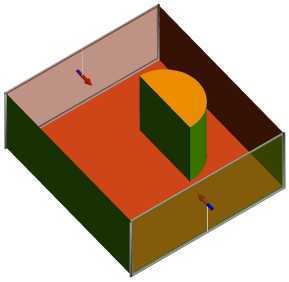


Figure 10.2: 3-D view of single half-cylinder pin geometry with parameters $r = 8$ mm, $d = 3$ mm and the waveguide top removed.

The parametric geometry of the single half-cylindrical pin is shown in Fig. 10.1. A 3-D view of the geometry with parameters used for subsequent calculations is shown in Fig. 10.2. The pin runs parallel to the $\vec{E}$ field polarisation of the TE01 mode. A particular modelling difficulty is the small gap between the shorting pin and the waveguide wall.

10.1.2 Dual Asymmetrical Half-Cylindrical Shorting Pins

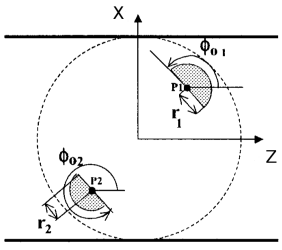


Figure 10.3: Geometric parameters of double half-cylinder pin geometry. Based on [1, Fig. 7], ©[2003] IEEE.

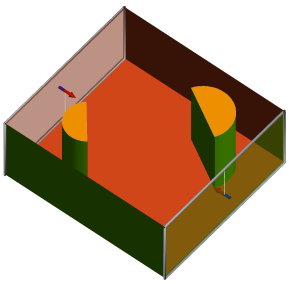


Figure 10.4: 3-D view of double half-cylinder pin geometry with the waveguide top removed. Parametric values are r_1=4 mm, ϕ_1=150°, r_2=3 mm, ϕ_2=310° mm, P_1=(5,5) mm and P_2=(−5, −5) mm.

The parametric geometry of the double half-cylindrical pin is shown in Fig. 10.3. A 3-D view of the geometry with parameters used for subsequent calculations is shown in Fig. 10.4. The pins run parallel to the $\vec{E}$ field polarisation of the TE01 mode. The resonance that occurs between the two pins is very sensitive to geometrical parameter varitions [1].

10.1.3 Four Half-Cylindrical Pin Filter

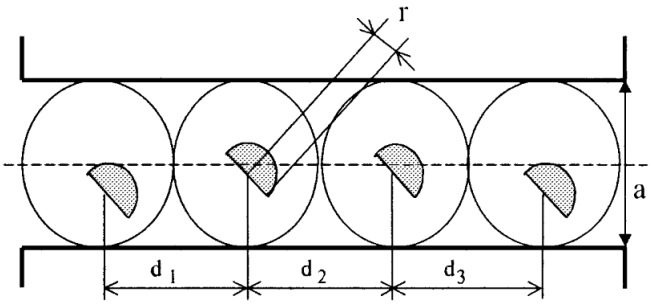


Figure 10.5: Geometric parameters of half-cylinder pin filter geometry. Based on [1, Fig. 4], ©[2003] IEEE.

The parametric geometry of the filter made up of four half-cylindrical pins is shown in Fig. 10.5. A 3-D view of the geometry with parameters used for subsequent calculations is shown in Fig. 10.6. The pins run parallel to the $\vec{E}$ field polarisation of the TE01 mode. This geometry is sensitive to small rotations of the posts [138]. This allows the filter to be tuned to correct for physical tolerances.

10.2 Practical Computational Efficiency

Conducting regular convergence comparisons as done in [24] to test the correct working of the underlying FEM methods is usually not possible with practical problems, since they very seldom have known analytical solutions. Apart from being impossible, it does not really make engineering sense to do so. A more salient comparison is the computational effort required to obtain a solution that is "accurate enough." Of course the value of "enough" is highly application dependent; a specific definition will be presented below.

In this section we first consider the geometry of Fig. 10.1 with $d = 3$ mm and $r = 8$ mm. To create a reference result, a calculation is made using a highly refined hybrid mesh and 3rd order field modelling. This is compared to the published results in [1] and also to a FEKO simulation to confirm its

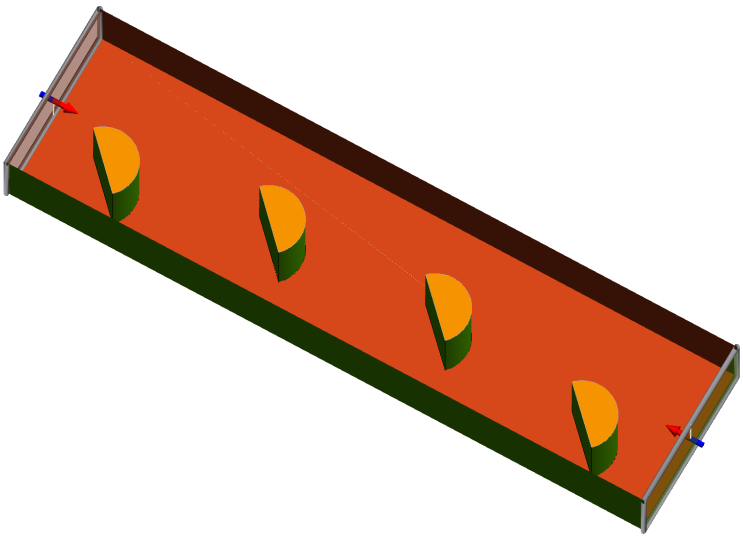


Figure 10.6: 3-D view of half-cylinder pin filter geometry with the waveguide top removed. Parametric values are r=4.41 mm, d_1=21.18 mm, d_2=23.09 mm, d_3=22.20 mm. The offsets of the cylinder-centers from the centerline of the waveguide are from left to right 5.5 mm, 2.2 mm, 2.2 mm and 5.5 mm.

correctness. Now we define "accurate enough" to be when the calculated $|S_{11}|$ result plotted in the frequency range 10–13.1 GHz (where 13.1 GHz is the TE_{02} cutoff frequency) is visually almost indistinguishable from the reference result.

A series of hybrid meshes at increasing levels of refinement is generated, and results calculated using field modelling of order 1 through 3 and Δt ranging from the stability limit for the given mesh/order combination through to a very small value that will be defined subsequently. A calculation is also made using the structured hexahedra and 1st order field modelling that is equivalent to the standard Yee FDTD method for comparison.

Since the main novel contribution of this work is the higher order hybrid mesh implicit/explicit FEM hybrid, it is of great interest to see how the accuracy of the new method compares to existing approaches. The existing approaches considered are the classic Yee FDTD[1] and the FDTD/FEM hybrid mesh hybrid as per Rylander[2] [4]. While the classic Yee FDTD can

[1]As noted in §5.3 the mixed first order explicit PDE I based FEM is numerically identical to the Yee FDTD, and was used to generate the FDTD results here. The results are identical, but the runtime performance would be worse than a dedicated FDTD code. However, all the methods used the same implementation and pay a similar performance penalty.

[2]The results calculated here are not exactly equivalent to Rylander's method, since

probably not be considered state of the art, it is well known and still widely used. Rylander's hybrid, on the other hand, is representative of the current state of the art. A significant improvement in computational efficiency over it would represent a meaningful advancement.

10.2.1 Computational Model Configuration

The TE10 mode is launched at port-1 using the TF/SF bootstrapping method described in §6.1, using the same number of PML cells as for the main simulation. The resulting TE10 waveguide response is recovered using the mode-integral method from §6.2. The $\vec{E}$-field at port-1 is actually in the total field region, but since the incident field is known, the scattered response can simply be recovered by subtracting the incident field from the measured response. The computational domain is terminated by a 30-cell PML on both sides, using the standard cubic profile recommended in the FDTD literature [6]. The beginning of the PML region on the port-2 side coincides with port-2, while the TF/SF source formulation requires at least one cell of freespace between the PML and the source port. Since the standard PML does not absorb evanescent waves [5, §6.2], the buffer space between the ports and the PML need to be long enough to allow the higher order modes excited by the waveguide device to decay. With the meshes with a finer cell spacing than $\frac{a}{16}$ it was found neccesary to add extra freespace[3] between the ports and the PML to prevent the evanescent modes from affecting the higher frequency results. This resulted in a fairly large additional computational cost.

The nominal hexahedral cell-sizes were chosen as fractions of a. Since an integral number of cells have to fit in each dimension of the waveguide, the cells in practice have slightly different x (a) and y (b) grid steps. Hybrid meshes with hex cell-sizes of $\frac{a}{3}$, $\frac{a}{4}$, $\frac{a}{8}$, $\frac{a}{16}$, $\frac{a}{32}$ and $\frac{a}{64}$ were set up. FDTD meshes were set up at the same cell spacings with the addition of $\frac{a}{11}$, $\frac{a}{21}$, $\frac{a}{48}$, $\frac{a}{95}$ and $\frac{a}{128}$. Details of the meshes are discussed further in the next subsection. In order to balance the error between the field discretisation and the time integration, solutions using higher than first order field representation often require Δt to be smaller than the minimum value required for stability. Hence, a series of Δt ranging from the stability limit for each mesh and field order combination down to about 2e-13 s was considered.

The implicit system matrix was solved using umfpack [139], a sparse direct LU solver. The included automatic matrix re-ordering routines seem to work *very* well on the problems considered here, resulting in relatively

Rylander uses pyramidal elements to connect the hexahedral and tetrahedral meshes. However it is reasonable to expect the performance of the present hybrid mesh hybrid using mixed first order field representation to be largely similar to Rylander's hybrid.

[3]Since a fixed number of PML cells are used, reducing the cell spacing results in a physically shorter PML region.

small fill-in values. Iterative solvers were also investigated, but were found to be less than optimal for these particular problems. This is discussed below in 10.4.

10.2.2 Meshing Strategies

When defining a regular hexahedral mesh, the cell size is the only parameter that can be adjusted. However, when dealing with a geometry that does not conform to the grid boundaries, several choices of stair-case approximation are possible. In common with [4] it was found that the best and most regularly convergent results were obtained by considering a hex-cell to be PEC if *all* of its nodes are inside the PEC geometry. Some hexahedral approximations to the pin geometry are shown in Fig. 10.7.

