Improvements of Finite Element Strategies in Electromagnetic Analysis

A thesis
submitted in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
by

Durgarao Kamireddy
(Reg. no.: 176103010)

Supervisor
Dr. Arup Nandy

Department of Mechanical Engineering
Indian Institute of Technology Guwahati,
Assam 781039, India
October 3, 2024

Dedicated to .....

My Parents and my family members, Gurus and Teachers

Declaration

I hereby declare that the information presented in the thesis titled **"Improvements of Finite Element Strategies in Electromagnetic Analysis"** is entirely my account of the research performed under the guidance of **Dr. Arup Nandy**. I affirm that none of the information within this thesis has been previously submitted for the purpose of obtaining any academic degree, diploma, associateship, fellowship, or any equivalent recognition from any other university or institution. Furthermore, I have complied with all copyright and plagiarism regulations, and all text and figures included in this work are my original contributions. In line with the general practice of reporting scientific observations, appropriate acknowledgements or citations have been included wherever the work described is based on the discoveries of fellow researchers.

Date:

Durgarao Kamireddy
Department of Mechanical Engineering,
Indian Institute of Technology Guwahati, India.
durga176103010@iitg.ac.in
Google Scholar || ResearchGate

Certificate

This is to certify that the work contained in thesis titled **"Improvements of Finite Element Strategies in Electromagnetic Analysis"**, submitted by **Durgarao Kamireddy** to the Indian Institute of Technology Guwahati, for the award of the Doctor of Philosophy degree in Mechanical Engineering, has been conducted under my direct supervision. The findings and results presented in this thesis have not been, either in whole or in part, presented to or considered for the conferral of any other academic degree or certification from any other university or institution.

Date:

Dr. Arup Nandy
Department of Mechanical Engineering,
Indian Institute of Technology Guwahati,
Guwahati 781039, Assam, India.
arupn@iitg.ac.in
https://www.iitg.ac.in/arupn/

Acknowledgement

I am very glad to have this opportunity to express my sincere gratitude to everyone who assisted me in different ways, to make possible my journey towards a doctoral degree.

At first, I am grateful to my thesis advisor Prof. Arup Nandy for giving the opportunity and allowing me to pursue my Ph.D. under his esteemed guidance. I am very much thankful for his help and guidance whenever I stuck in my thesis work by proposing lots of ideas. Furthermore, I am thankful to my doctoral committee, the chair person Prof. Atanu Banerjee, doctoral members Prof. Karuna Kalita, and Prof.Sisir Kumar Nayak for their valuable suggestions in my thesis work.

I would like to express my sincere gratitude to Indian Institute of Technology Guwahati and Department of Mechanical Engineering for providing the necessary resources and facilities that made this thesis possible. I am also thankful to all the faculty members and office staff of Mechanical Engineering department for their direct and indirect support during my Ph.D. journey. Furthermore, I would like to acknowledge the financial support provided by MHRD, Government of India. Also, I gratefully acknowledge the support from SERB, DST under the project IMP/2019/000276 and VSSC, ISRO through MoU No.:ISRO:2020:MOU:No:480., which has facilitated the execution of this research.

Throughout my Ph.D. work, I conducted all my simulations in Computational Mechanics and Optimization Lab (CMO Lab). I am deeply grateful to Prof. Prasenjit Khanikar (Lab in charge), Prof. Sachin Singh Gautam, Prof. Deepak Sharma, and Prof. Arup Nandy for providing all the computational resources and other facilities for my research work. I have cherished many beautiful memories with other students from this lab who are doing their research work along with me. At this point of time, I want to acknowledge the contributions of two M.Tech students, Mr. Saurabh Madhukar Chavan and Mr. Karanam Sreekanth, for their support and assistance in certain aspects of my research work. I would also like to express my sincere appreciation to Dr. Dhiraj Bombarde, a fellow research scholar pursuing a Ph.D. under our Supervisor. His collaboration, support, and insightful discussions have been invaluable throughout this journey. Together, we navigated through challenges, exchanged ideas, and contributed to each other's growth. Similarly, I want to acknowledge other research scholars from the Mechanics of Composites lab, particularly Dr. Chukka Achyuta Rao, Dr. S. Sai

Kumar, Dr. Pran Jyoti Saikia, Mr. Pabbu Krishna Murthy, Dr. Shasanka Kakoty, for their direct and indirect support in my research journey. Their moral support and discussions have been instrumental in shaping my academic endeavors.

I am profoundly indebted to my parents, my wife and he family members, my siblings and their family for their unwavering patience, moral support, and continuous encouragement during the challenging days in my doctoral journey. Their steadfast belief in me has been a constant source of strength and motivation. I am immensely grateful for their love, guidance, and unwavering presence throughout this transformative journey. Finally, the spiritual teachings of Secrets of Nature (Pranic Healing) and Hamsa Veda Yoga (Arhatic yoga) by Guruji Sri Mount Kailash, along with the blessings from the Shankaracharyas of Kanchi Mutt and Sringeri Mutt, have been the guiding light throughout this academic journey. They have instilled in me the strength, wisdom, and persistence to overcome challenges and reach this significant milestone.

Durgarao Kamireddy
Department of Mechanical Engineering,
Indian Institute of Technology Guwahati, India.
durga176103010@iitg.ac.in
Google Scholar || ResearchGate

Date:

Abstract

Standard nodal finite elements in electromagnetic analysis have well-known limitation of occurrence of spurious solution. In order to circumvent the problem, a penalty function method or a regularization method is used with potential formulation. These methods solve the problem partially by pushing the spurious mode to the higher end of the spectrum. But it fails to capture singular eigen values in case of the problem domains with sharp edges and corners. To circumvent this limitation, edge elements have been developed for electromagnetic analysis where degree of freedoms are along the edges. In this work, we have developed a technique to convert nodal data structure to edge data structure for electromagnetic analysis. Thereafter, we have demonstrated the relative performances of different nodal and edge elements through a series of examples. All possible complexities like curved boundaries, non-convex domains, sharp corners, non-homogeneous domains have been addressed in those examples. The robustness of edge elements in predicting the singular eigen values for the domains with sharp edges and corners is evident in the analysis. A better coarse mesh accuracy has been observed for all the edge elements as compared to the respective nodal elements.

In time domain to solve electromagnetic radiation and scattering problems, several numerical methods are available in literature. Among these numerical methods Finite Difference Time Domain (FDTD) method and Time Domain FEM (TDFEM) method are more popular. Here we have proposed a quantity in continuum, which is conserved under certain boundary conditions. We have also proposed a time marching scheme within edge element framework, which satisfy this conservation characteristics in algorithmic sense irrespective of the choice of time step. We have used same conversion algorithm as mentioned earlier.

While doing electromagnetic analysis using FEM (Finite element method), if we can implement the underlying symmetric nature of the problem, there will be significant reduction in the computational cost. Symmetric nature of the problem can be identified from the given physical loading and boundary conditions of the problem. But for electromagnetic analysis in potential formulation, it is not very straight forward to implement the symmetric boundary condition. Here, a novel implementation of symmetric boundary condition in potential formulation within nodal framework, has been demon-

strated. The implementation has been carried out in both electromagnetic harmonic and transient analysis for a wide range of radiation and scattering problems.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms/Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
ABC	Absorbing Boundary Condition
ADI-FDTD	Alternating Direction Implicitly Finite Domain Time Domain
BI	Boundary Integral
BIEM	Boundary Integral Element Method
CN-FDTD	Crank-Nicholson Finite Domain Time Domain
Con	Conventional
FD	Full Domain
FDOF	Free Degrees of Freedom
FDTD	Finite Difference Time Domain
FE	Finite Element
FEM	Finite Element Method
HDWS	Half Domain With Symmetry
HFSS	High Frequency Structure Simulator
LOD-FDTD	Locally-One Dimensional Finite Domain Time Domain
OWF	Outward Wave Favouring
PDE	Partial Differential Equation
PML	Perfectly Matched Layer
SS-FDTD	Split Step Finite Domain Time Domain
TDFEM	Time Domain Finite Element Method

Element abbreviation

B27	Conventional 27-node brick element
EQ12	12-edge quadrilateral element
EQ4	4-edge quadrilateral element
ET3	3-edge triangular element
ET8	8-edge triangular element
HexE12	12-edge hexahedral element

HexN8	Conventional 8-node hexahedral element
Q4	Conventional 4-node quadrilateral element
Q9	Conventional 9-node quadrilateral element
T6	Conventional 6-node triangular element
TetE6	6-edge tetrahedral element
TetN10	Conventional 10-node tetrahedral element
W18	Conventional 18-node wedge element

Variables defining computational limits of the programme

mx_exedge	Maximum additional edges after four edges, shared by one node
mx_nm_eleedge	Maximum number of edges in one element
mx_nm_elend	Maximum number of nodes in one element
mx_nmedge	Maximum number of edges
mx_nmele	Maximum number of elements
mx_nmnnd	Maximum number of nodes which are connecting more than four edges
mx_nmnode	Maximum number of global nodes

Scalar and Array variables used in conversion algorithm

edgearr	Array of dimension:(*mx_nm_eleedge*)
edgecon	Array of dimension:(*mx_nmele*, *mx_nm_eleedge*)
edgej	Number of existing connecting edges at the end node *ndi*
edgenode	Array of dimension:(*mx_nmedge*, 2)
eletyp	Type of the element
*nd*1	Starting node of the edge
*nd*2	End node of the edge
ndexn	Array of dimension:(*mx_nmnnd*)
nedge	Number of edges in each element
nele	Number of elements
nndexn	Number of nodes which are associated with more than 4 edges
nodarr	Array of dimension:(*mx_nm_elend*)
nodecon	Array of dimension:(*mx_nmele*, *mx_nm_elend*)
nodeedge	Array of dimension:(*mx_nmnode*, 8)
nodeedgeexn	Array of dimension:(*mx_nmnnd*, 2 × *mx_exedge*)
nodeedgenum	Array of dimension:(*mx_nmnode*)

Symbols

$\boldsymbol{A}$	Magnetic vector potential
$\boldsymbol{A}_\delta$	Variation of $\boldsymbol{A}$

$\bar{E}$	One mathematical quantity derived in transient analysis
$\bar{t}$	Non-dimensional parameter in thickness variation analysis
$\boldsymbol{B}$	Magnetic induction or magnetic flux
$\boldsymbol{D}$	Electric Displacement
$\boldsymbol{E}$	Electric field
$\boldsymbol{E}_{\delta}$	Variation of Electric field
$\boldsymbol{E}_{inc}$	Incident electric field
$\boldsymbol{E}_{scat}$	Scattered electric field
$\boldsymbol{f}$	Load vector
$\boldsymbol{H}$	Magnetic field
$\boldsymbol{J}$	Jacobian matrix of transformation
$\boldsymbol{j}$	Current density
$\boldsymbol{K}$	Stiffness matrix
$\boldsymbol{M}$	Mass matrix
$\boldsymbol{n}$	Unit normal vector
$\boldsymbol{\Gamma}$	Inverse of $\boldsymbol{J}$
$\boldsymbol{w}$	$\frac{\partial(\boldsymbol{\nabla}\psi)}{\partial t}$
$\boldsymbol{t}$	Tangent vector to the surface
$\boldsymbol{v}$	$\frac{\partial \boldsymbol{A}}{\partial t}$
$\dot{\boldsymbol{E}}$	$\frac{\partial \boldsymbol{E}}{\partial t}$
$\dot{\boldsymbol{H}}$	$\frac{\partial \boldsymbol{H}}{\partial t}$
ϵ	Electric permittivity
ϵ_0	Electric permittivity of vacuum
ϵ_r	Relative electric permittivity
Γ	Boundary
Γ_{∞}	Truncated boundary of the domain for exterior domain problems
Γ_e	Boundary where $\boldsymbol{E} \times \boldsymbol{n}$ is prescribed
Γ_h	Boundary where $\boldsymbol{H} \times \boldsymbol{n}$ is prescribed
$\hat{\boldsymbol{A}}$	Vector of magnetic vector potential
$\hat{\boldsymbol{E}}$	Vector of tangential electric field
$\hat{\psi}$	Vector of scalar potential electric field
λ	Lagrange multiplier
μ	Magnetic permeability
μ_0	Magnetic permeability of vacuum
μ_r	Relative magnetic permeability
Ω	Domain
ω	Harmonic frequency
ψ	Scalar potential of Electric field
ψ_{δ}	Variation of ψ
ρ	Charge density

$c = 1/\sqrt{\epsilon\mu}$	Speed of light
$c_0 = 1/\sqrt{\epsilon_0\mu_0}$	Speed of light in vacuum
E_i	Tangential electric field of i^{th} edge
$j_n()$	Spherical bessel function of first kind
k	Wave number in space
k_0	Wave number in vacuum
$P_n^{(m)}$	Associated Legendre polynomial
R_∞	Radius of the spherical truncation boundary for exterior domain problems
t	Time/Thickness of thin patch
t_Δ	Time-step in transient strategy

SI Units of different field variables

Symbol	SI units	SI Base units
$\boldsymbol{A}$	volts/meter	$\mathrm{kg{\cdot}m{\cdot}s^{-3}{\cdot}A^{-1}}$
$\boldsymbol{B}$	webers/meter	$\mathrm{kg{\cdot}m^{1}{\cdot}s^{-2}{\cdot}A^{-1}}$
c	meter/sec	$\mathrm{m{\cdot}s^{-1}}$
$\boldsymbol{D}$	coulomb/meter^2	$\mathrm{A{\cdot}s{\cdot}m^{-2}}$
$\boldsymbol{E}$	volts/meter	$\mathrm{kg{\cdot}m{\cdot}s^{-3}{\cdot}A^{-1}}$
ϵ	farads/meter	$\mathrm{kg^{-1}{\cdot}m^{-3}{\cdot}s^{4}{\cdot}A^{2}}$
$\boldsymbol{H}$	amperes/meter	$\mathrm{A{\cdot}m^{-1}}$
$\boldsymbol{j}$	amperes/meter^2	$\mathrm{A{\cdot}m^{-2}}$
k	radian/meter	$\mathrm{rad{\cdot}m^{-1}}$
μ	henry/meter	$\mathrm{kg{\cdot}m^{1}{\cdot}s^{-2}{\cdot}A^{-2}}$
ρ	coulomb	$\mathrm{A{\cdot}s}$
ψ	volt	$\mathrm{kg{\cdot}m^{2}{\cdot}s^{-3}{\cdot}A^{-1}}$
t	sec	s
ω	radian/sec	$\mathrm{rad{\cdot}s^{-1}}$

Chapter 1

Introduction

This chapter presents a brief overview of various numerical methods in computational electromagnetics, with a focus on finite element strategies in electromagnetic analysis. Following this, the section identifies the unexplored research gaps within the existing literature and outlines the objectives of the current work. Finally, it provides an overview of the thesis structure and arrangement.

Several numerical methods have been developed to solve the partial differential equations (PDE) in studying the transmission of electromagnetic waves, performing either harmonic or transient analysis in various engineering and scientific applications. Among the mesh based methods, Finite Element Method (FEM), and Finite Difference Method have gained a wide popularity due to their application to complex structures [1]. However, there are other powerful alternative methods also exist, such as mesh-free methods [2–5] and Isogeometric Analysis (IGA) [6–10]. The Spectral element method [11–14], an advanced numerical technique that combines aspects of FEM and spectral methods, has been applied to electromagnetic radiation and scattering problems. Recently, the Domain Decomposition Method (DDM) [15–17] has gained attention in recent years within the CEM community. Some of the applications of electromagnetic waves includes wireless satellite communication and optics, radar, remote sensing, bioelectromagnetics, electromagnetic forming etc. In the transmission of electromagnetic waves, waveguide structures are used to transmit the waves with minimum losses of energy.

The Finite Element Method (FEM) has been widely used for radiation and scattering problems in interior and exterior domains which has large applications in antenna radiations, waveguide transmissions etc. [1, 18]. In order to apply the FEM technique the domain can be discretized with either edge element or nodal element. Like other wave propagation problems, in electromagnetic wave analysis, the mesh size of finite elements for converged results is dependent on the operating frequency. At higher frequencies, a refined mesh is required for accurate results. However, the occurrence of spurious solutions in nodal FEM cannot be eliminated with mesh refinement strategy. In order to circumvent the problem, the penalty function method and regularization

method [19–21] have been used extensively in nodal FEM framework. These methods solve the problem partially by pushing the spurious mode to the higher end of the spectrum. In order to capture singular eigen value in case of sharp edges and corners, in the regularization method we have to use a penalty parameter varying from 0 at the sharp edge to 1 at a large distance from the sharp edge. Also, in order to take care of inherent tangential continuity and normal discontinuity across material interface, potential formulation is used in nodal finite element framework [22–26]. In [27], a two field variation formulation in electromagnetics was presented, which can predict the eigen frequencies very accurately with correct multiplicities. There is no ad-hoc term in the mixed formulation as in the penalty function or the regularization method. This method worked very well for all two dimensional geometries like non-convex domains with sharp corners, in-homogeneous domains, curved domains etc. In three-dimensions, mixed FEM worked for plane structures (structures without any curvature) quite well; there, it worked flawlessly with sharp edges and in-homogeneous domains. But in the case of curved three dimensional geometries, this mixed formulation failed. But, in literature to overcome the aforementioned limitations, edge elements were introduced.

1.1 Edge elements in electromagnetic analysis

Edge elements were introduced by Whitney, which are also called curl-conforming elements. J C Nedelec presented the conceptual theory of edge elements [1]. He presented a non conforming tetrahedron and cube finite elements construction [28] conforming the H curl and H div spaces. Whitney spaces can act as bases for edge elements in FEM for field type of problems [29] and eddy current problems. In [30–36] edge elements are used to solve the eigenvalue problems with different shapes. In [37], Alain Bossavit et al. solved the 3-D eddy current problems using the combination of FEM and Boundary Integral Element Method (BIEM) methods. Edge elements can be easily applied to an exterior domain problems where the coupling of other methods like Absorbing Boundary Condition (ABC) [38], Boundary Integral (BI) [39], Perfectly Matched Layer (PML) [40] are required. Zoltan J. Cendes et al. [39] presented the implementation of the tangential vector finite element method to analyse the dielectric waveguides. In the literature various methods like method of moment, spectral-domain methods, finite difference and finite element are adopted for analysis of dielectric waveguide problems. Recently edge finite elements have been applied to model widely different domain problems, like marine controlled source electromangetics [41–44], geophysical electromagnetic problems [45, 46].

J P Webb [47] broadly discussed the useful properties of edge elements. Vector finite element or edge element [28, 30, 31, 48, 49] was proposed to circumvent the spurious solution problem of nodal FEM. Electromagnetic radiation and scattering problems require special elements where normal discontinuity and tangential continuity exists

across material interfaces met by edge elements. In [50], M. L. Barton et al. showed that continuity of tangential components of the vector field is sufficient in vector based FEM to compute the magnetic fields. Another advantage of edge elements is that electric or magnetic fields can be directly computed without any differentiation on potentials. Also, no penalty or regularization term is required in edge element framework. While modelling sharp, perfectly conducting objects, the electric field has to be infinite inside the domain and its direction changes rapidly at the sharp edges and corners. In order to do eigen analysis of such geometries in nodal framework singular trial functions are required, whereas due to tangential continuity of edge elements, no such function is required.

T. V. Yioultsis et al. [51, 52] presented the systematic approach to construct the higher order tetrahedral edge elements and used them to solve the waveguide problems with material discontinuity. General expressions for the shape functions were presented and unknown coefficients were found by following the decoupling procedure. In [53], the formation of higher order Whitney p-elements was shown. In [54], the construction of higher order two dimensional and three dimensional H1 curl elements are presented to solve the electromagnetic scattering problems. In [55], Roberto D. Graglia et al. presented the general approach of interpolatory vector basis functions of various two dimensional and three dimensional elements. Edge elements can be constructed by using hierarchial vector basis functions also. Jon P. Webb [56] proposed hierarchical vector basis functions for higher order triangle and tetrahedral finite elements. In [57], Seung-Cheol Lee et al. implemented higher order hierarchical vector finite elements in the field of microwave engineering to the wave guiding structures. In hierarchial type implementation, there can be p refinement in some part of the domain and in some part we can have h refinement. But, in interpolatory type we can only have one type of refinement in the entire domain. However, recently, a new type of second-order hexhedral and tetrahedral edge elements were proposed by using the idea of edge unification. Here, the element has one edge variable on each side as a result of unifying both edge on the side. By adopting these edge elements, considerable reduction in computational cost without any loss of accuracy was achieved [58, 59].

When defining edge degrees of freedom (DoFs), special attention is needed to ensure proper continuity across cells and to avoid sign conflicts. Various solutions have been proposed to ensure consistency [60–63]. For instance, in [61, 62], the authors introduced a sign flip condition for first-order 3D finite elements, ensuring that all local DoFs shared by two or more tetrahedral elements correspond to the same global DoF. In [60, 61], another approach based on the constructing all possible shape functions combined with a permutation that depends on the local edge/face orientations. There are several software libraries, such as NGSOLVE [64], FreeFEM++ [65], and HERMES [66] that provide higher-order elements defined on curved boundaries, suited for complex computations. However, these codes are often challenging to understand and modify unless the user is

already familiar with them. Modifying or adding features that are not readily available can be difficult. Commercial mesh generation packages (ANSYS, ABAQUS, Maxwell 2D & 3D, and COMSOL) for both nodal and edge elements are also widely available, but understanding their implementation can be tedious and making adjustments to meet specific needs can be challenging.

To address these challenges, a general conversion algorithm that provides full implementation details would be highly beneficial for students and researchers who wish to familiarize themselves with edge elements and prefer to implement their own solutions. In our research, one of our objectives is to develop a systematic conversion algorithm from nodal to edge elements. Further, we want to validate the generated node-to-edge elements by performing electromagnetic eigen analysis.

1.2 Electromagnetic harmonic and transient analysis

While solving electromagnetic radiation and scattering problems using FEM, the infinite exterior domain is required to be truncated at some finite radius as our computational domain should be finite. To mimic the infinite nature, some appropriate absorbing boundary condition (ABC) is imposed on the truncation surface [22]. This ABC is determined based on Sommerfeld radiation condition [67]. In [67–70], various second-and higher-order ABCs were implemented. Over past 20 years, analytical ABCs, Perfectly Matched Layers (PML) and different versions of Huygens ABCs [71, 72] have been introduced and applied to finite element methods. ABCs are not limited to FEM, as many authors have also used them in electromagnetic radiation and scattering analysis within the finite-difference time-domain (FDTD) method. In [73–76], different types of ABCs were implemented in FDTD to simulate electromagnetic wave propagation. A brief historical review of ABCs, along with their advantages and limitations, can be found in [77, 78]. Overall, the FEM-ABC technique is effective for handling open-region problems, but ABCs are not exact cannot perfectly absorb all outgoing fields. This introduces small errors in the solution. To minimize these errors, the artificial boundary must be placed far from the source to reduce reflections, though this increases computational cost.

In [67–70, 79, 80], authors modelled the interior computational domain with edge FEM and the truncation surface with Boundary Integral (BI) method. Even though this technique is more accurate than traditional ABCs, a key disadvantage is the generation of densely populated matrix by BI elements, which results in higher memory requirement and computational time.

In certain electromagnetic problems, the presence of symmetry allows for a reduction in the computational domain, leading to a significant decrease in computational cost. If a plane of symmetry exists based on the loading conditions and boundary setup, only half (or even a portion) of the domain needs to be considered in the analysis. To achieve

this, appropriate boundary conditions must be applied on the plane of symmetry. This approach significantly reduces computational cost.

The implementation of symmetric boundary condition in computational electromagnetics has been extensively studied. Numerous studies have demonstrated the use of various numerical methods that leverage symmetry for solving cavity resonant problems, as well as radiation and scattering problems. Key developments based on a comprehensive literature survey are outlined here. The connection between Maxwell's equations and symmetry lies in the fact that these equations, along with boundary conditions and material properties, remain invariant under specific symmetry transformations, such as reflections, rotations, or periodic transformations [81]. For instance, in [82], resonance frequencies for cylindrically symmetric cavities were computed using the finite integration technique (FIT), which shares similarities with finite difference methods. In [81], author discusses boundary conditions for various types of symmetries, such as reflection symmetry about a plane, periodic structures, symmetric periodic structures and axisymmetric structures. Eigenvalue analysis was then performed on these different 2D domains to explore these symmetries in the context of FEM. David et al. proposed an efficient formulation based on the method-of-moments to analyze planar structures with azimuthal symmetry [83]. Additionally, in [84], 3D inhomogeneous cavities were simulated using symmetries in the variational meshless method (VMM). Recently, an efficient method based on FEM was introduced in [85] to characterize waveguides with arbitrary cross sections that exhibit discrete rotational symmetries (C_N) symmetries. In [86], a method for imposing symmetry planes using conformal and non-conformal uniform meshes was introduced for waveguide ports in both FDTD and FDFD simulations.

On the symmetry plane, symmetric conditions are typically enforced by applying the perfect electric conductor (PEC) boundary condition or perfect magnet conductor (PMC) boundary condition. However, implementing these conditions in potential formulation is not very straightforward [87]. Inspired by this idea, we aim to propose some novel technique to implement this symmetric boundary condition in potential formulation in the nodal framework, for both electromagnetic harmonic and transient analysis.