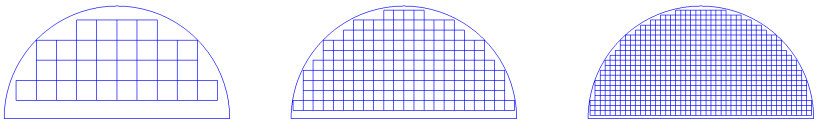


Figure 10.7: Top view of hexahedral approximations of the pin geometry. From left to right, h16, h32, h64.

When setting up the hybrid mesh, the tetrahedral region clearly allows infinite variety. One of the great benefits of tetrahedral meshes is their ability to be refined locally. In light of this, two meshes were created for each hexahedral mesh size up to $\frac{a}{8}$. In the first case, the tetrahedral mesh was created with a nominal edge length set equal to that of the hexahedral mesh. In the second locally refined case, the nominal tetrahedral edge length was set to be 1 mm on the circular part of the post, and also on a section of the waveguide wall at its closest point to the pin. The back face of the pin was meshed to 2 mm. Some refined and unrefined meshes are shown in Fig. 10.8.

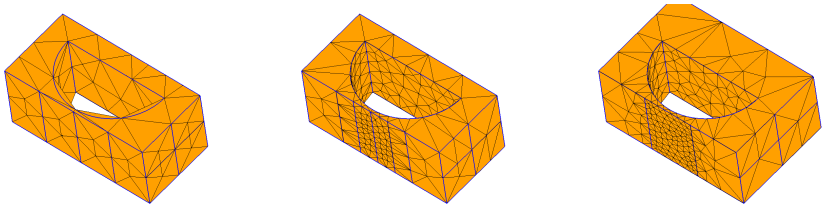


Figure 10.8: From left to right, unrefined h4 mesh, h4_refined mesh and h3_refined mesh. Note the significantly larger tetrahedral region for the h3_refined mesh due to the larger hexahedra.

The following naming scheme is used:

hX Indicates that the explicit region is meshed with Cartesian hexahedra with a nominal edge length of $\frac{a}{X}$, where a is the waveguide cross-section. The hexahedral edge-length will be slightly smaller in the y-axis, since the height of the waveguide is not a multiple of the width. The tetrahedra in the implicit region is also meshed with a nominal edge-length of $\frac{a}{X}$.

hX_refined The explicit region is meshed as for hX. The tetrahedral mesh still has a nominal edge-length of $\frac{a}{X}$, but the mesh has been locally refined along the post surface as described in the previous paragraph.

hX_refhY The hexahedral mesh is created with a nominal edge-length of $\frac{a}{X}$, and the tetrahedral region with a nominal edge-length of $\frac{a}{Y}$.

10.2.3 Reference Result

A reference result was calculated using the h16 mesh and mixed 3rd order field modelling. Another reference result was calculated using FEKO [140]. These calculated results are superimposed over the result from [1]. Excellent agreement is noted between all methods.

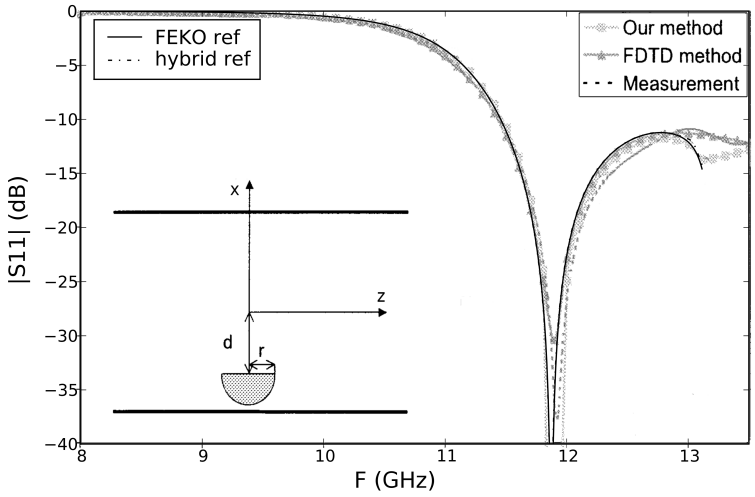


Figure 10.9: Reference result for r=8 mm, d=3 mm half cylindrical pin. Based on [1, Fig. 5], where the r=5 mm results have been removed. The right caption is from [1], the left caption is the results calculated by the current author. The "hybrid ref" result was calculated using the $\frac{a}{16}$ hybrid mesh with Δt=6.25e-5. Excellent agreement is noted.

10.2.4 Convergence Study

The convergence of the FDTD equivalent method is shown in Fig. 10.10, and in Fig. 10.11 the mixed 1st order hybrid mesh result is shown. As is the case for the FDTD, having Δt at the stability limit achieves the best results with the mixed 1st order hybrid mesh. The use of the locally refined meshes made no appreciable difference to the hybrid mesh and is not shown. This indicates that the field and geometrical modelling errors are balanced in this case. It is clear that the FDTD grid is unable to model the pin geometry effectively, requiring the very refined h95 mesh to provide even reasonable accuracy. The convergence is also somewhat erratic, with, for example, the h21 mesh being more accurate than the h32 mesh. The hybrid mesh is clearly much more accurate, with the hybrid h8 mesh being on par with the FDTD h32 mesh. The accuracy criteria are met by the h128 FDTD grid and the h64 mixed 1st order hybrid mesh respectively.

The convergence of the mixed 2nd order hybrid mesh is shown in Fig. 10.12, and in Fig. 10.13 the mixed 2nd order locally refined hybrid mesh result is shown. The 2nd order results are notably more accurate than the 1st order results. Even the coarse h4 mesh with only 4 elements accross the waveguide's width delivers reasonable results, while the h8 result meets the accuracy criteria. Compared to the 1st order hybrid mesh, which required the h64 mesh to reach the accuracy criteria, the difference is quite stark. The local mesh refinement does not have a large effect; the h4_refined result is slightly better than the h4 results around 12.5 GHz, but the difference is not large enough to make it worthwhile.

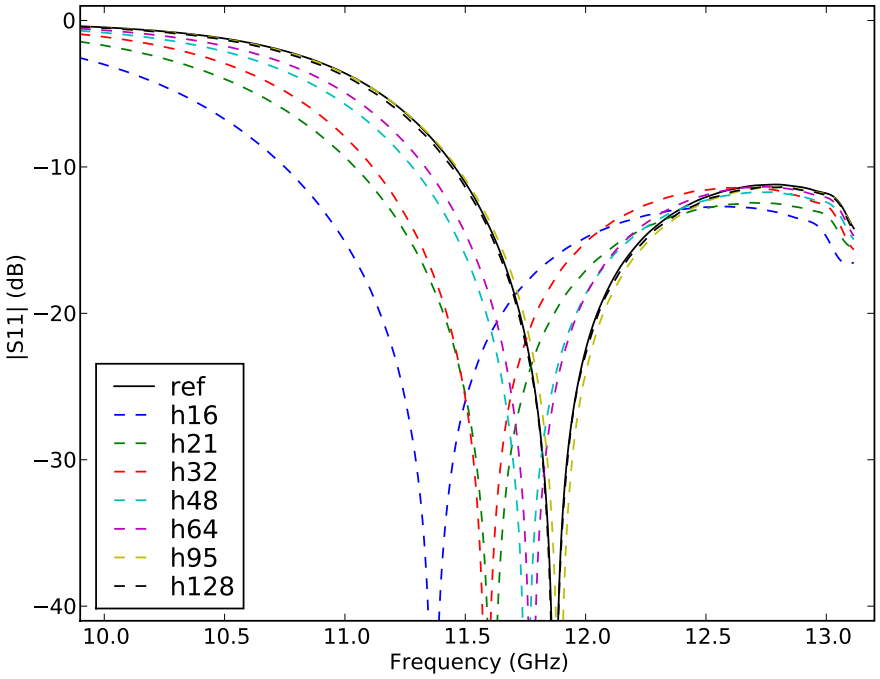


Figure 10.10: Convergence of FDTD equivalent mixed 1st order hexahedral mesh. with Δt at the stability limit. The "ref" caption indicates the reference result.

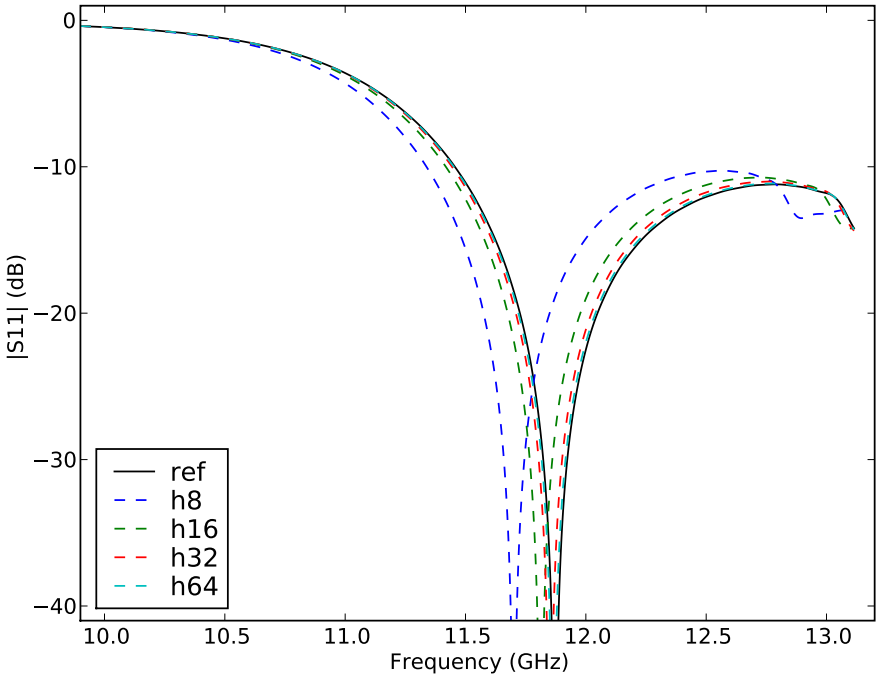


Figure 10.11: Convergence of mixed 1st order hybrid mesh with Δt at the stability limit. The "ref" caption indicates the reference result.

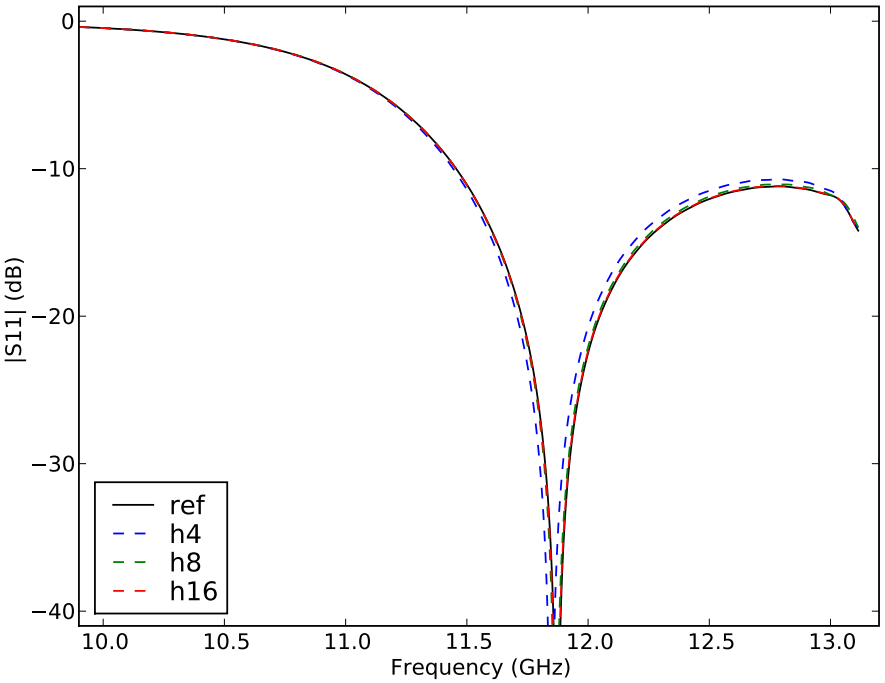


Figure 10.12: Convergence of mixed 2nd order hybrid mesh.

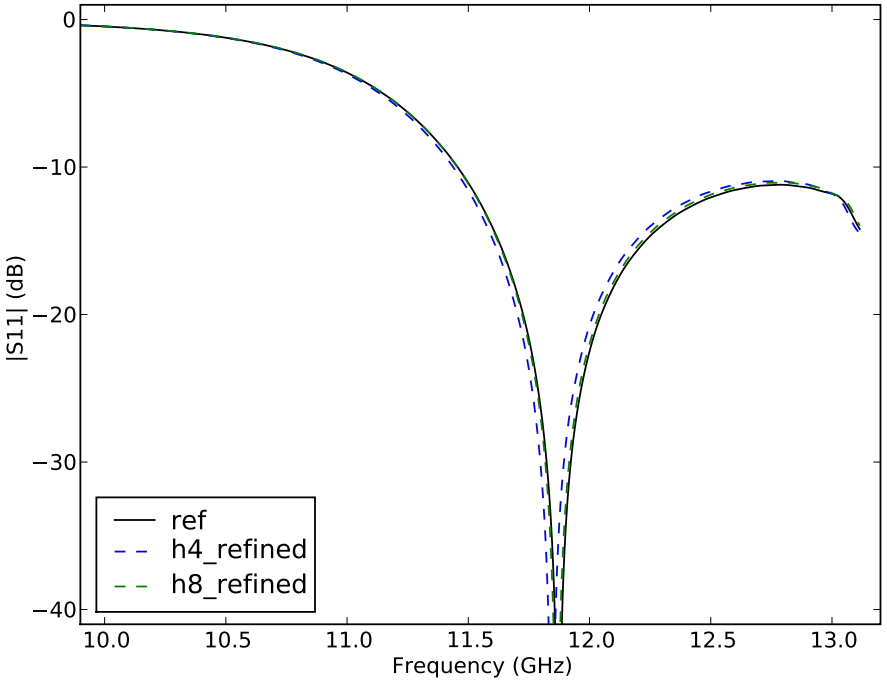


Figure 10.13: Convergence of mixed 2nd order locally refined hybrid mesh.

The convergence of the the mixed 3rd order hybrid mesh is shown in Fig. 10.14, and in Fig. 10.15 the mixed 3rd order locally refined hybrid mesh result is shown. All the third order methods are very accurate, with even the unrefined h3 and h4 meshes almost meeting the accuracy criteria; the h3 and h4 results only diverge slightly at the higher frequencies. On the other hand, all the refined meshes meet the criteria. In spite of the h3 mesh being nominally less finely discretised, the larger hexahedral cells lead to a larger volume discretised by tetrahedra; this is shown in Fig. 10.8. Since the CPU time is dominated by the implicit solve, the h4 mesh ends up solving faster.

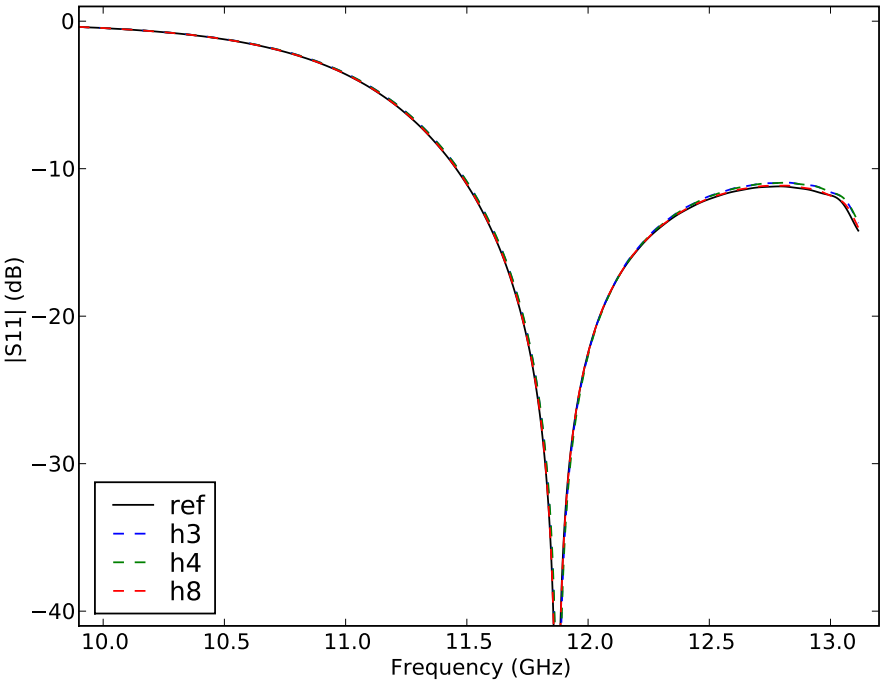


Figure 10.14: Convergence of mixed 3rd order hybrid mesh.

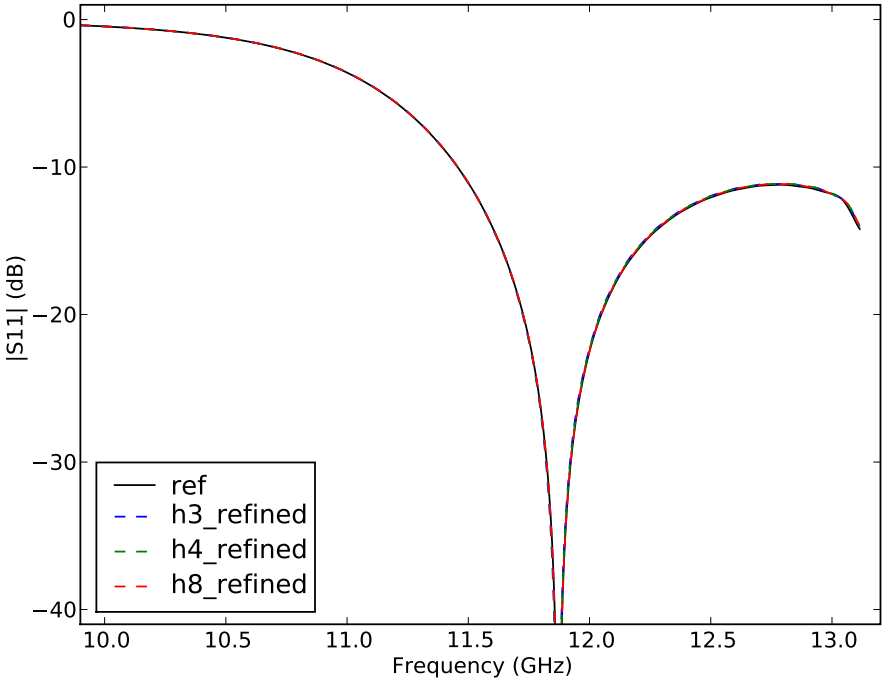


Figure 10.15: Convergence of mixed 3rd order locally refined hybrid mesh.

10.2.5 A modified result

The results in the previous subsection suggest that the accuracy is not overly sensitive to local mesh refinement around the pin. The homogenous h8 mesh provides adequate geometric modelling while mixed second order modelling in the tetrahedral region provides sufficient field modelling. The mixed 2nd order refined h4 result in Fig. 10.13 indicates that even with sufficient geometrical modelling around the pin, the propagation error induced by the 2nd order h4 explicit mesh is unacceptable.