Being inspired from the Wilcox asymptotic expansion for electric field in exterior domain [88], A. Nandy et al. proposed an amplitude formulation [22] in nodal FEM, using potentials, to carry out harmonic electromagnetic analysis, for solving electromagnetic radiation and scattering problems in exterior domain. In the current work, we want to extend our novel implementation of symmetric boundary condition in the Amplitude formulation [22].

A wide range of numerical methods are available to solve Maxwell's electromagnetic wave equation in time domain. They can be classified in two broad categories such as Finite Difference Time Domain (FDTD) method and Time Domain FEM (TDFEM) method. FDTD, introduced by Yee [89], gained popularity due to its ease in implementation of the numerical integration. TDFEM [90–96] soon replaced FDTD as it could

be easily generalized to complex geometries. In TDFEM, two approaches were followed to solve for electromagnetic field variables. In first approach [94–99], second order electromagnetic wave equation was solved for either electric or magnetic field. Then from post processing, other field was obtained. In the other approach [90, 92, 93], two coupled first order Maxwell's equations were solved for both field variables simultaneously. In [94, 100–102], Finite Element Boundary Integral (FEBI) method was developed for exterior domain problems. In transient electromagnetic analysis, different time stepping strategies such as leapfrog strategy [103, 104], central difference method [94, 101, 105], were used for temporal discretization.

Ensuring the stability of the algorithm is crucial for obtaining reliable solutions in transient electromagnetic analysis. Factors such as the finite element mesh size and the time step used during the time-marching process, influence the stability of the algorithm [106]. There exists a close relationship between the unconditional stability of an algorithm and the conservation or dissipation of energy in the physical model [106]. According to Poynting's theorem [107], in a lossless medium without sources, the electromagnetic energy of a wave remains constant over time. Therefore, any numerical scheme must adhere to this principle to achieve unconditional stability. In [99], authors developed an unconditional stability time marching scheme for higher order TD-MFEM for different boundary conditions. In [24], the authors introduced a time-stepping strategy for TDFEM, which demonstrated unconditional stability from an energy perspective. Consequently, we aim to extend the time-stepping strategy outlined in [24] into the edge element framework to solve electromagnetic radiation and scattering problems.

1.3 Maxwell's Equations in Electromagnetics

The governing differential equations for electromagnetic analysis are Maxwell's equations, given in the strong form as [108]

$$\frac{\partial \boldsymbol{B}}{\partial t} + \boldsymbol{\nabla} \times \boldsymbol{E} = \boldsymbol{0}, \tag{1.1a}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{B} = 0, \tag{1.1b}$$

$$\frac{\partial \boldsymbol{D}}{\partial t} - \boldsymbol{\nabla} \times \boldsymbol{H} = -\boldsymbol{j}, \tag{1.1c}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{D} = \rho, \tag{1.1d}$$

where the electric and magnetic fields are given by $\boldsymbol{E}$ and $\boldsymbol{H}$, the electric displacement (electric flux) is $\boldsymbol{D}$, the magnetic induction (magnetic flux) is $\boldsymbol{B}$, the charge density is ρ, and the current density is $\boldsymbol{j}$. The following constitutive relations complement the above governing equations

$$\boldsymbol{D} = \epsilon \boldsymbol{E}, \tag{1.2a}$$

$$\boldsymbol{B} = \mu \boldsymbol{H}, \tag{1.2b}$$

where μ and ϵ are the magnetic permeability and electric permittivity, respectively. Considering that ϵ and μ are independent of time and substituting the constitutive relations into Eqs. (1.1a), (5.16) and (1.1d), and also after eliminating $\boldsymbol{H}$ we get

$$\epsilon \frac{\partial^2 \boldsymbol{E}}{\partial t^2} + \frac{\partial \boldsymbol{j}}{\partial t} + \boldsymbol{\nabla} \times \left(\frac{1}{\mu} \boldsymbol{\nabla} \times \boldsymbol{E} \right) = \boldsymbol{0}. \tag{1.3}$$

We get the compatibility condition from Eqs. (5.16), (1.2a) and (1.1d) as

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{j} = \boldsymbol{0}.$$

We have the boundary condition as on conducting boundary, $\boldsymbol{E} \times \boldsymbol{n} = \boldsymbol{0}$. Also, across the material interface, both $\boldsymbol{E} \times \boldsymbol{n}$ and $\boldsymbol{H} \times \boldsymbol{n}$ are continuous as there is no impressed surface currents.

Introducing the relative permeability and relative permittivity $\mu_r := \mu/\mu_0$ and $\epsilon_r := \epsilon/\epsilon_0$, where μ_0 and ϵ_0 are the permeability and permittivity for the vacuum, Eq. (1.3) can be written in the frequency domain as

$$\boldsymbol{\nabla} \times \left(\frac{1}{\mu_r} \boldsymbol{\nabla} \times \boldsymbol{E} \right) - k_0^2 \epsilon_r \boldsymbol{E} = -i\omega\mu_0 \boldsymbol{j}, \tag{1.4}$$

where $k_0 = \omega/c$ is the wave number, $c = 1/\sqrt{\epsilon_0 \mu_0}$, and $i = \sqrt{-1}$. Considering $\boldsymbol{j}$ to be zero, the Eq. 1.4 becomes

$$\boldsymbol{\nabla} \times \left(\frac{1}{\mu_r} \boldsymbol{\nabla} \times \boldsymbol{E} \right) = k_0^2 \epsilon_r \boldsymbol{E}. \tag{1.5}$$

which governs the eigenvalue problem.

1.4 Objectives of the present work

Based on the scope of research gaps identified from the literature survey, the objectives of the present work are as follows:

1. Developing a node to edge conversion algorithm. This algorithm uses coordinates and connectivity data of nodal finite element and it does necessary conversion to solve the problem using edge finite element.
2. Performing eigen value analysis of different electromagnetic problems in edge element framework using this conversion algorithm. Thereafter, to carry out detailed comparison of the performance between nodal and the edge elements.
3. Applying this conversion algorithm for transient electromagnetic analysis. In transient electromagnetic, one mathematical quantity ($\bar{E}$) is proposed in continuum framework which conserves under certain conditions. A suitable time

stepping strategy is proposed. It has been proved both algorithmically as well as with numerical examples that the proposed time stepping strategy mimic conservation of $\bar{E}$ under required condition.

4. In nodal finite element frame work with potential formulation, to implement symmetric boundary condition in electromagnetic harmonic and transient analysis.

1.5 Organization of the thesis

We divide the thesis into six chapters. Throughout the thesis we follow the SI unit convention, unless otherwise stated. SI units of different variables are given in page xxvi. For solving the final matrix equations, we use the WSMP (Watson Sparse Matrix Package) package [109]. The outline of each chapter is discussed below.

In **Chapter 1**, an overview of the literature survey is presented based on application of numerical methods in computational electromagnetics, focusing on finite element strategies in electromagnetic analysis. Additionally, the chapter identifies unexplored research gaps within the existing literature and outlines the objectives of the current work, with the explicit aim of introducing a novel contribution to the scientific discourse.

In **Chapter 2**, a standalone conversion algorithm is presented, capable of generating edge connectivity information from the supplied nodal connectivity data. Additionally, this chapter provides the details of the various two-dimensional and three-dimensional nodal-to-edge elements utilized in our work. For electromagnetic wave equation under no loading conditions, variational and FEM formulations in both nodal and edge framework are outlined. Finally, the conversion flow of the algorithm is presented, accompanied by suitable examples in both two and three-dimensions.

In **Chapter 3**, focus lies on assessment of numerical performance of the transformed edge elements introduced in chapter 2. The chapter provides the details of the several numerical examples of eigenvalue analysis in both 2D and 3D. These examples are part of our comparative performance analysis between the generated edge elements and corresponding conventional nodal finite elements. Finally, important observations made in our comparative analysis are summarized.

In **Chapter 4**, the initial focus is on providing an overview of state-of-the-art in transient electromagnetics. Following that, the chapter presents the variational statement and edge based FEM implementation to solve electromagnetic radiation and scattering problems in time domain. Following this, a specific quantity ($\bar{E}$) is derived, and its conservation is demonstrated both in the continuum framework and algorithmically, for the adopted time-stepping strategy. Finally, the chapter presents details of the benchmark examples considered for simulating electric fields, along with the demonstration of the conservation of $\bar{E}$.

In **Chapter 5**, a novel implementation of symmetric boundary condition in potential formulation within nodal framework is demonstrated through the thin-patch method.

Then, this method is applied to selected numerical examples that exhibits a plane of symmetry, to demonstrate the computational efficacy of the proposed method as compared to the existing formulation. Finally, an extensive study is conducted for proper selection of the thickness in the thin patch method.

In **Chapter 6**, a comprehensive summary of the work and its novel contributions to the existing literature is presented. The conclusions drawn from the results presented in various chapters are then outlined, offering insights into the implications of the findings. The chapter concludes by presenting the potential avenues for future research, identifying areas where further exploration and advancements can be made in the field.

Chapter 2

Conversion algorithm for creating Edge data structure from Nodal data structure*

For electromagnetic analysis in edge element framework, element data is required in the form of edge data structure. To achieve this, a standalone conversion algorithm is needed where the edge information is generated as output from the supplied nodal data as input. Here, every edge is generated by joining the two nodes of the element. In this chapter, we are presenting such conversion algorithm to different edge elements in both two-dimensions and three-dimensions. Such kind of program is necessary because most of the available commercial mesh generator create and generate nodal element data structure.

The remaining chapter is organized as follows: In section 2.1, variational and finite element formulations of electromagnetic eigen analysis is presented. In section 2.2, details of the different two-dimensional edge elements used in our work are given. Then, different variables used in the conversion algorithm are presented in section 2.3. Thereafter, in section 2.4 we have explained the flow of the algorithm in minute details with associated flowcharts and simple conversion examples for different two-dimensional elements like four edge quadrilateral, three edge triangle, twelve edge quadrilateral and eight edge triangle. Then, it is followed by the description of the different three-dimensional edge finite elements in section 2.5 and the conversion flow in generating the edge connectivity data for these elements in section 2.6.

*D. Kamireddy and A. Nandy, "A novel conversion technique from nodal to edge finite element data structure for electromagnetic analysis", *Computer Assisted Methods in Engineering and Science* 28(4), 291−319 (2022). https://cames.ippt.gov.pl/index.php/cames/article/view/384

2.1 Variational and FEM formulations

2.1.1 Variational statement for Potential formulation

For homogeneous domains, we get wrong multiplicities of eigenvalues and for inhomogeneous domains, we get spurious values using regularized formulations in the nodal framework [19]. It is partially overcome in the regularized potential formulation [110], which is fairly robust in all convex domains, homogeneous or inhomogeneous. Only, it fails to find the singular eigenvalues for a non-convex domain, owing to the penalty term.

Replacing $\boldsymbol{E}$ by $\boldsymbol{A}+\boldsymbol{\nabla}\psi$ in Eq. (1.5), we have

$$\boldsymbol{\nabla}\times\left(\frac{1}{\mu_r}\boldsymbol{\nabla}\times\boldsymbol{A}\right)=k_0^2\epsilon_r\boldsymbol{A}+k_0^2\epsilon_r\boldsymbol{\nabla}\psi, \tag{2.1}$$

where ψ and $\boldsymbol{A}$ are scalar and vector potentials. We get the required variational statement after adding the penalty term [110] as

$$\int_\Omega \frac{1}{\mu_r}(\boldsymbol{\nabla}\times\boldsymbol{A}_\delta)\cdot(\boldsymbol{\nabla}\times\boldsymbol{A})\,d\Omega+\int_\Omega\frac{1}{\epsilon_r\mu_r}(\boldsymbol{\nabla}\cdot\boldsymbol{A}_\delta)[\boldsymbol{\nabla}\cdot(\epsilon_r\boldsymbol{A})]\,d\Omega= k_0^2\int_\Omega\epsilon_r\boldsymbol{A}_\delta\cdot\boldsymbol{A}\,d\Omega+k_0^2\int_\Omega\epsilon_r\boldsymbol{A}_\delta\cdot\boldsymbol{\nabla}\psi. \tag{2.2}$$

Multiplying Eq. (1.1d) by the variation ψ_δ, replacing $\boldsymbol{E}$ by $\boldsymbol{A}+\boldsymbol{\nabla}\psi$, and considering no charge for eigen analysis we have

$$k_0^2\int_\Omega\boldsymbol{\nabla}\psi_\delta\cdot(\epsilon_r\boldsymbol{A})\,d\Omega+k_0^2\int_\Omega\boldsymbol{\nabla}\psi_\delta\cdot(\epsilon_r\boldsymbol{\nabla}\psi)\,d\Omega=\boldsymbol{0}. \tag{2.3}$$

On parts of the boundary where $\boldsymbol{E}\times\boldsymbol{n}=\boldsymbol{0}$, we specify ψ and $\boldsymbol{A}\times\boldsymbol{n}$ to be zero.