Using the fact that the tetrahedral elements are hierarchical and the ability of the hex-tet interface to work with an over-discretised triangulation, an attempt can be made at constructing a more efficient solution. The mixed 3rd order h4 explicit mesh is combined with a mixed 2nd order tetrahedral mesh in an attepmpt to gain a beter overall efficiency. The optimised solution is shown in Fig. 10.16. The optimised result is compared to the previous

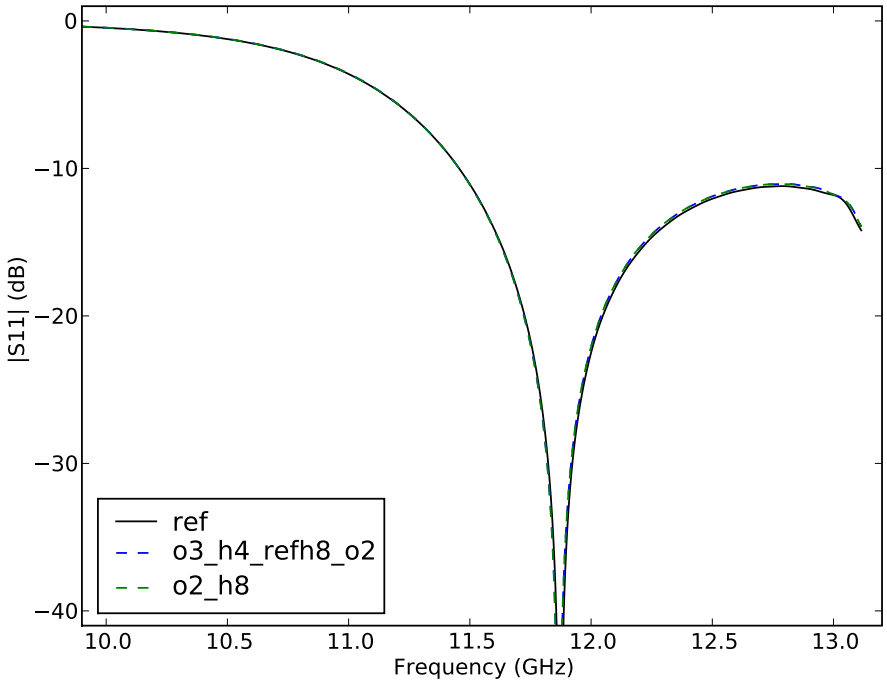


Figure 10.16: Convergence of mixed 3rd order in hexahedral region and mixed 2nd order in the tetrahedral region.

all 2nd order h8 result. They agree more or less exactly, indicating that the optimised result also satisfies the accuracy criteria. However, the timings in Table 10.1 show that the CPU run-times of both the modified discretisation

with 3rd order hexahedra and the original consistently 2nd order result are the same within measurement tolerance. This is due to the solution of the implicit system matrix dominating the CPU time use. While the modified discretisation is not in this instance a win, it does demonstrate that the flexible nature of the hybrid mesh combined with hierarchical tetrahedra should make it well suited to the application of *p*- or *hp*-adaptive methods.

10.2.6 Computational Efficiency

Solver Type	Mesh	Runtime	Implicit Runtime %	Improvement
FDTD Equiv	h128	58h14	0	n/a
$p = 1$ hyb	h64	6h43	19.4	8.7
$p = 2$ hyb	h8	28m20	53.5	123
$p = 3$ hyb	h4_refined	1h24	81.9	41.4
$p = 3/p = 2$ hyb	h4_refh8	28m35	71.5	122

Table 10.1: Computational efficiency of various solutions.

10.3 Further Comparisons

To further validate the accuracy of the current method, it is compared to the published results of the two- and four-pin geometries described in §10.1.2 and §10.1.3. The comparative results are shown in Fig. 10.17 and Fig. 10.18. Excellent agreement between all methods is seen in Fig. 10.17 for the the two-pin geometry. The measured data in Fig. 10.17 differs significantly from all the computed results, although in [1] it is mentioned that the geometry is very sensitive to manufacturing tolerances. The current hybrid method utilised a locally refined mesh with discretisation parameters equivalent to the h8_refined mesh described in §10.2.2.

There seems to be more of a discrepancy with the four-post filter result shown in Fig. 10.18. The FDTD and special method results from [1] do not agree that well to start with. The close agreement between the FEKO result and the current hybrid method seems to suggest that the dimensions published in [1] might not be precise enough. This particular geometry seems to be very sensitive to small geometric variations. The sensitivity was reflected in the level of mesh discretisation that was required around the posts to provide convergence. The result shown utilises a mesh similar to the h8_refined mesh described in §10.2.2, with the exception that the

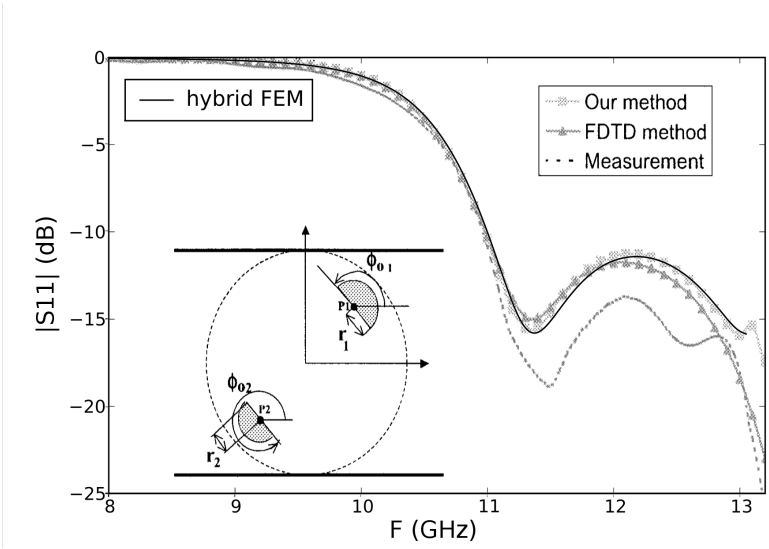


Figure 10.17: Results for the problem geometry in Fig. 10.2. Based on [1, Fig. 7], ©[2003] IEEE.

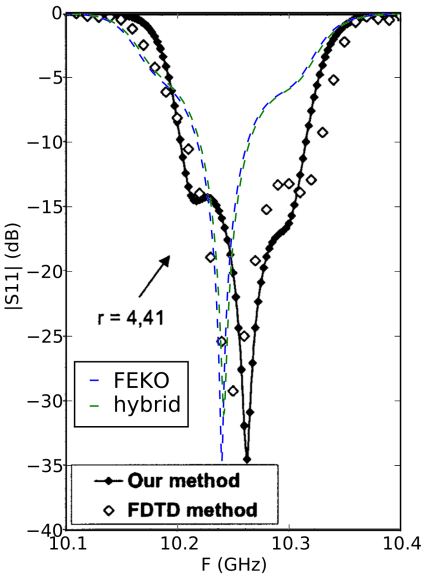


Figure 10.18: Results for the problem geometry in Fig. 10.6 [1, Fig. 4], ©[2003] IEEE.

surface refinement was twice as fine. The tetrahedral part of the hybrid mesh geometry is shown in Fig. 10.19

10.4 Iterative Solution

Iterative solvers are often used with FEM codes as matrix solvers. Due to the highly sparse nature of FEM matrices, iterative solution is frequently the best option. All else being equal, iterative solvers are usually most competetive on very large systems. Their efficiencies also depend critically on the conditioning of the matrix equations. The iterative solution was tested by saving the system matrix as well as representative RHS and previous time-step value vectors. The previouse time-step vector is used as a starting guess for iterative solution.

10.4.1 Preconditioners and Solvers Evaluated

Several preconditioners were accessed via the petsc4py bindings to PETSc [141, 142, 143] libraries, including preconditioners from [144]. A very common preconditioner for symmetric matrices is the incomplete LU decompo-

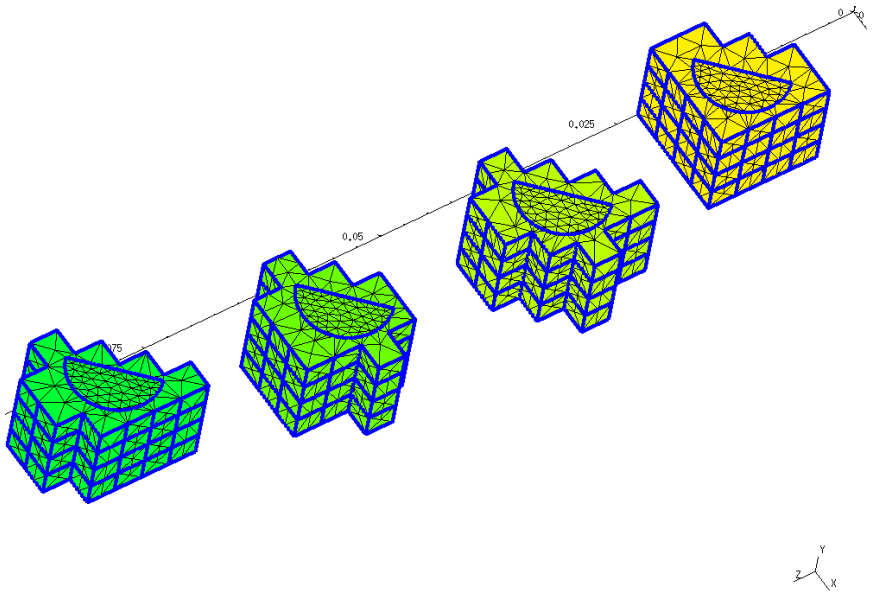


Figure 10.19: Hybrid mesh of the half-cylinder pin filter geometry with the waveguide top removed. Parametric values are r=4.41 mm, d_1=21.18 mm, d_2=23.09 mm, d_3=22.20 mm. The offsets of the cylinder-centers from the centerline of the waveguide are from left to right 5.5 mm, 2.2 mm, 2.2 mm and 5.5 mm. The blue lines indicate the boundaries of the hexahedral faces where they meet the tetrahedral mesh. Note that no triangle edges cross any blue lines.

sition (ILU). For symmetric positive definite (SPD) matrices such as $[A]$, the closely related incomplete Choleskey decomposition (ICC) results in a factor 2 improvement in preconditioner storage and computational cost over the ILU.