2.1.2 FEM Formulation in nodal framework

The vector potential $\boldsymbol{A}$ and its variation $\boldsymbol{A}_\delta$ are discretized as

$$\boldsymbol{A}=\boldsymbol{N}\hat{\boldsymbol{A}},$$
$$\boldsymbol{A}_\delta=\boldsymbol{N}\hat{\boldsymbol{A}}_\delta,$$

leading to

$$\boldsymbol{\nabla}\times\boldsymbol{A}=\boldsymbol{B}\hat{\boldsymbol{A}},\qquad \boldsymbol{\nabla}\cdot\boldsymbol{A}=\boldsymbol{B}_p\hat{\boldsymbol{A}},$$
$$\boldsymbol{\nabla}\times\boldsymbol{A}_\delta=\boldsymbol{B}\hat{\boldsymbol{A}}_\delta,\qquad \boldsymbol{\nabla}\cdot\boldsymbol{A}_\delta=\boldsymbol{B}_p\hat{\boldsymbol{A}}_\delta.$$

Similarly, for the scalar potential ψ, we have

$$\psi = \boldsymbol{N}_\psi \hat{\psi}, \qquad \boldsymbol{\nabla}\psi = \boldsymbol{B}_\psi \hat{\psi},$$
$$\psi_\delta = \boldsymbol{N}_\psi \hat{\psi}_\delta, \qquad \boldsymbol{\nabla}\psi_\delta = \boldsymbol{B}_\psi \hat{\psi}_\delta,$$

where

$$\boldsymbol{N} = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & \dots \\ 0 & N_1 & 0 & 0 & N_2 & 0 & \dots \\ 0 & 0 & N_1 & 0 & 0 & N_2 & \dots \end{bmatrix}, \quad \boldsymbol{N}_\psi = \begin{bmatrix} N_1 & N_2 & N_3 & \dots \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} 0 & -\frac{\partial N_1}{\partial z} & \frac{\partial N_1}{\partial y} & 0 & -\frac{\partial N_2}{\partial z} & -\frac{\partial N_2}{\partial y} & \dots \\ \frac{\partial N_1}{\partial z} & 0 & -\frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial z} & 0 & -\frac{\partial N_2}{\partial x} & \dots \\ -\frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & 0 & -\frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & 0 & \dots \end{bmatrix},$$

$$\boldsymbol{B}_p = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial z} & \dots \end{bmatrix},$$

$$\boldsymbol{B}_\psi = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} & \dots \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} & \dots \\ \frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial z} & \frac{\partial N_3}{\partial z} & \dots \end{bmatrix}.$$

The discretized forms of Eqs. (2.2) and (2.3) are given by

$$\begin{bmatrix} \boldsymbol{K}_{AA} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{A}} \\ \hat{\psi} \end{bmatrix} = k_0^2 \begin{bmatrix} \boldsymbol{M}_{AA} & \boldsymbol{M}_{A\psi} \\ \boldsymbol{M}_{\psi A} & \boldsymbol{M}_{\psi\psi} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{A}} \\ \hat{\psi} \end{bmatrix}, \tag{2.4}$$

where

$$\boldsymbol{K}_{AA} = \int_\Omega \frac{1}{\mu_r} \left[\boldsymbol{B}^T \boldsymbol{B} + \boldsymbol{B}_p^T \boldsymbol{B}_p \right] d\Omega, \tag{2.5a}$$

$$\boldsymbol{M}_{AA} = \int_\Omega \epsilon_r \boldsymbol{N}^T \boldsymbol{N}\, d\Omega, \tag{2.5b}$$

$$\boldsymbol{M}_{A\psi} = \int_\Omega \epsilon_r \boldsymbol{N}^T \boldsymbol{B}_\psi\, d\Omega, \tag{2.5c}$$

$$\boldsymbol{M}_{\psi A} = \int_\Omega \epsilon_r \boldsymbol{B}_\psi^T \boldsymbol{N}\, d\Omega, \tag{2.5d}$$

$$\boldsymbol{M}_{\psi\psi} = \int_\Omega \epsilon_r \boldsymbol{B}_\psi^T \boldsymbol{B}_\psi\, d\Omega. \tag{2.5e}$$

2.1.3 FEM formulation in edge element framework

The variational formulation of Eq. (1.5) can be derived as

$$\int_{\Omega} \frac{1}{\mu_r}(\boldsymbol{\nabla} \times \boldsymbol{E}_\delta) \cdot (\boldsymbol{\nabla} \times \boldsymbol{E})\, d\Omega = k_0^2 \int_{\Omega} \epsilon_r \boldsymbol{E}_\delta \cdot \boldsymbol{E}\, d\Omega. \tag{2.6}$$

where the boundary conditions $\boldsymbol{E} \times \boldsymbol{n}$ and $\boldsymbol{H} \times \boldsymbol{n}$ are specified on the surfaces Γ_e and Γ_h of the domain respectively. For the conducting boundary, we have $\boldsymbol{E} \times \boldsymbol{n} = \boldsymbol{0}$ and for eigen analysis, we have $\boldsymbol{H} = \boldsymbol{0}$.

We discretize the fields and their variations in Eq.(2.6) as

$$\boldsymbol{E} = \boldsymbol{V}\hat{\boldsymbol{E}}, \qquad \boldsymbol{E}_\delta = \boldsymbol{V}\hat{\boldsymbol{E}}_\delta,$$
$$\boldsymbol{\nabla} \times \boldsymbol{E} = \boldsymbol{B}\hat{\boldsymbol{E}}, \qquad \boldsymbol{\nabla} \times \boldsymbol{E}_\delta = \boldsymbol{B}\hat{\boldsymbol{E}}_\delta,$$

where $\hat{\boldsymbol{E}}$ is the values of $\boldsymbol{E}$ at different edges, $\hat{\boldsymbol{E}}_\delta$ denote the respective variation of $\hat{\boldsymbol{E}}$. Edge shape functions matrix, $\boldsymbol{V}$ and $\boldsymbol{B}$-matrix are given as

$$\boldsymbol{V} = \begin{bmatrix} v_{1x} & v_{2x} & \dots \\ v_{1y} & v_{2y} & \dots \\ v_{1z} & v_{2z} & \dots \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} \frac{\partial v_{1z}}{\partial y} - \frac{\partial v_{1y}}{\partial z} & \frac{\partial v_{2z}}{\partial y} - \frac{\partial v_{2y}}{\partial z} & \dots \\ \frac{\partial v_{1x}}{\partial z} - \frac{\partial v_{1z}}{\partial x} & \frac{\partial v_{2x}}{\partial z} - \frac{\partial v_{2z}}{\partial x} & \dots \\ \frac{\partial v_{1y}}{\partial x} - \frac{\partial v_{1x}}{\partial y} & \frac{\partial v_{2y}}{\partial x} - \frac{\partial v_{2x}}{\partial y} & \dots \end{bmatrix}.$$

where v_x, v_y and v_z are the x, y, z components of each edge shape function.

After substituting the above discretizations into Eq.(2.6) and using the arbitrariness of variations we get

$$\boldsymbol{K}\hat{\boldsymbol{E}} = k_0^2 \boldsymbol{M}\hat{\boldsymbol{E}}, \tag{2.7}$$

where,

$$\boldsymbol{K} = \int_{\Omega} \frac{1}{\mu_r} \boldsymbol{B}^T \boldsymbol{B}\, d\Omega, \tag{2.8a}$$

$$\boldsymbol{M} = \int_{\Omega} \epsilon_r \boldsymbol{V}^T \boldsymbol{V}\, d\Omega. \tag{2.8b}$$

2.1.4 Calculation of $\frac{\partial \xi}{\partial x}$, $\frac{\partial \xi}{\partial y}$, $\frac{\partial \xi}{\partial z}$, $\frac{\partial \eta}{\partial x}$, $\frac{\partial \eta}{\partial y}$, $\frac{\partial \eta}{\partial z}$, $\frac{\partial \zeta}{\partial x}$, $\frac{\partial \zeta}{\partial y}$ and $\frac{\partial \zeta}{\partial z}$

The shape functions of edge element contains the terms of inverse Jacobian ($\boldsymbol{\Gamma}$) and its components $\frac{\partial \xi}{\partial x}$, $\frac{\partial \xi}{\partial y}$, $\frac{\partial \xi}{\partial z}$, $\frac{\partial \eta}{\partial x}$, $\frac{\partial \eta}{\partial y}$, $\frac{\partial \eta}{\partial z}$, $\frac{\partial \zeta}{\partial x}$, $\frac{\partial \zeta}{\partial y}$, and $\frac{\partial \zeta}{\partial z}$ which can be obtained as discussed below. In two dimensions, we have the following relation as

$$\begin{Bmatrix} \frac{\partial f}{\partial \xi} \\ \frac{\partial f}{\partial \eta} \end{Bmatrix} = \boldsymbol{J} \begin{Bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{Bmatrix} \implies \begin{Bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{Bmatrix} = \boldsymbol{J}^{-1} \begin{Bmatrix} \frac{\partial f}{\partial \xi} \\ \frac{\partial f}{\partial \eta} \end{Bmatrix}$$

where $\boldsymbol{J}$ is Jacobian can be written as

$$\boldsymbol{J} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \text{ and assume } \boldsymbol{J}^{-1} = \boldsymbol{\Gamma} = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{21} & \Gamma_{22} \end{bmatrix};$$

$$\text{therefore, } \begin{Bmatrix} \frac{\partial \xi}{\partial x} \\ \frac{\partial \xi}{\partial y} \end{Bmatrix} = \begin{Bmatrix} \Gamma_{11} \\ \Gamma_{21} \end{Bmatrix} \text{ and } \begin{Bmatrix} \frac{\partial \eta}{\partial x} \\ \frac{\partial \eta}{\partial y} \end{Bmatrix} = \begin{Bmatrix} \Gamma_{12} \\ \Gamma_{22} \end{Bmatrix}.$$

Similarly the components of inverse Jacobian in three dimensions can be obtained as:

From the relation,

$$\begin{Bmatrix} \frac{\partial f}{\partial \xi} \\ \frac{\partial f}{\partial \eta} \\ \frac{\partial f}{\partial \zeta} \end{Bmatrix} = \boldsymbol{J} \begin{Bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{Bmatrix} \implies \begin{Bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{Bmatrix} = \boldsymbol{J}^{-1} \begin{Bmatrix} \frac{\partial f}{\partial \xi} \\ \frac{\partial f}{\partial \eta} \\ \frac{\partial f}{\partial \zeta} \end{Bmatrix}$$

where

$$\boldsymbol{J} = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{21} & J_{22} & J_{23} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \text{ and } \boldsymbol{\Gamma} = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} & \Gamma_{13} \\ \Gamma_{21} & \Gamma_{22} & \Gamma_{23} \\ \Gamma_{31} & \Gamma_{32} & \Gamma_{33} \end{bmatrix};$$

$$\text{therefore, } \begin{Bmatrix} \frac{\partial \xi}{\partial x} \\ \frac{\partial \xi}{\partial y} \\ \frac{\partial \xi}{\partial z} \end{Bmatrix} = \begin{Bmatrix} \Gamma_{11} \\ \Gamma_{21} \\ \Gamma_{31} \end{Bmatrix}, \begin{Bmatrix} \frac{\partial \eta}{\partial x} \\ \frac{\partial \eta}{\partial y} \\ \frac{\partial \eta}{\partial z} \end{Bmatrix} = \begin{Bmatrix} \Gamma_{12} \\ \Gamma_{22} \\ \Gamma_{32} \end{Bmatrix} \text{ and } \begin{Bmatrix} \frac{\partial \zeta}{\partial x} \\ \frac{\partial \zeta}{\partial y} \\ \frac{\partial \zeta}{\partial z} \end{Bmatrix} = \begin{Bmatrix} \Gamma_{13} \\ \Gamma_{23} \\ \Gamma_{33} \end{Bmatrix}.$$

2.2 Different edge elements in two dimensions

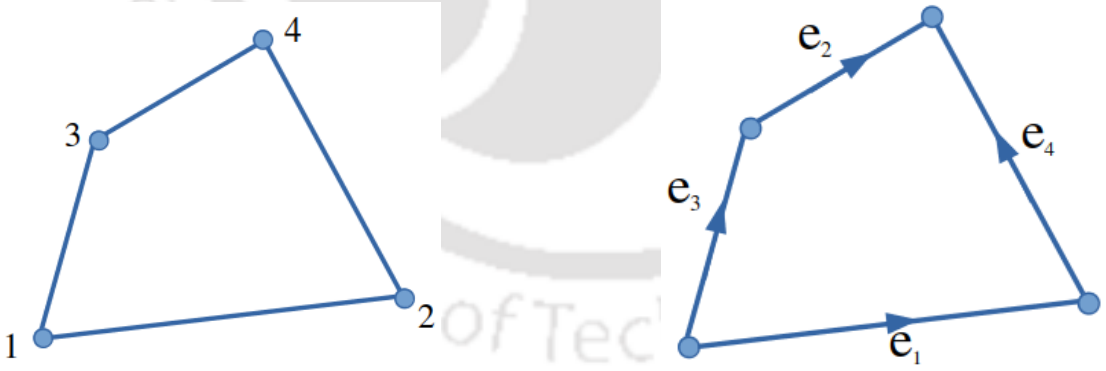
Different two-dimensional nodal elements used in this work are well known in the literature. We denote Four node quadrilateral elements by Q4, Nine node quadrilateral elements by Q9, and Six node triangular elements by T6. In this section, we are presenting different edge elements adopted in our work.