For the ILU/ICC preconditioners, a fill-in level can be set. ILU0 has zero fill-in; i.e. the preconditioner has the same sparsity structure as the original matrix. The ICC0 preconditioner is similar, but due to symmetry it will have half as many entries as the original. Successive levels of fill-in, ICC1, ICC2, ... result in larger fill-in factors relative to the original matrix in exchange for better preconditioning performance. At some stage the computer's memory will be exhausted; otherwise a diminishing or even negative level of performance return is reached due to the expense of applying the large preconditioner. The ILU0/ICC0 preconditioners are frequently used since they are cheap to calculate and almost always provide a useful benefit.

A fairly sophisticated preconditioner is the Algebraic Multi-Grid (AMG) method implemented by BoomerAMG [145]. Algebraic multi-grid solvers aim to attain multi-grid like performance on unstructured systems; it is perfectly usable as a "black box" preconditioner with any matrix.

Preconditioners need to be combined with a sparse solver. For SPD systems, the standard Conjugate Gradients (CG), or when used with a preconditioner, Preconditioned Conjugate Gradients (PCG) method is usually optimal. For the matrices considered here, the preconditioner was the largest factor in solution speed; the Bi-Conjugate Stabilised (BiCG-Stab) and Successive Over Relaxation (SSOR) methods yielded similar timings, but the CG method was slightly faster.

10.4.2 Properties of the Matrix Equations

Due to the use of the hybrid mesh, the implicit system matrices considered here are rather small. Furthermore, the requirement for local refinement around the waveguide shorting posts resulted in the system matrices being rather "stiff." This is another term with origins in the computational structural dynamics field. The $[S]$ matrix that applies the discrete curl-curl operator is similar to the stiffness matrix in computational structural dynamics. This matrix is singular due to the large null-space of the curl-curl operator when using $H(curl)$ discrete spaces. In (4.4.3) we see that the $[A]$ matrix is a linear combination of $[M_\epsilon]$ and $[S]$. The larger the relative contribution of $[S]$, the closer to singular, and hence the more ill-conditioned $[A]$ becomes. The "stiffness" of $[A]$ of $[S]$ is determined by the level of discretisation relative to Δt; the finer the mesh or the larger Δt, the stiffer $[A]$ becomes. Increasing the discretisation order p is also equivalent to refining the mesh in stiffness.

10.4.3 Preconditioner Timings

Iterative timings for the solution of various implicit system matrices are presented in the following tables. The preconditioner "umf" refers to the use of UMFPACK to solve the matrices by means of a sparse LU decomposition. The "LU time" field was the time taken for the LU decomposition. The preconditioners used are labeled:

none The use of conjugate gradients without a preconditioner.

icc(0123) Incomplete Choleskey decomoposition with fill-levels of 0,1,2 or 3.

icc(0123)pd Same as Incomplete Choleskey decomoposition above, but with the "pc_factor_shift_positive_definite=1" parameter passed to PETSc. This was needed to avoid "Preconditioner indefinite" errors.

amg BoomerAMG Algebraic Multigrid preconditioner. Fill-in is not listed since BoomerAMG does not report memory use information. By looking at process size while the preconditioner was running it was concluded that it usually uses an amount of memory between the icc0 and icc1 case.

The "Fill-in factor" column describes the memory use of the preconditioner relative to the original matrix. A factor of 1 implies that the preconditioner is the same size as the original matrix, implying a total memory cost of double that of the original matrix. Some general observations

Preconditioner	Fill-in factor	Solution time	LU time
umf	7.9	0.329	2.74
none	0	0.795	
icc0	0.53	0.415	
icc1	1.02	0.410	
icc2	1.66	0.384	
icc3	2.48	0.357	
amg		2.97	

Table 10.2: Matrix solution timings for the implicit system matrix of the $p = 1$, h64 solution in Table 10.1, with dimension 66767 and 1.2e6 nonzeros

are that the iterative solvers are only competitive with UMFPACK when $p = 1$ discretisation is used. This seems to indicate that unless the limitation is memory use, UMFPACK is preferred. The incomplete Choleskey

Preconditioner	Fill-in factor	Solution time	LU time
umf	4.7	0.0397	0.302
none	0	0.559	
icc0pd	0.51	0.954	
icc1pd	1.55	0.647	
icc2pd	3.31	0.548	
icc3	5.70	0.177	
icc3pd	5.70	0.162	
amg		0.587	

Table 10.3: Matrix solution timings for the implicit system matrix of the $p = 2$, h8 solution in Table 10.1, with dimension 7284 and 293384 nonzeros

Preconditioner	Fill-in factor	Solution time	LU time
umf	4.42	0.272	2.73
none	0	46.7	
icc0pd	0.505	92.9	
icc1pd	1.65	120	
icc2pd	3.66	27.3	
icc3pd	6.42	334	
amg		7.39	

Table 10.4: Matrix solution timings for the implicit system matrix of the $p = 3$, h8 solution (not shown in Table 10.1), with dimension 22089 and 2.3e6 nonzeros

preconditioners with a fill-in level of more than 0 are occasionally numerically unstable. Also is also noted that the BoomerAMG preconditioner is uncompetitive with small systems, but as the system size and discretisation order increases it's relative performance improves. What is quite attractive about BoomerAMG is its low memory overhead.

10.5 Conclusion

The results show that the higher order hybrid mesh implicit/explicit hybrid FEM technique developed in this thesis improves the efficiency of time domain EM simulations. Taking an FDTD solution as the baseline, compar-

Preconditioner	Fill-in factor	Solution time	LU time
umf	4.33	0.181	1.83
none	0	28.9	
icc0pd	0.507	18.4	
icc1pd	2.01	97.7	
icc2pd	4.96	16.9	
icc3pd	9.05	2.40	
amg		4.53	

Table 10.5: Matrix solution timings for the implicit system matrix of the $p = 3$, h4_refined solution in Table 10.1, with dimension 20997 and 1.5e6 nonzeros

Preconditioner	Fill-in factor	Solution time	LU time
umf	6.9	2.54	51.6
none	0	368	
icc0pd	0.506	117	
icc1pd	Attempted to allocate 1.84e18 bytes of memory		
icc2pd	4.89	3416	
icc3pd	Attempted to allocate 1.84e18 bytes of memory		
amg		68.02	

Table 10.6: Matrix solution timings for the implicit system matrix of the $p = 3$ and mesh shown in Fig. 10.19 with dimension 184737 and 16.2e6 nonzeros

ative timings for the waveguide with half-cylindrical shorting posts indicate a speedup factor of 123 over the FDTD and a speedup of factor 14 over a previous mixed first order hybrid mesh approach. The best efficiency was obtained by using mixed second order elements in the tetrahedral region. The geometrical modelling error made by rectilinear tetrahedra seem to limit the computational efficiency achievable by the third order elements. However, the ability of the hybrid mesh method presented here to incorporate both inhomogeneous h and inhomogeneous p refinement indicates the application of p- or hp-adaptive methods as future work. Another complementary approach would be the use of curvilinear elements in the tetrahedral region to improve the geometrical modelling.

The solution of the implicit region system matrices by iterative methods was also evalued. It was found that in general UMFPACK was an order of magnitude faster than the iterative solvers, indicating that it is the best choice if sufficient memory is available. The BoomerAMG preconditioner shows good performance characteristics relative to the incomplete Choleskey decomoposition as the problem size increases while using a relatively modest amount of memory. The iterative results indicate that it would be beneficial to research methods than can improve the conditioning of the implicit system matrices.

Chapter 11

Conclusions and Future Work

In this thesis, a novel hybrid time-domain FEM method that is suitable for RF and microwave electromagnetics problems is presented. Developing the novel method required the use of much disparate supporting theory and prior art. A significant portion of this thesis is devoted to putting the needed theory and prior art into a cohesive framework. This framework enables a clear description of the novel method and its development. The aim of the novel method is to improve the computational efficiency of existing methods by the combination of the following features and their respective advantages:

Higher Order Field Modelling Higher order modelling is highly beneficial when very accurate solutions are required. It also addresses the issue of numerical dispersion when solving electrically large problems.

Hybrid Implicit/Explicit Operation Explicit operation is very computationally efficient since no matrices have to be inverted. In regions of complex geometry, it may result in an excessively small time-step size, outweighing the savings of avoiding matrix inversion. Implicit methods do not suffer from reduced time-step sizes when dealing with complex geometries. In hybrid operation, only small parts of the problem are treated with an implicit method as needed. The efficiency benefits of explicit operation through most of the domain are accrued without being limited by the time-step constraints in regions of complex geometry.

Hexahedral/Tetrahedral Hybrid Discretisation Hexahedral elements are required for explicit operation, but they suffer from poor geometrical modelling properties. Tetrahedra excell at geometrical modelling, but are computationally more expensive. By using tetrahedra only in regions with complex geometry, the most computationally efficient combination is realised.

The combination of these features in an electromagnetic FEM method is, to the author's best knowledge, a first. The explicit portion of this method can be seen as a higher order generalisation of the FDTD. This close relationship allows many of the peripheral FDTD methods to be adapted to the current method. This was exploited to derive an explicit, convolution free PML method for domain truncation, and a total field/scattered field formulation for waveguide analysis.

The current method is first validated using simple problems that have well known and tractable solutions. With the knowledge that the basic method works, more challenging benchmark problems are solved. The performance of the current method was compared to related existing methods. Realtive to an FDTD solution of the same accuracy, the current method achieved a speed-up factor of over 100 on the bencmark problem considered. As compared to a previous low-order implicit/explicit hybrid method, a speed-up factor of 14 was achieved. The improved computational efficiency witnessed indicates that the new method represents a significant improvement over previous methods.

11.1 Research Contributions

The research contributions made in this thesis fall into two categories. The first is the systematic synthesis of disparate existing electromagnetic, mathematical and finite elements theory into a consistent whole. The second is the formulation of novel methods, or the novel extension of existing methods. Contributions in the first category include:

- A comparative description of Maxwell's systems in the language of standard Gibbsian vector algebra, and in the language of differential forms in §3.
- A cohesive summary of all the FEM theory needed to develop the current hybrid formulation and its peripheral support formulations in §4.1-4.2 and §6.
- The systematic development of the coupled Maxwell's equations and the vector Helmholtz equation into two fully discrete systems in §4.3-4.5.
- A survey of the FEM discretisation techniques required for electromagnetic simulation in §5. The general properties of FEM discretisations are summarised, and their implications for the current method discussed.

Contributions in the latter category include:

- The extention of the FDTD total field/scattered field formulation to the higher order explicit FEM formulation for waveguide calculations in §6.1.
- The extention and analysis of the FDTD convolution-free UPLM implementation to the higher order explicit FEM formulation in §6.3.
- The extention of the implicit/explicit Newmark-β/leapfrog time stepping scheme to a form applicable to arbitrary order FEM field discretisation, and the description of the block-diagonal system that results in §7.
- A novel arbitrary order curl-conforming hexahedral/tetrahedral hybrid mesh discretisation in §8.

11.2 Future Work

Suggestions for future work include:

- The implementation of curvilinear elements, allowing more effective use of the modelling capabilities of the higher order tetrahedra by largely eliminating geometrical modelling error. It may also help to improve the conditioning of the implicit system matrices by avoiding the need to over-discretise around curved geometries.
- Researching alternative tetrahedral basis sets in order to improve the conditioning of the implicit system matrices, including the use of tree-cotree gauging.
- Investigating the optimal choice of iterative solver and preconditioners for solving the implicit system matrices.
- Exploiting the structured nature of the explicit system in order to optimise its implementation.
- Investigating the use of more sophisticated frequency analysis techniques to reduce the simulated time requirement for narrow-band systems.
- Multiple implicit regions result in multiple uncoupled implicit systems. The benefits of this to parallel operation merits investigation.
- Optimal and automatic hybrid mesh generation. The hybrid formulation is very flexible, and in principle multiple implicit and multiple explicit domains can exist. For instance, a part of the interior region of a penetrable scatterer can be meshed with hexahedra.

- Research into the formulation of an implicit/explicit hybrid with higher order time integration schemes [146].

Bibliography

[1] R. Lech, M. Polewski, and J. Mazur, "Scattering in junction by posts consisting of a segment of conducting cylinder," *IEEE Trans. Microwave Theory Tech.*, vol. 51, 2003.

[2] D. B. Davidson, *Computational Electromagnetics for RF and Microwave Engineering*. Cambridge, UK: Cambridge University Press, 2005.

[3] T. Rylander and A. Bondeson, "Stable FEM-FDTD hybrid method for Maxwell's equations," *Computer Physics Communications*, vol. 125, pp. 75–82, 2000.

[4] T. Rylander and A. Bondeson, "Stability of explicit-implicit hybrid time-stepping schemes for Maxwell's equations," *Journal of Computational Physics*, vol. 179, pp. 426–438, July 2002.

[5] J. Pierre Berenger, "Perfectly matched layer (PML) for computational electromagnetics," *Synthesis Lectures on Computational Electromagnetics*, vol. 8, 2007.

[6] A. Taflove and S. C. Hagness, eds., *Computational Electrodynamics The Finite-Difference Time-Domain Method*. Artech House, 3rd ed., 2005.

[7] M. Bingle, D. Davidson, and J. Cloete, "Scattering and absorption by thin metal wires in rectangular waveguide-FDTD simulation and physical experiments," *IEEE Trans. Microwave Theory Tech.*, vol. 50, pp. 1621–1627, 2002.

[8] J. P. Webb, "Hierarchal vector basis functions of arbitrary order for triangular and tetrahedral finite elements," *IEEE Trans. Antennas Propagat.*, vol. 47, pp. 1244–1253, Aug. 1999.

[9] G. S. Warren and W. R. J. Scott, "Numerical dispersion of higher order nodal elements in the finite-element method," *IEEE Trans. Antennas Propagat.*, vol. 44, pp. 317–320, 1996.

[10] J. C. Nédélec, "Mixed finite elements in $\Re^3$," *Numerische Mathematik*, vol. 35, pp. 315–341, 1980.

[11] J. C. Nédélec, "A new family of mixed finite elements in $\Re^3$," *Numerische Mathematik*, vol. 50, pp. 57–81, 1986.

[12] T. J. R. Hughes, *Analysis of Transient Algorithms with Particular Reference to Stability Behaviour*, vol. 1: Computational Methods for Transient Analysis of *Computational Methods in Mechanics*, pp. 67–155. Amsterdam: North-Holland, 1983. eds. T. Belytschko and T. J. R. Hughes.

[13] T. J. R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, 1987.

[14] O. C. Zienkiewicz, "A new look at the Newmark, Houboult and other time stepping formulas. a weighted residual approach," *Earthquake engineering and structural dynamics*, vol. 5, pp. 413–418, 1977.

[15] A. C. Cangellaris and D. B. Wright, "Analysis of the numerical error caused by the stair-stepped approximation of a conducting boundary in FDTD simulations of electromagnetic phenomena," *IEEE Trans. Antennas Propagat.*, vol. 39, pp. 1518–1525, 1991.

[16] R. Holland, "Pitfalls of staircase meshing," *IEEE Trans. Electromagn. Compat.*, vol. 35, pp. 414–439, 1993.

[17] G. van Rossum *et al.*, "Python." Available: `http://www.python.org`.

[18] T. E. Oliphant, *Guide to NumPy*. USA: Trelgol Publishing, 2006. Available `http://www.tramy.us/`.

[19] T. E. Oliphant, "Python for scientific computing," *Computing in Science and Engineering*, vol. 9, pp. 10–20, 2007.

[20] "SciPy - Scientific tools for Python." Available: `http://www.scipy.org/`.

[21] J. Jin, *The Finite Element Method in Electromagnetics*. John Wiley and Sons, 2 ed., 2002.

[22] J.-F. Lee, R. Lee, and A. Cangellaris, "Time-domain finite-element methods," *IEEE Trans. Antennas Propagat.*, vol. 45, pp. 430–442, Mar. 1997.

[23] J.-F. Lee and Z. Sacks, "Whitney elements time domain (WETD) methods," *IEEE Trans. Magn.*, vol. 31, pp. 1325–1329, May 1995.

[24] N. Marais and D. Davidson, "Numerical evaluation of high-order finite element time domain formulations in electromagnetics," *IEEE Trans. Antennas Propagat.*, vol. 56, pp. 3743 – 3751, Dec. 2008.

[25] M.-F. Wong, O. Picon, and V. Fouad Hanna, "A finite element method based on Whitney forms to solve Maxwell equations in the time domain," *IEEE Trans. Magn.*, vol. 31, pp. 1618–1621, May 1995.

[26] D. A. White, *Discrete Time Vector Finite Element Methods for Solving Maxwell's Equations on 3D Unstructured Grids*. PhD thesis, Lawrence Livermore National Laboratory, 1997.

[27] H.-P. Tsai, Y. Wang, and T. Itoh, "An unconditionally stable extended (USE) finite-element time-domain solution of active nonlinear microwave circuits using perfectly matched layers," *IEEE Trans. Microwave Theory Tech.*, pp. 2226–2232, Oct. 2002.

[28] A. Monorchio and R. Mittra, "A hybrid finite-element finite-difference time-domain (FE/FDTD) technique for solving complex electromagnetic problems," *IEEE Microwave Guided Wave Lett.*, vol. 8, pp. 93–95, Feb. 1998.

[29] D. Degerfeldt and T. Rylander, "A brick-tetrahedron finite-element interface with stable hybrid explicit-implicit time-stepping for Maxwell's equations," *Journal of Computational Physics*, vol. 220, pp. 383–393, Dec. 2006.

[30] F. Edelvik and G. Ledfelt, "A comparison of time-domain hybrid solvers for complex scattering problems," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 15, pp. 475–487, 2002.

[31] A. S. Stratton, *Electromagnetic Theory*. McGraw-Hill Book Company, 1941.

[32] P. Monk, *Finite Element Methods for Maxwell's Equations*. Oxford University Press, USA, June 2003.

[33] R. Courant and D. Hilbert, *Methods of Mathematical Physics, Vol. II*. Wiley-Interscience, 1 ed., 1962.

[34] G. A. Deschamps, "Electromagnetics and differential forms," *Proc. IEEE*, vol. 69, pp. 676–969, June 1981.

[35] D. Baldomir and P. Hammond, "Global geometry of electromagnetic systems," *IEE Proceedings, Part A: Science, Measurement and Technology*, vol. 140, pp. 142–150, Mar. 1993.

[36] K. F. Warnick and R. H. Selfridge, "Teaching electromagnetic field theory using differential forms," *IEEE Trans. Educ.*, vol. 40, pp. 53–68, Feb. 1997.

[37] H. Cartan, *Differential Forms*. London: Kershaw Publishing Company Ltd., 1971.

[38] H. Flanders, *Differential Forms with Applications to the Physical Sciences*. Academic Press Inc.,U.S., Dec. 1963.

[39] D. Lovelock and H. Rund, *Tensors, Differential Forms, and Variational Principles*. Dover Publications, Apr. 1989.

[40] I. V. Lindell, *Differential Forms in Electromagnetics*. Wiley-IEEE Press, Apr. 2004.

[41] P. Silvester and R. L. Ferrari, *Finite elements for electrical engineers*. Cambridge University Press, 1996.

[42] M. Salazar-Palma, T. K. Sarkar, L. Emilio Garcia-Costillo, and T. Roy, *Iterative and Self-Adaptive Finite-Elements in Electromagnetic Modeling*. Artech House Publishers, Sept. 1998.

[43] P. Monk and K. Parrott, "Phase-accuracy comparisons and improved far-field estimates for 3-d edge elements on tetrahedral meshes," *Journal of Computational Physics*, vol. 170, pp. 614–641, July 2001.

[44] A. F. Peterson and D. R. Wilton, "Curl-conforming mixed-order edge elements for discretizing the 2D and 3D vector helmholtz equation," in *Finite Element Software for Microwave Engineering*, ch. 5, pp. 101–125, John Wiley & Sons, 1996.

[45] N. Marais, "Higher order hierarchal curvilinear triangular vector elements for the finite element method in computational electromagnetics," Master's thesis, University of Stellenbosch, 2003. Available: `http://students.ee.sun.ac.za/~nmarais/publications/Neilen_Marais_Masters_thesis.pdf`.