2.2.1 Four edge quadrilateral element

For four node quadrilateral element Fig. 2.1a shows the local nodal connectivity for which Fig. 2.1b shows the required local edge connectivity sequence. Edge e_1 is formed by connecting the local node set (1,2). Similarly, e_2, e_3 and e_4 are the other three edges generated by connecting the node sets (4,3), (1,4) and (2,3) respectively. We denote this element by EQ4. The edge shape functions of four edges are given as [1]

$$\begin{Bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \boldsymbol{v}_3 \\ \boldsymbol{v}_4 \end{Bmatrix} = \begin{Bmatrix} \frac{l_1}{4}(1-\eta)\boldsymbol{\nabla}\xi \\ \frac{l_2}{4}(1+\eta)\boldsymbol{\nabla}\xi \\ \frac{l_3}{4}(1-\xi)\boldsymbol{\nabla}\eta \\ \frac{l_4}{4}(1+\xi)\boldsymbol{\nabla}\eta \end{Bmatrix},$$

where l_1, l_2, l_3, and l_4 are lengths of edges e_1, e_2, e_3, and e_4 respectively.



(a) Element with four nodes. **(b)** Element with four edges.

Figure 2.1: Quadrilateral element

The components of $\boldsymbol{\nabla} \times \boldsymbol{E}$ can be calculated by using the relation,

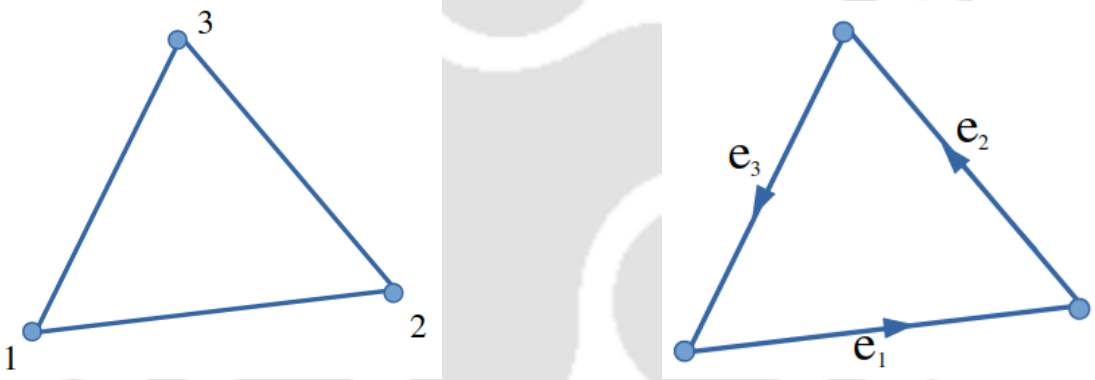
$$\boldsymbol{\nabla} \times \boldsymbol{E} = \begin{bmatrix} \frac{\partial v_{1y}}{\partial x} - \frac{\partial v_{1x}}{\partial y} & \frac{\partial v_{2y}}{\partial x} - \frac{\partial v_{2x}}{\partial y} & \frac{\partial v_{3y}}{\partial x} - \frac{\partial v_{3x}}{\partial y} & \frac{\partial v_{4y}}{\partial x} - \frac{\partial v_{4x}}{\partial y} \end{bmatrix} \begin{Bmatrix} E_1 \\ E_2 \\ E_3 \\ E_4 \end{Bmatrix} = \boldsymbol{B}\hat{\boldsymbol{E}},$$

where E_1, E_2, E_3 and E_4 are the tangential electric fields along the edges 1, 2, 3 and 4 edges respectively. The components of the $\boldsymbol{B}$-matrix is obtained as below.

$$\begin{bmatrix} \frac{\partial v_{1x}}{\partial x} & \frac{\partial v_{1x}}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial v_{1x}}{\partial \xi} & \frac{\partial v_{1x}}{\partial \eta} \end{bmatrix} \begin{bmatrix} \Gamma_{11} & \Gamma_{21} \\ \Gamma_{12} & \Gamma_{22} \end{bmatrix}.$$

We can derive partial derivative terms $\frac{\partial v_{1x}}{\partial \xi}$, $\frac{\partial v_{1y}}{\partial \eta}$, ... etc. with finite difference or by mathematica software [111].

2.2.2 Three edge triangular element



(a) Element with three nodes. **(b)** Element with three edges.

Figure 2.2: Triangular element

Lower order three edge triangular element is formed from three node triangle and Fig. 2.2a shows the nodal connectivity (local) of the triangular element. For this element the expected edge connectivity can be shown in Fig. 2.2b. With the help of node sets (1,2), (2,3) and (3,1) three edges e_1, e_2 and e_3 are formed respectively and we represent this element as ET3. For this triangular element three edge shape functions are [1] given as

$$\begin{Bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \boldsymbol{v}_3 \end{Bmatrix} = \begin{Bmatrix} l_1(\xi \boldsymbol{\nabla}\eta - \eta \boldsymbol{\nabla}\xi) \\ l_2(-\eta \boldsymbol{\nabla}\xi - (1-\xi)\boldsymbol{\nabla}\eta) \\ l_3((1-\eta)\boldsymbol{\nabla}\xi + \xi \boldsymbol{\nabla}\eta) \end{Bmatrix},$$

where l_1, l_2 and l_3 are the edge lengths of three edges.

From the relation,

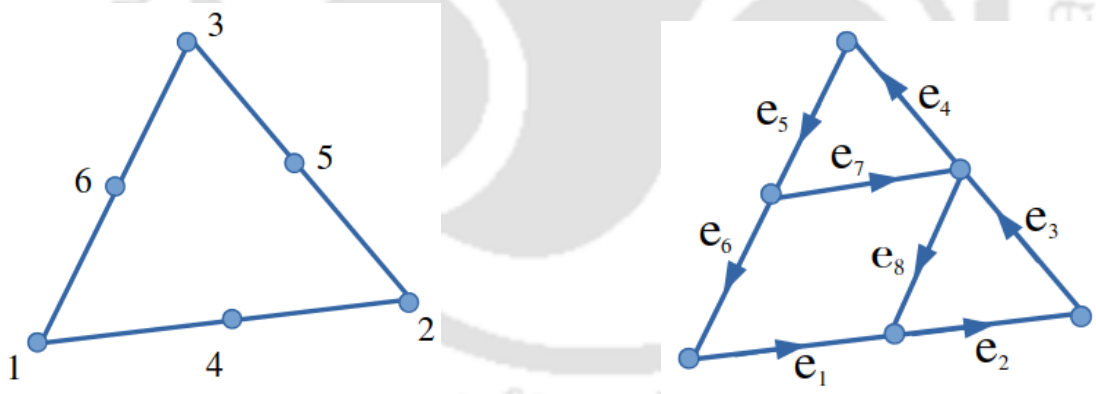
$$\boldsymbol{\nabla} \times \boldsymbol{E} = \begin{bmatrix} \frac{\partial v_{1y}}{\partial x} - \frac{\partial v_{1x}}{\partial y} & \frac{\partial v_{2y}}{\partial x} - \frac{\partial v_{2x}}{\partial y} & \frac{\partial v_{3y}}{\partial x} - \frac{\partial v_{3x}}{\partial y} \end{bmatrix} \begin{Bmatrix} E_1 \\ E_2 \\ E_3 \end{Bmatrix} = \boldsymbol{B}\hat{\boldsymbol{E}},$$

where E_1, E_2 and E_3 are tangential components of electric fields along the three edges e_1, e_2 and e_3 respectively. The components of the $\boldsymbol{B}$-matrix can be obtained by using the relation,

$$\begin{bmatrix} \frac{\partial v_{1x}}{\partial x} & \frac{\partial v_{1x}}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial v_{1x}}{\partial \xi} & \frac{\partial v_{1x}}{\partial \eta} \end{bmatrix} \begin{bmatrix} \Gamma_{11} & \Gamma_{21} \\ \Gamma_{12} & \Gamma_{22} \end{bmatrix}.$$

We can derive the $\frac{\partial v_{1x}}{\partial \xi}$, $\frac{\partial v_{1x}}{\partial \eta}$, ... etc., explicitly.

2.2.3 Eight edge triangular element



(a) Triangular element with six nodes. **(b)** Triangular element with eight edges.

Figure 2.3: Higher order triangular element

Higher order triangular edge element (eight edge triangle) is formed from six node triangular element. Fig. 2.3a is the local nodal connectivity of six node triangular element. Fig. 2.3b represents the desired local edge connectivity of eight edge triangular element. Here eight edges e_1, e_2, e_3, e_4, e_5, e_6, e_7 and e_8 are formed by using the node sets (1,4), (4,2), (2,5), (5,3), (3,6), (6,1), (6,5) and (5,4) respectively. l_1, l_2, ..., l_8 are the edge lengths of the element. Here, three edges can be formed on the face of the element. But one edge can be ignored due to independancy of three edges. This element

is denoted by ET8. The edge shape functions of the element are [112] expressed as

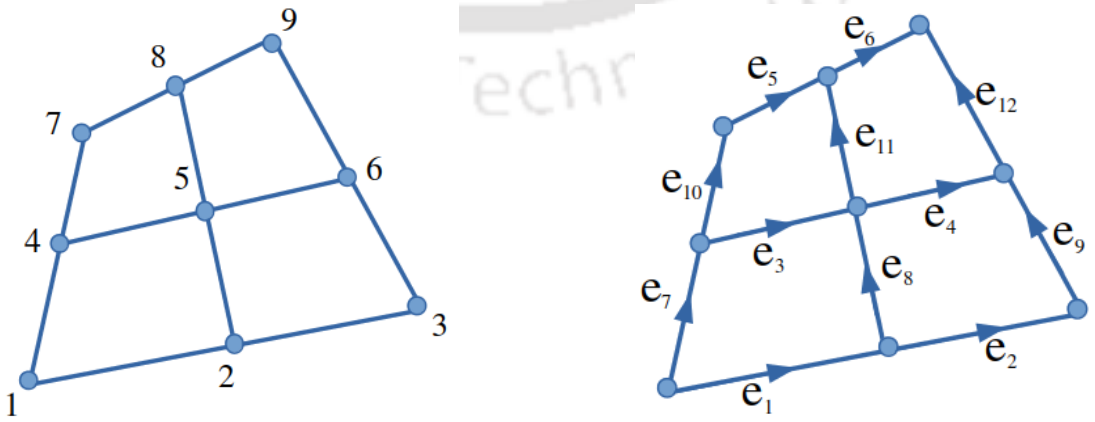
$$\begin{Bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \boldsymbol{v}_3 \\ \boldsymbol{v}_4 \\ \boldsymbol{v}_5 \\ \boldsymbol{v}_6 \\ \boldsymbol{v}_7 \\ \boldsymbol{v}_8 \end{Bmatrix} = \begin{Bmatrix} l_1(4\xi-1)(\xi\boldsymbol{\nabla}\eta - \eta\boldsymbol{\nabla}\xi) \\ l_2(4\eta-1)(\xi\boldsymbol{\nabla}\eta - \eta\boldsymbol{\nabla}\xi) \\ l_3(4\eta-1)(\eta\boldsymbol{\nabla}\alpha - \alpha\boldsymbol{\nabla}\eta) \\ l_4(4\alpha-1)(\eta\boldsymbol{\nabla}\alpha - \alpha\boldsymbol{\nabla}\eta) \\ l_5(4\alpha-1)(\alpha\boldsymbol{\nabla}\xi - \xi\boldsymbol{\nabla}\alpha) \\ l_6(4\xi-1)(\alpha\boldsymbol{\nabla}\xi - \xi\boldsymbol{\nabla}\alpha) \\ 4l_7\eta(\alpha\boldsymbol{\nabla}\xi - \xi\boldsymbol{\nabla}\alpha) \\ 4l_8\xi(\eta\boldsymbol{\nabla}\alpha - \alpha\boldsymbol{\nabla}\eta) \end{Bmatrix},$$

where $\alpha = 1 - \xi - \eta$ and $\boldsymbol{\nabla} \times \boldsymbol{E}$ can be calculated by using the relation,

$$\boldsymbol{\nabla} \times \boldsymbol{E} = \begin{bmatrix} \frac{\partial v_{1y}}{\partial x} - \frac{\partial v_{1x}}{\partial y} & \frac{\partial v_{2y}}{\partial x} - \frac{\partial v_{2x}}{\partial y} & \cdots & \frac{\partial v_{7y}}{\partial x} - \frac{\partial v_{7x}}{\partial y} & \frac{\partial v_{8y}}{\partial x} - \frac{\partial v_{8x}}{\partial y} \end{bmatrix} \begin{Bmatrix} E_1 \\ E_2 \\ \vdots \\ E_7 \\ E_8 \end{Bmatrix} = \boldsymbol{B}\hat{\boldsymbol{E}},$$

where E_1, E_2, ..., E_8 are tangential components of electric fields along the edges e_1, e_2, ..., e_8 respectively. Here, we can obtain the components of the $\boldsymbol{B}$-matrix explicitly.

2.2.4 Twelve edge quadrilateral element



(a) Quadrilateral element with nine nodes.

(b) Quadrilateral element with twelve edges.

Figure 2.4: Higher order quadrilateral element

Higher order quadrilateral edge element with twelve edges is formed from the nine node quadrilateral element. The edges e_1, e_2, e_3, ..., e_{12} are formed from the local node sets (1,2), (2,3), (4,5), (5,6), (7,8), (8,9), (1,4), (2,5), (3,6), (4,7), (5,8) and (6,9) respectively. Local nodal connectivity and edge connectivity are shown in Fig. 2.4a and Fig. 2.4b respectively. EQ12 is used to denote this element and twelve edge shape functions are [1]

$$\begin{Bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \boldsymbol{v}_3 \\ \boldsymbol{v}_4 \\ \boldsymbol{v}_5 \\ \boldsymbol{v}_6 \\ \boldsymbol{v}_7 \\ \boldsymbol{v}_8 \\ \boldsymbol{v}_9 \\ \boldsymbol{v}_{10} \\ \boldsymbol{v}_{11} \\ \boldsymbol{v}_{12} \end{Bmatrix} = \begin{Bmatrix} \frac{-l_1}{2}\eta(\eta-1)(\xi-0.5)\boldsymbol{\nabla}\xi \\ \frac{l_2}{2}\eta(\eta-1)(\xi+0.5)\boldsymbol{\nabla}\xi \\ l_3(\eta^2-1)(\xi-0.5)\boldsymbol{\nabla}\xi \\ -l_4(\eta^2-1)(\xi+0.5)\boldsymbol{\nabla}\xi \\ \frac{-l_5}{2}\eta(\eta+1)(\xi-0.5)\boldsymbol{\nabla}\xi \\ \frac{l_6}{2}\eta(\eta+1)(\xi+0.5)\boldsymbol{\nabla}\xi \\ \frac{-l_7}{2}\xi(\xi-1)(\eta-0.5)\boldsymbol{\nabla}\eta \\ l_8(\xi^2-1)(\eta-0.5)\boldsymbol{\nabla}\eta \\ \frac{-l_9}{2}\xi(\xi+1)(\eta-0.5)\boldsymbol{\nabla}\eta \\ \frac{l_{10}}{2}\xi(\xi-1)(\eta+0.5)\boldsymbol{\nabla}\eta \\ -l_{11}(\xi^2-1)(\eta+0.5)\boldsymbol{\nabla}\eta \\ \frac{l_{12}}{2}\xi(\xi+1)(\eta+0.5)\boldsymbol{\nabla}\eta \end{Bmatrix},$$

where l_1, l_2, ..., l_{11}, and l_{12} are lengths of edges 1, 2, ..., 11, and 12 respectively and $\boldsymbol{\nabla} \times \boldsymbol{E}$ of the element can be calculated by using the following relation.

$$\boldsymbol{\nabla} \times \boldsymbol{E} = \begin{bmatrix} \frac{\partial v_{1y}}{\partial x} - \frac{\partial v_{1x}}{\partial y} & \frac{\partial v_{2y}}{\partial x} - \frac{\partial v_{2x}}{\partial y} & \cdots & \frac{\partial v_{11y}}{\partial x} - \frac{\partial v_{11x}}{\partial y} & \frac{\partial v_{12y}}{\partial x} - \frac{\partial v_{12x}}{\partial y} \end{bmatrix} \begin{Bmatrix} E_1 \\ E_2 \\ \vdots \\ E_{11} \\ E_{12} \end{Bmatrix} = \boldsymbol{B}\hat{\boldsymbol{E}},$$

where E_1, E_2, ..., E_{12} are tangential components of electric fields along the edges e_1, e_2, ..., e_{12} respectively. We can obtain the components of the $\boldsymbol{B}$-matrix explicitly by using the finite difference method or mathematica software tool [111].

2.3 Different variables in the conversion algorithm

In this section we define and discuss the various variables used in the algorithm.

1. Variables defining computational limits of the programme:
   (i) *mx_nmnode* - maximum number of global nodes,
   (ii) *mx_nmele* - maximum number of elements,
   (iii) *mx_nm_elend* - maximum number of nodes in one element,
   (iv) *mx_nm_eleedge* - maximum number of edges in one element,
   (v) *mx_nmnnd* - maximum number of nodes which are connecting more than four edges,
   (vi) *mx_exedge* - maximum additional edges after four edges, shared by one node,
   (vii) *mx_nmedge* - maximum number of edges.

2. Other scalar variables:
   (i) *nedge* - number of edges in each element,
   (ii) *nele* - number of elements,
   (iii) *eletyp* - type of the element,
   (iv) *edgej* - number of existing connecting edges at the end node *ndi*,
   (v) *nd*1 - starting node of the edge,
   (vi) *nd*2 - end node of the edge,
   (vii) *nndexn* - number of nodes which are associated with more than 4 edges.

3. Array variables:
   (i) *nodecon* :
   dimension : (*mx_nmele*, *mx_nm_elend*)
   The i^{th} row contain global node numbers of the i^{th} element as per nodal connectivity.
   (ii) *nodarr* :
   dimension : (*mx_nm_elend*)
   This is a local variable inside the loop of element numbers. It extract the nodal connectivity data from *nodecon* for the running element.
   (iii) *edgecon* :
   dimension : (*mx_nmele*, *mx_nm_eleedge*)
   The i^{th} row contain global edge numbers of the i^{th} element as per edge connectivity.
   (iv) *edgearr* :
   dimension : (*mx_nm_eleedge*)
   This is a local variable inside the loop of element numbers. It accumulates the edge connectivity data for the running element in the programme flow.

(v) *edgenode* :
dimension : (mx_nmedge, 2)
(j, 1)$^{\text{th}}$ entry $\Rightarrow$ starting node of the j^{th} edge.
(j, 2)$^{\text{th}}$ entry $\Rightarrow$ end node of the j^{th} edge.

(vi) *nodeedgenum* :
dimension : (mx_nmnode)
The i^{th} row contains total number of edges connected with global node i.

(vii) *nodeedge* :
dimension : (mx_nmnode, 8)
i^{th} row contains of informations about first 4 connecting edges associated with the i^{th} global node. The odd columns (1, 3, 5, 7) contain the edge numbers and the even columns (2, 4, 6, 8) have the other node number of the edge in previous column. Odd columns of *nodeedge* also contain direction information of the edge in the following manner. The edge number is stored as positive number if the i^{th} global node is the starting node of the edge. On the other hand, odd column contain negative integer for the edge if the edge directs from the other node to the i^{th} global node.

(viii) *ndexn* :
dimension : (mx_nmnnd)
This array contains only those node numbers which has more than four connecting edges. The entries are according to the order of occurrence.

(ix) *nodeedgeexn* :
dimension : (mx_nmnnd, $2 \times mx\_exedge$)
(i, 1)$^{\text{th}}$ entry $\Rightarrow$ 5$^{\text{th}}$ edge no. for the node $ndexn(i)$. Direction convention is same as *nodeedge* array.
(i, 2)$^{\text{th}}$ entry $\Rightarrow$ Other end node of the 5$^{\text{th}}$ edge for node $ndexn(i)$.
All subsequent odd columns have the edge numbers for 6$^{\text{th}}$, 7$^{\text{th}}$, ... connecting edges. Each of the even column contain other node no. of the edge in the previous odd column.

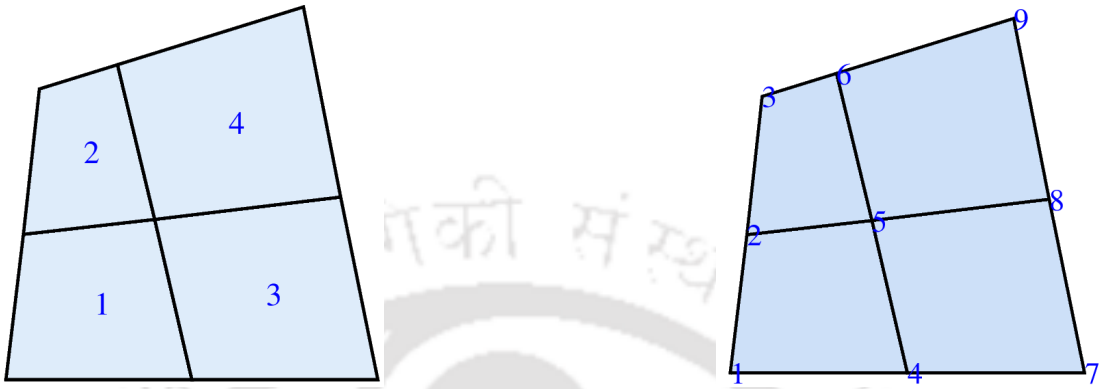
2.4 Conversion algorithm in two dimensions

In this section, we explain the flow of the conversion technique to generate edge connectivity information from nodal connectivity data for two-dimensional finite elements.

2.4.1 Conversion algorithm with four edge quadrilateral element: one representative example

To describe the conversion algorithm, we have considered a domain which is meshed with quadrilateral elements as shown in Fig. 2.5a. Fig. 2.5b shows the global nodal

connectivity of the meshed domain. Nodal connectivity list for all the elements for the mesh is given in Table 2.1.



(a) Domain with global elements. **(b)** Domain with global nodes.

Figure 2.5: Discretized domain

Table 2.1: Element nodal connectivity

Element number	**Nodal connectivity (Global node no.)**
1	1, 4, 5, 2
2	2, 5, 6, 3
3	4, 7, 8, 5
4	5, 8, 9, 6

In this algorithm, the outermost loop runs over the total number of discretized elements and the next inner loop runs over total number of local edges (*nlocedge*) of each element. One global counter e_l is used which is updated to the last assigned global edge number at the end of each element loop. Inside the loop of *nlocedge*, a subroutine ***edgend*** as shown in Fig. 2.6 returns global node numbers of two end nodes (*endnd*1 and *endnd*2) of the edge using local nodal connectivity and local edge connectivity information as shown in Fig. 2.1a and Fig. 2.1b. We follow the convention that each edge direct from *endnd*1 to *endnd*2.

For example, two local nodes 4 and 3 (see Fig. 2.1a) are connected to form the local edge e_2 of the element as shown in Fig. 2.1b. These two local node numbers are the position in the nodal connectivity array (*nodarr*) of the element, this array contain global node numbers as shown in Table 2.1. Thus, for example, for second element, global nodes 6 and 3 are two end nodes for edge e_2. Local edges and its corresponding connected nodes of the elements for the discretized domain is tabulated in Table 2.2.

After successful collection of output from the ***edgend*** subroutine i.e., information about two end nodes of local edge (i), last assigned global edge (e_l) and existing edge connectivity array (*edgearr*) are further supplied into ***edgedata*** structure as shown in

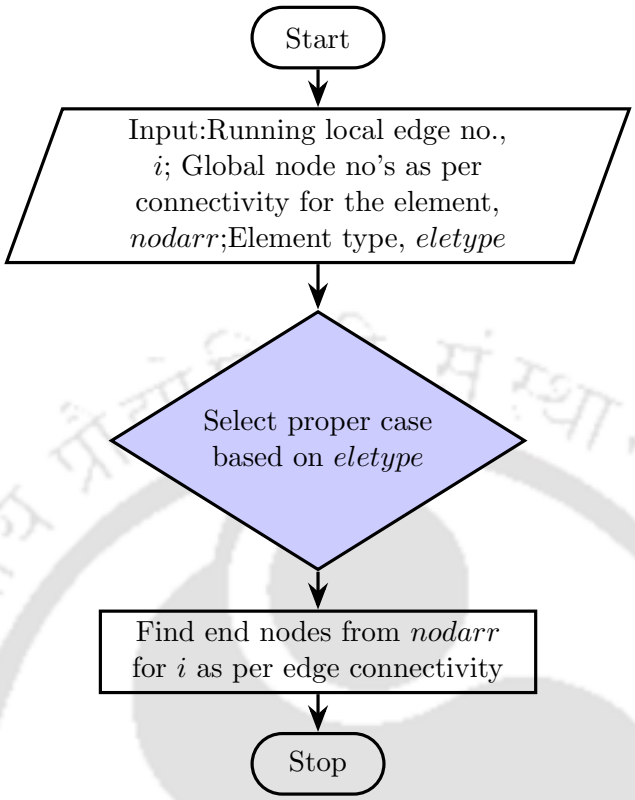


Figure 2.6: Flow chart of edgend structure

Table 2.2: Local edges and its end nodes of elements of discretized domain.