[46] J. P. Swartz and D. B. Davidson, "Curvilinear vector finite elements using a set of hierarchical basis functions," *IEEE Trans. Antennas Propagat.*, vol. 55, pp. 440–446, 2007.

[47] E. Martini and S. Selleri, "Innovative class of curvilinear tetrahedral elements," *IEE Electron. Lett.*, vol. 37, pp. 557–558, Apr. 2001.

[48] R. Hiptmair, "Canonical construction of finite elements," *Mathematics of Computation*, vol. 68, pp. 1325–1346, May 1999.

[49] M. Costabel and M. Dauge, "Maxwell and Lamé eigenvalues on polyhedra," *Mathematical Methods in the Applied Sciences*, vol. 22, pp. 243–258, 1999.

[50] J. Fa Lee and D. Kow Sun, "p-type multiplicative schwarz (pMUS) method with vector finite elements for modeling three-dimensional waveguide discontinuities," *IEEE Trans. Microwave Theory Tech.*, vol. 52, pp. 864–870, 2004.

[51] A. Bossavit, "Whitney forms: a class of finite elements for three-dimensional computations in electromagnetism," *IEE Proceedings*, vol. 135, pp. 493–500, nov 1988.

[52] R. Hiptmair, "Higher order Whitney forms," *Progress In Electromagnetics Research*, vol. 32, pp. 271–299, 2001.

[53] D. B. Davidson, "An evaluation of mixed-order versus full-order vector finite elements," *IEEE Trans. Antennas Propagat.*, vol. 51, pp. 2430–2441, Sept. 2003.

[54] A. Bossavit and L. Kettunen, "Yee-like schemes on a tetrahedral mesh, with diagonal lumping," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 12, pp. 129–142, 1999.

[55] F. L. Teixeira and W. C. Chew, "Lattice electromagnetic theory from a topological viewpoint.," *Journal of Mathematical Physics*, vol. 40, p. 169, 1999.

[56] F. Teixeira, "Geometric aspects of the simplicial discretization of Maxwell's equations," *Progress In Electromagnetics Research*, vol. PIER 32, pp. 171–188, 2001.

[57] R. Hiptmair, "Discrete Hodge-operators: An algebraic perspective," *Progress In Electromagnetics Research*, vol. PIER 32, pp. 247–269, 2001.

[58] B. He and F. Teixeira, "On the degrees of freedom of lattice electrodynamics," *Physics Letters A*, vol. 336, pp. 1–7, Feb. 2005.

[59] B. He and F. Teixeira, "Geometric finite element discretization of Maxwell equations in primal and dual spaces," *Physics Letters A*, vol. 349, pp. 1–14, 2006.

[60] B. He and F. L. Teixeira, "Differential forms, Galerkin duality, and sparse inverse approximations in finite element solutions of Maxwell equations," *IEEE Trans. Antennas Propagat.*, vol. 55, pp. 1359–1368, 2007.

[61] A. Bossavit, "'Generalized finite differences' in computational electromagnetics," *Progress In Electromagnetics Research*, vol. PIER 32, pp. 45–64, 2001.

[62] T. Tarhasaari, L. Kettunen, and A. Bossavit, "Some realizations of a discrete Hodge operator: a reinterpretation of finite element techniques," *IEEE Trans. Microwave Theory Tech.*, vol. 35, pp. 1494–1497, 1999.

[63] R. Rieben, D. White, and G. Rodrigue, "High-order symplectic integration methods for finite element solutions to time dependent Maxwell equations," *IEEE Trans. Antennas Propagat.*, vol. 52, pp. 2190–2195, 2004.

[64] T. Belytschko, *An Overview of Semidiscretization and Time Integration Procedures*, vol. 1: Computational Methods for Transient Analysis of *Computational Methods in Mechanics*, pp. 67–155. Amsterdam: North-Holland, 1983. eds. T. Belytschko and T. J. R. Hughes.

[65] R. N. Rieben, *A Novel High Order Time Domain Vector Finite Element Method for the Simulation of Electromagnetic Devices.* Phd thesis, University of California Davis, July 2004.

[66] N. M. Newmark, "A method of computation for structural dynamics," *Journal of the Engineering Mechanics Division, American Society of Civil Engineers*, vol. 85, pp. 67–94, July 1959.

[67] R. Lehoucq, D. Sorensen, and C. Yang, *Arpack users' guide: Solution of large scale eigenvalue problems with implicitly restarted Arnoldi methods.* 1997.

[68] G. Rodrigue and D. White, "A vector finite element time-domain method for solving Maxwell's equations on unstructured hexahedral grids," *SIAM Journal of Scientific Computing*, vol. 23, pp. 683–706, 2001.

[69] M. S. Min and C. H. Teng, "The instability of the Yee scheme for the "Magic Time Step"," *Journal of Computational Physics*, vol. 166, pp. 418–424, 2001.

[70] E. Lezar and D. Davidson, "Implementation of arbitrarily high order hierarchical vector basis functions for the finite element analysis of a rectangular waveguide," in *AFRICON 2007*, pp. 1–7, 2007.

[71] J. S. Savage and A. F. Peterson, "Quadrature rules for numerical integration over triangles and tetrahedra," *IEEE Antennas Propagat. Mag.*, vol. 38, pp. 100–102, June 1996.

[72] P. Keast, "Moderate-degree tetrahedral quadrature formulas," *Computer Methods in Applied Mechanics and Engineering*, vol. 55, no. 3, pp. 339–348, 1986.

[73] N. Marais and D. Davidson, "Numerical evaluation of hierarchical vector finite elements on curvilinear domains in 2-D," *IEEE Trans. Antennas Propagat.*, vol. 54, pp. 734–738, 2006.

[74] P. Castillo, J. Koning, R. Rieben, M. Stowell, and D. A. White, *Discrete Differential Forms: A Novel Methodology for Robust Computational Electromagnetics*. 2003. Available: `http://dx.doi.org/10.2172/15003246`.

[75] P. Fernandes and M. Raffetto, "Characterization of spurious-free finite element methods in electromagnetics," *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 21, pp. 147–164, 2002.

[76] S. Zaglmayr, *High Order Finite Element Methods for Electromagnetic Field Computation*. PhD thesis, Johannes Kepler University, 2006. Available: `http://www.numerik.math.tugraz.at/~zaglmayr/pub/szthesis.pdf`.

[77] M. M. Botha, "Fully hierarchical divergence-conforming basis functions on tetrahedral cells, with applications," *International Journal for Numerical Methods in Engineering*, vol. 71, pp. 127–148, 2007.

[78] G. Cohen and P. Monk, "Gauss point mass lumping schemes for Maxwell's equations," *Numerical Methods for Partial Differential Equations*, vol. 14, pp. 63–88, 1998.

[79] Z. Ren and N. Ida, "High order differential form-based elements for the computation," *IEEE Trans. Magn.*, vol. 36, pp. 1472–1478, July 2000.

[80] D.-K. Sun, J.-F. Lee, and Z. C., "Construction of nearly orthogonal nedelec bases for rapid convergence with multilevel preconditioned solvers," *SIAM Journal on Scientific Computing*, vol. 23, no. 4, pp. 1053–1076, 2001.

[81] R. D. Graglia, D. R. Wilton, and A. F. Peterson, "Higher order interpolatory vector bases for computational electromagnetics," *IEEE Trans. Antennas Propagat.*, vol. 45, pp. 329–342, Mar. 1997.

[82] R. N. Rieben, D. A. White, and G. H. Rodrigue, "Improved conditioning of finite element matrices using new high-order interpolatory bases," *IEEE Trans. Antennas Propagat.*, vol. 52, pp. 2675–2683, Oct. 2004.

[83] M. M. Botha, *Efficient finite element electromagnetic analysis of antennas and microwave devices: the FE-BI-FMM formulation and a posteriori error estimation for p adaptive analysis.* PhD thesis, Stellenbosch University, Sept. 2002.

[84] M. M. Botha and D. B. Davidson, "A quasi-static condition for enhancing p-adaptive, mixed-order element, FE analysis," *Electromagnetics*, vol. 24, pp. 13–24, January–March 2004.

[85] W. Rachowicz and A. Zdunek, "An hp-adaptive finite element method for scattering problems in computational electromagnetics," *International Journal for Numerical Methods in Engineering*, vol. 62, pp. 1226–1249, 2005.

[86] Y. Zhu and A. C. Cangellaris, "Hierarchical multilevel potential preconditioner for fast finite-element analysis of microwave devices," *IEEE Trans. Microwave Theory Tech.*, vol. 50, pp. 1984–1989, Aug. 2002.

[87] J. P. Webb, "Matching a given field using hierarchal vector basis functions," *Electromagnetics*, vol. 24, pp. 113–122, January–March 2004.

[88] L. E. R. Petersson and J.-M. Jin, "An efficient procedure for the projection of a given field onto hierarchical vector basis functions of arbitrary order," *Electromagnetics*, vol. 25, pp. 81–91, 2005.

[89] A. Bossavit and J. C. Verite, "Mixed fem-biem method to solve 3-D eddy-current problems.," *IEEE Trans. Magn.*, vol. MAG 18, pp. 431–435, 1981.

[90] A. Bossavit and J. C. Verite, "'TRIFOU' code: Solving the 3-D eddy-currents problem by using H as state variable.," *IEEE Trans. Magn.*, vol. MAG-19, pp. 2465–247, 1983.

[91] G. Mur and A. de Hoop, "A finite-element method for computing three-dimensional electromagnetic fields in inhomogeneous media," *IEEE Trans. Magn.*, vol. 21, pp. 2188–2191, Nov. 1985.

[92] A. Bossavit and I. Mayergoyz, "Edge elements for scattering problems," *IEEE Trans. Antennas Propagat.*, vol. 25, pp. 2816–2821, jul 1989.

[93] J. P. Webb and B. Forghani, "Hierarchal scalar and vector tetrahedra," *IEEE Trans. Magn.*, vol. 29, pp. 1495–1498, Mar. 1993.