Element number	**Local edge number**	**End nodes (*endnd*1, *endnd*2)**
1	1	1, 4
	2	2, 5
	3	1, 2
	4	4, 5
2	1	2, 5
	2	3, 6
	3	2, 3
	4	5, 6
3	1	4, 7
	2	5, 8
	3	4, 5
	4	7, 8
4	1	5, 8
	2	6, 9
	3	5, 6
	4	8, 9

Fig. 2.7. In this data structure at the end of each iteration (for each local edge), different

variables like *nodeedgenum, nodeedge, nodeedgexn, edgenode, edgearr* are updated for every endnode ($nd1$ and $nd2$) of each edge which are discussed as below.

1. *nodeedgenum*: Global one dimensional static array of dimension maximum number of global nodes (*mx_nmnode*) in which i^{th} row store the number of edges shared by the i^{th} global node.
2. *nodeedge*: This is two dimensional array of dimension (*mx_nmnode* $\times$ 8) where 8 columns of i^{th} row store two information of 4 connecting edges shared by i^{th} node. 1^{st} column store global edge no. of 1^{st} connecting edge, second column store other end node of that connecting edge. 3^{rd} and 4^{th} column store (edge no., other end node no.) of the second connecting edge. 5^{th} to 8^{th} column store similar set of information for third and fourth connecting edge. If some node is shared by more than four connecting edges then additional information from 5^{th} edge are stored in *nodeedgeexn*.
   Also this array *nodeedge* have the direction information of the edge. If the i^{th} node is the start node of the edge i.e. if the edge is going from i^{th} node to the other node then edge no. is stored in the odd column as positive integer. If the edge is towards i^{th} node from the other node then edge no. is stored as negative integer. Total no. of negative edges are kept in account with one counter variable.
3. *nodeedgeexn*: Global two dimensional array of size(maximum number of nodes in the programme shared by more than 4 edges (*mx_nmnnd*), 2 $\times$ maximum additional edges after four edges sharing one node (*mx_exedge*)). Each row consists of the information about the node having more than four connecting edges. It consists of the information from the fifth edge onwards. Global number of the node whose information are stored in the i^{th} row of *nodeedgeexn* is stored in i^{th} row of *ndexn*.
   nndexn - No. of nodes which are associated with more than 4 edges.
   ndexn(i) - i^{th} node number connected with more than 4 edges (according to occurrence)
   nodeedgeexn(i, 1) and (i, 2) are 5^{th} edge no. for the node ndexn(i) and other end node of that edge, (i, 3) and (i, 4) are 6^{th} edge no. for the node ndexn(i) and other end node of that edge and so on.
4. *edgenode*: Global variable of dimension(Maximum number of edges in the programme, 2) in which j^{th} row contains global number of starting node and ending node of j^{th} edge in two columns respectively.
5. *edgearr*: Local argument variable in the element loop which store in the process global edge numbers of all the edges of the element according to edge connectivity.

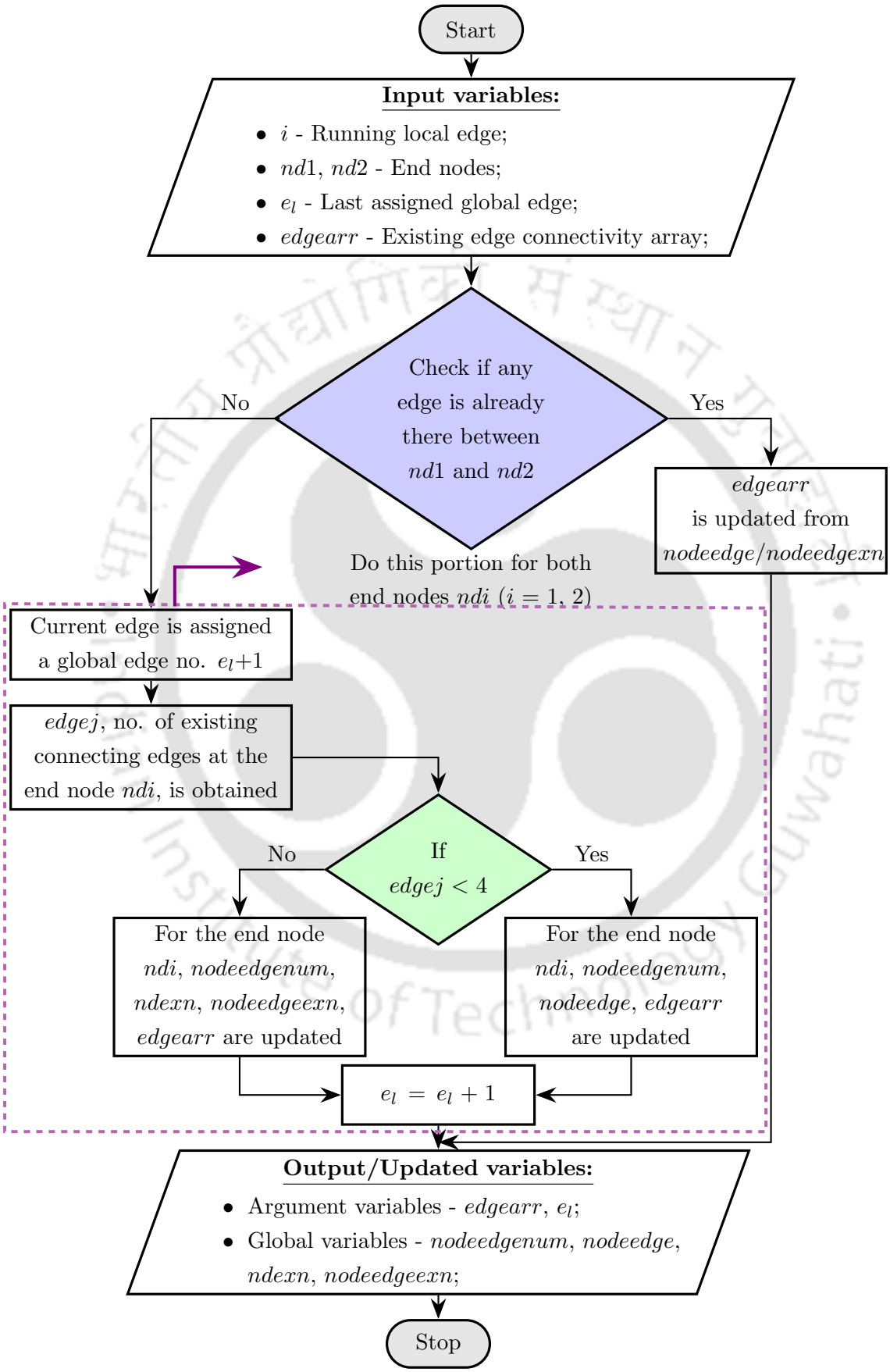


Figure 2.7: Flow chart of *edgedata* structure.

Table 2.3: Edge connectivity array of elements of the discretized domain.

Element number	Edge connectivity (Global edge no.)
1	1, 2, 3, 4
2	2, 5, 6, 7
3	8, 9, 4, 10
4	9, 11, 7, 12

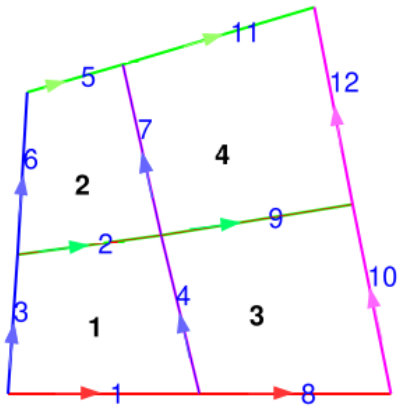


Figure 2.8: Element to edge connectivity sequence of the meshed domain

Global edge(i)	Starting node	End node
1	1	4
2	2	5
3	1	2
4	4	5
5	3	6
6	2	3
7	5	6
8	4	7
9	5	8
10	7	8
11	6	9
12	8	9

Table 2.4: *edgenode* array of nodes of the discretized domain.

In ***edgedata*** subroutine, there is a running counter called *edgej* which stores the associated number of edges of the starting end nodes of the current local edge. The information is interchanged between local *edgej* and global array *nodeedgenum*. In order to understand the update of the variables let us first summarize stored values in different variables after completion of element loop for first element.

(i) first, second, fourth and fifth rows of '*nodeedgenum*' array are assigned 'two' because first element has global nodes 1, 2, 4, and 5 (see Fig. 2.5b), and after the loop for element 1 each of these nodes associate with 'two' edges.

(ii) Last assigned global edge i.e., e_l with the value 4.

(iii) *nodeedge* will have following values as shown in Table 2.5.
For first row (for global node 1) we have 1 (associated first edge), 4 (other node of the first edge), 3 (associated second edge), and 2 (other node of the second

Table 2.5: *nodeedge* array of nodes after element loop of the first element

Global node	**Associated edge and other node of the edge(nodeedge(1:8))**							
	1^{st} edge	**Other node**	**2^{nd} edge**	**Other node**	**3^{rd} edge**	**Other node**	**4^{th} edge**	**Other node**
1	1	4	3	2	0	0	0	0
2	-3	1	2	5	0	0	0	0
3	0	0	0	0	0	0	0	0
4	-1	1	4	5	0	0	0	0
5	-2	2	-4	4	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0

edge). The reason for the negative sign in (2,1) position of *nodeedge* is that the first associated edge of the node 2 i.e. edge 3 is directing from the other node (1, stored in (2,2)) towards the current node 2 (see Fig. 2.8). For node 1, both the associated edges (1 and 3) are directing from the current node 1 to the respective other nodes. Therefore, those edge numbers are stored with '+' sign. In the first row (i.e., for first global edge) of *edgenode* array, starting node (1) and end node (4) are stored as shown in Fig. 2.4. Similarly for the other global edges of the element 1, starting and end nodes are stored in 2^{nd}, 3^{rd} and 4^{th} rows of this array. Table 2.4 shows such information for the edges 1 to 4.

(iv) First row of edge connectivity array is stored with the edge numbers of the first element, 1, 2, 3 and 4 as shown in the first row of Table 2.3.

Now for the next entity of the outer element loop i.e., for element 2, for local edge 1 we have starting and end nodes as 2 and 5 (see Fig. 2.5b) as per convention of Fig. 2.1a and Fig. 2.1b. At first, from *nodeedgenum*, we get the total number of already associated edges of the starting node. For our starting node 2, there are two associated edges. After that we get the other node numbers of the associated edges from the even columns of row 2 (our current node) of *nodeedge* array. If any of these other node number match with our current end node (5) then the associated edge no. (available in respective odd column) will be the global edge no. of that local edge. In this case the edge no. is 2, we will update the current edge number with global edge number 2. In this local edge loop we will not update last assigned global edge(e_l). Also, the first column of *edgearr* is assigned with 2.

For the 2^{nd} local edge starting and end nodes are 3 and 6 as shown in Fig. 2.5b. As there is no data available in 3^{rd} row of *nodeedge* array, the end node 3 is appearing for the first time. So, a new edge number is assigned just by updating e_l to $e_l + 1$ i.e., with the digit 5. Therefore, in the *nodeedge* array, first and second columns of

third(current end node) row are assigned with 5 (associated edge no.) and 6 (other end node) respectively. For end node number 6, the sixth row of this array is updated with -5 and 3 in the first two columns. Because the edge 5 is pointing away from the current end node 6, a negative symbol is assigned to the digit 5. 5^{th} row of the *edgenode* array is updated with the end node informations (3 and 6) of this new edge (5). In the 2^{nd} column of *edgearr*, this new edge number 5 is assigned. In *nodeedgenum*, existing number in third and sixth rows are incremented by 1 because global node 3 and global node 6 become associated with new edge '5' in this local edge loop.