[94] T. V. Yioultsis and T. D. Tsiboukis, "Development and implementation of second and third order vector finite elements in various 3-D electromagnetic field problems," *IEEE Trans. Magn.*, vol. 33, pp. 1812–1815, Mar. 1997.

[95] L. S. Andersen and J. L. Volakis, "Development and application of a novel class of hierarchical tangential vector finite elements for electromagnetics," *IEEE Trans. Antennas Propagat.*, vol. 47, pp. 112–120, Jan. 1999.

[96] M. Hano, H. Komatsu, and K. Taniguchi, "Systematic construction of three-dimensional ultra high-order nedelec's elements," *IEEE Trans. Magn.*, vol. 36, pp. 1623–1626, jul 2000.

[97] M. Ainsworth and J. Coyle, "Conditioning of hierarchic p-version ndlec elements on meshes of curvilinear quadrilaterals and hexahedra," *SIAM Journal on Numerical Analysis*, vol. 41, no. 2, pp. 731–750, 2003.

[98] D. A. White, "Orthogonal vector basis functions for time domain finite element solution of the vector wave equation," *IEEE Trans. Magn.*, vol. 35, pp. 1458–1461, 1999.

[99] A. Elmkies and P. Joly, "Élments finis d'arête et condensation de masse pour les équations de Maxwell: le cas de dimension 3," *C. R. Acad. Sci. Paris Sér. I Math.*, vol. 325, pp. 1217–1222, Dec. 1997.

[100] C. W. Crowley, P. P. Silvester, and H. Hurwitz, "Covariant projection elements for 3D vector field problems," *IEEE Trans. Magn.*, vol. 24, pp. 397–400, 1988.

[101] G. von Winckel, "MATLAB Central file exchange - Legende-Gauss-Lobatto nodes and weights." Available `http://www.mathworks.com/matlabcentral/fileexchange/4775`.

[102] A. Fisher, R. N. Rieben, G. H. Rodrigue, and D. A. White, "A generalized mass lumping technique for vector finite-element solutions of the time-dependent Maxwell equations," *IEEE Trans. Antennas Propagat.*, vol. 53, pp. 2900–2910, 2005.

[103] L. Brutman, "Lebesgue functions for polynomial interpolation–a survey," *Annals Numerical Math.*, vol. 4, 1997.

[104] M. M. Ilić and B. M. Notaroš, "Higher order hierarchical curved hexahedral vector finite elements for electromagnetic modeling," *IEEE Trans. Microwave Theory Tech.*, vol. 51, pp. 1026–1033, 2003.

[105] M. M. Ilić and B. M. Notaroš, "Higher order large-domain hierarchical FEM technique for electromagnetic modeling using Legendre basis functions on generalized hexahedra," *Electromagnetics*, vol. 26, pp. 517–529, 2006.

[106] R. Rieben and D. White, "Arbitrary order hierarchical bases for computational electromagnetics," 2002. Available `http://nsdl.org/resource/2200/20061005075138563T`.

[107] R. Rieben, D. White, and G. Rodrigue, "Arbitrary order hierarchical vector bases for hexahedrons [FEM applications]," in *Antennas and Propagation Society International Symposium, 2003. IEEE*, vol. 2, pp. 181–184 vol.2, 2003.

[108] D. Beazley *et al.*, "Simplified Wrapper and Interface Generator (SWIG)." Available: `http://www.swig.org/index.html`.

[109] P. Peterson, "F2PY: Fortran to Python interface generator." Available: `http://cens.ioc.ee/projects/f2py2e/`.

[110] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*. Philadelphia, PA, or `http://www.netlib.org/lapack/lug/lapack_lug.html`: Society for Industrial and Applied Mathematics, 3rd ed., 1999.

[111] P. N. Swarztrauber, "FFTPACK." Available: `http://www.netlib.org/fftpack/`.

[112] D. Ascher, P. F. Dubois, K. Hinsen, J. Hugunin, T. Oliphant, *et al.*, *Numerical Python*. `http://numeric.scipy.org/numpydoc/numdoc.htm`.

[113] F. Pérez, "iPython: An Enhanced Interactive Python shell." Available: `http://ipython.scipy.org/`.

[114] J. Hunter *et al.*, "Matplotlib." Available: `http://matplotlib.sourceforge.net/`.

[115] N. Marais, "Driving and extending legacy codes using python - [em programmer's notebook]," *Antennas and Propagation Magazine, IEEE*, vol. 49, pp. 140–148, 2007.

[116] G. Booch, *Object-Oriented Analysis and Design with Applications*. Addison-Wesley Professional, 2 ed., 1993.

[117] "Test-driven development." Available: `http://en.wikipedia.org/wiki/Test_driven_development`.

[118] "Test-driven development." Available: `http://en.wikipedia.org/wiki/Revision_control`.

[119] "Unit test." Available: `http://en.wikipedia.org/wiki/Unit_test`.

[120] "Nose: a discovery-based unittest extension." Available: `http://somethingaboutorange.com/mrl/projects/nose/`.

[121] T. Lord, "GNU Arch." Available: `http://www.gnuarch.org/`.

[122] "Bazaar-NG, next-generation distributed revision control." Available: `http://bazaar-vcs.org/`.

[123] D. M. Pozar, *Microwave Engineering*. John Wiley & Sons, 2nd ed., 1998.

[124] S. L. Dvorak, "Exact, closed-form expressions for transient fields in homogeneously filled waveguides," *IEEE Trans. Microwave Theory Tech.*, vol. 42, pp. 2164–2170, 1994.

[125] J.-P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *Journal of Computational Physics*, vol. 114, pp. 185–200, Oct. 1994.

[126] T. Rylander and J.-M. Jin, "Perfectly matched layer in three dimensions for the time-domain finite element method applied to radiation problems," *IEEE Trans. Antennas Propagat.*, vol. 53, pp. 1489–1499, 2005.

[127] D. Jiao, J. Jin, E. Michielssen, and D. J. Riley, "Time-domain finite-element simulation of three-dimensional scattering and radiation problems using perfectly matched layers," *IEEE Trans. Antennas Propagat.*, vol. 51, pp. 296–305, Feb. 2003.

[128] S. Wang, R. Lee, and F. L. Teixeira, "Anisotropic-medium PML for vector FETD with modified basis functions," *IEEE Trans. Antennas Propagat.*, vol. 54, pp. 20–26, 2006.

[129] B. Donderici and F. L. Teixeira, "Conformal perfectly matched layer for the mixed finite element time-domain method," *IEEE Trans. Antennas Propagat.*, vol. 56, pp. 1017–1026, 2008.

[130] S. Wang and F. L. Teixeira, "Some remarks on the stability of time-domain electromagnetic simulations," *IEEE Transactions on Antennas and Propagation*, vol. 52, pp. 895–898, 2004.

[131] R. A. Chilton and R. Lee, "The discrete origin of FETD-Newmark late time instability, and a correction scheme," *Journal of Computational Physics*, vol. 224, pp. 1293–1306, June 2007.

[132] N. Marais and D. B. Davidson, "Conforming arbitrary order hexahedral/tetrahedral hybrid discretisation," *IET Electron. Lett.*, vol. 44, pp. 1384–385, Nov. 2008.

[133] Jean-Louis Coulomb, Francois-Xavier Zgainski, and Yves Marechal, "Pyramidal element to link hexahedral, prismatic and tetrahedral edge finite elements," *IEEE Trans. Magn.*, vol. 33, pp. 1362–1365, 1997.

[134] R. D. Graglia and I. Gheorm, "Higher order interpolatory vector bases on pyramidal elements," *IEEE Trans. Antennas Propagat.*, vol. 47, pp. 775–782, May 1999.

[135] V. Gradinaru and R. Hiptmair, "Whitney elements on pyramids," *Electronic Transactions on Numerical Analysis*, vol. 8, pp. 154–168, 1999.

[136] Z. Zeng and C.-C. Lu, "Discretization of hybrid VSIE using mixed mesh elements with zeroth-order Galerkin basis functions," *IEEE Trans. Antennas Propagat.*, vol. 54, pp. 1863–1870, 2006.

[137] H. A. Haus and J. R. Melcher, *Electromagnetic Fields and Energy.* Prentice Hall, Sept. 1989.

[138] R. Coccioli, A. Morini, G. Pelosi, and T. Rozzi, "Design of tolerance-corrected filters employing half-cylinder posts," *IEEE Trans. Microwave Theory Tech.*, vol. 46, pp. 116–118, 1998.

[139] T. A. Davis, "Algorithm 832: UMFPACK v4.3—an unsymmetric-pattern multifrontal method," *ACM Trans. Math. Softw.*, vol. 30, pp. 196–199, 2004.

[140] "FEKO website." Avaliable `http://www.feko.info/`. Developed by EMSS SA (Pty) Ltd.

[141] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, "Efficient management of parallelism in object oriented numerical software libraries," in *Modern Software Tools in Scientific Computing* (E. Arge, A. M. Bruaset, and H. P. Langtangen, eds.), pp. 163–202, Birkhäuser Press, 1997.

[142] S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, "PETSc users manual," Tech. Rep. ANL-95/11 - Revision 2.3.3, Argonne National Laboratory, 2007.

[143] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, "PETSc Web page," 2008. `http://www.mcs.anl.gov/petsc`.

[144] R. Falgout, J. Jones, and U. Yang, *Numerical Solution of Partial Differential Equations on Parallel Computers*, vol. 51, ch. The Design and Implementation of hypre, a Library of Parallel High Performance Preconditioners, pp. 267–294. Springer-Verlag, 2006.

[145] V. E. Henson and U. M. Yang, "BoomerAMG: A parallel algebraic multigrid solver and preconditioner," *Applied Numerical Mathematics*, vol. 41, pp. 155–177, Apr. 2002.

[146] S. J. Kim, J. Y. Cho, and W. D. Kim, "From the trapezoidal rule to higher-order accurate and unconditionally stable time-integration method for structural dynamics," *Computer Methods in Applied Mechanics and Engineering*, vol. 149, pp. 73–88, 1997.