For the third local edge, 2 and 3 are supplied as starting and end nodes (see Fig. 2.5b) respectively from the *edgend* subroutine. For starting node 2, there are two connected edges 3 and 2. Other end nodes of these connected edges (available in the even columns) i.e., 1 and 5 are not matching with other end node 3. As there is no already existing edge between these two nodes, last assigned global edge (e_l) is incremented from 5 to 6. After this, *edgearr* is updated with this value in the third row. In the second row of *nodeedge* array this new edge data (global edge 6) and its other end node (3) are updated in the fifth and sixth columns. For end node 3, the corresponding row of this array (3^{rd} row) is updated with -6 and 2 in the third and fourth columns. '-' symbol is assigned to the digit 6 because the edge 6 is pointing away from the current node 3. Also, the second row of *nodeedgenum* array(related to 2^{nd} global node) is updated from the previous count 2 to 3. Similarly, third row (related to 3^{rd} global node) is updated from 1 to 2. *edgenode* array is also updated with the end nodes (2 and 3) of the newly formed edge in the first and second columns of the corresponding (sixth) row.

For the fourth local edge, end nodes as 5 (starting node) and 6 (end node), e_l with 6 and existing *edgearr* (2, 5 and 6) are supplied. After checking the end node 6 with the other nodes (2 and 4) available in the fifth row of the *nodedge* array, e_l is updated from 6 to 7. This value is assigned to the running local edge. Now, fourth column of *edgearr* is updated with 7. Thus we complete the 2^{nd} row of Table 2.3 which shows the element to edge connectivity array of second element. Fig. 2.8 shows such element to edge connectivity for all the global elements of finite element meshed domain.

After this, *nodeedge* array is updated with the new edge number 7 and its associated other node 6 in the 5^{th} and 6^{th} columns of fifth row. Similarly, in the 6^{th} row corresponding to global node 6, third and fourth columns is updated with the digits -7 and 5 respectively. Also, the existing no. of edges in the fifth and sixth rows of *nodeedgenum* array are incremented by 1 and updated as 3 and 2. Finally, global edge number 7 and its end nodes (5 and 6) are stored corresponding row (seventh) of *edgenode* array as shown in Table 2.4.

After updation of all the global variables, program comes out from the inner (local edge) loop as shown in Fig. 2.9 and enters in to outer (element) loop after incrementing as $ele = ele + 1$. Now, the program runs for local edges of third element. This process repeats until all the discretized elements are finished. Table 2.3 shows the edge con-

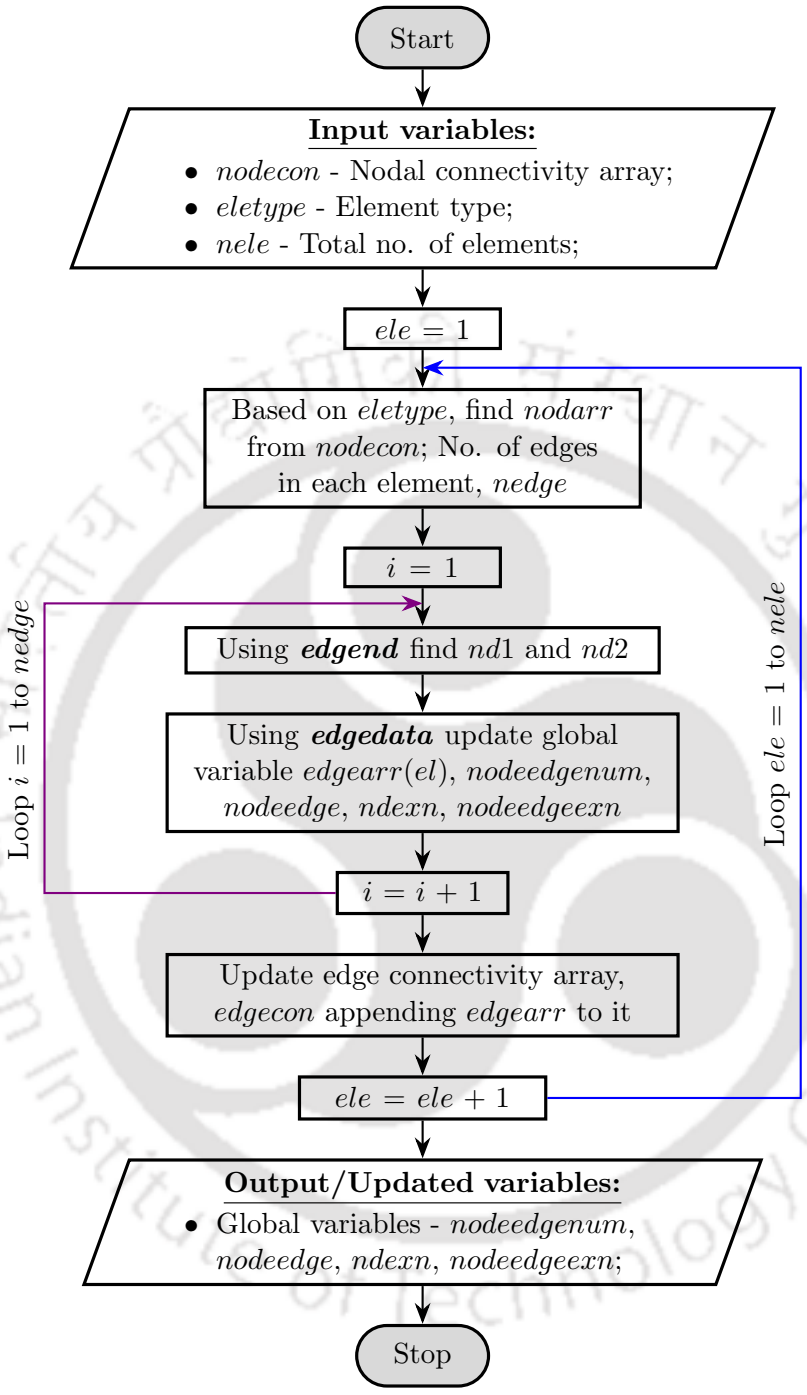


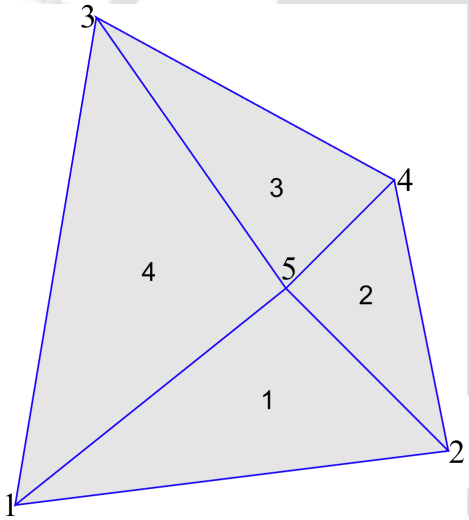
Figure 2.9: Flow chart of node to edge structure.

nectivity array for the entire domain; Table 2.4 shows the end nodes for all the edges; Table 2.6 shows complete *nodeedge* array of all the global nodes of the finite element meshed domain after finishing the outer loop for all four elements.

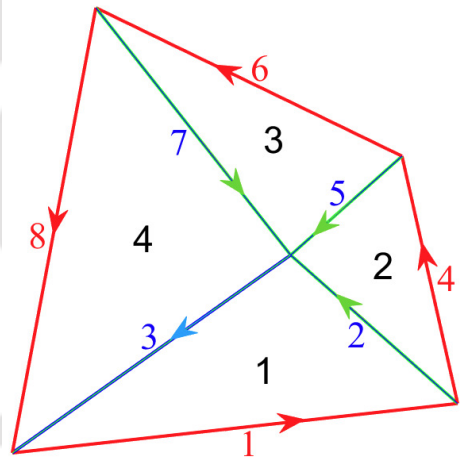
This algorithm can be implemented to other edge elements. These edge elements include 3-edge triangle, 8-edge triangle and 12-edge quadrilateral elements. In the following sections we discuss about the implementation of these elements.

Table 2.6: *nodeedge* array of global nodes of the meshed domain.

Global node	Total no. of connecting edges (nodeedenum)	Connecting edge and corresponding other node(nodeedge(1:8))							
		1^{st} edge	Other node	2^{nd} edge	Other node	3^{rd} edge	Other node	4^{th} edge	Other node
1	2	1	4	3	2	-	-	-	-
2	3	-3	1	2	5	6	3	-	-
3	2	5	6	-6	2	-	-	-	-
4	3	-1	1	4	5	8	7	-	-
5	4	-2	2	-4	4	7	6	9	8
6	3	-5	3	-7	5	11	9	-	-
7	2	-8	4	10	8	-	-	-	-
8	3	-9	5	-10	8	12	9	-	-
9	2	-11	6	-12	8	-	-	-	-



(a) Domain discretized with 3-node triangular element.

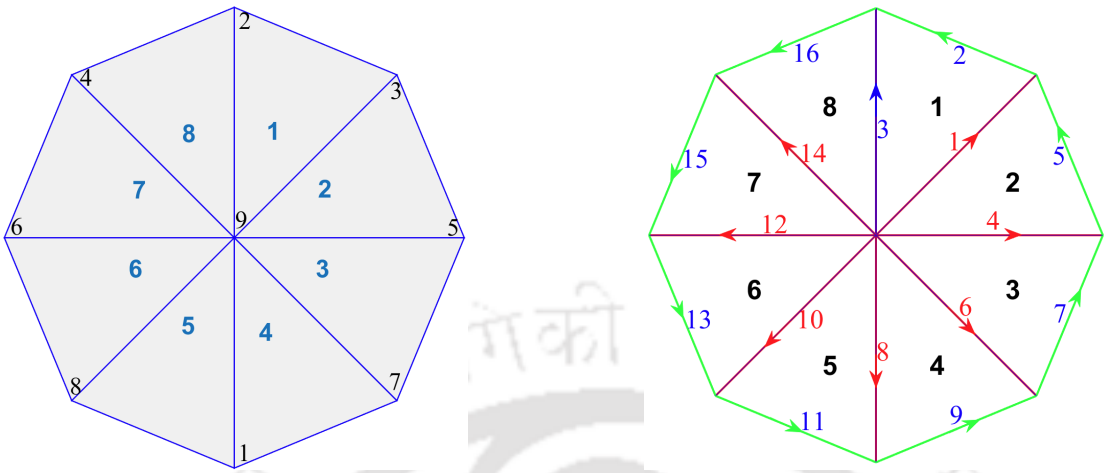


(b) Domain discretized with 3-edge triangular element.

Figure 2.10: Discretized domain

2.4.2 Conversion technique with three edge triangle: one illusrative example

Like four edge quadrilateral elements, edge data structure for three edge triangle can be constructed as described in section 2.4.1. Fig. 2.10a shows one general domain discretized with nodal elements which can be transformed into a domain discretized with three edge triangular elements as shown in Fig. 2.10b. If any node is shared by more than four edges then some additional data-structure is required as follows. In this al-



(a) Nodal connectivity of the elements. **(b)** Edge connectivity of the elements.

Figure 2.11: Discretized circular domain with coarse mesh

gorithm if any i^{th} node is shared by more than 4 edges then the information from the fifth edge onwards is stored in *nodeedgeexn*, *nndexn* and *ndexn* arrays. In entire geometry, there will be few nodes which will be associated with more than four edges. Our *nodeedge* array has number of rows as total number of nodes and number of column as 8 in order to accommodate first four associated edges which is very common. *nodeedgeexn* is initialized with number of rows far less than total number of nodes, it has 16 columns to accommodate next eight edges. When we come across a node which is associated with more than four edges *nndexn* is incremented by 1. Suppose in a flow, we have j^{th} occurrence of such node which is associated with more than four edges. Then *ndexn(j)* will store the corresponding node number. j^{th} row of *nodeedgeexn* will store the information of associated edge and other node from the fifth edge onwards. This can be understood from the case as shown in Fig. 2.11a and Fig. 2.11b Node no. 9 of the discretized domain is shared by 8 edges. So the information of the first four connecting edges of Node no. 9 and its other end nodes are stored in *nodeedge* array as shown in Table 2.7 after element loop for fifth element. But from the fifth to eighth associated edge of node 9, the information is stored in *nodeedgeexn* array. Table 2.8 shows updation of the information of *nodeedgeexn* array after the end of the element loop for fifth element. After the element loop runs for the remaining existing elements, *nodeedgeexn* array is updated as shown in Table 2.9. Here, *nndexn* will be 1 and *ndexn*(1) will be 9.

2.4.3 Illustrative example of conversion from six node triangle to eight edge triangle

We have used the conversion algorithm as discussed in section 2.4.1 to convert one general domain as shown in Fig. 2.12a discretized with four 6-node triangular elements into the domain meshed with four eight edge elements shown in Fig. 2.12b.

Table 2.7: *nodeedge* array of nodes after element loop of the fifth element.

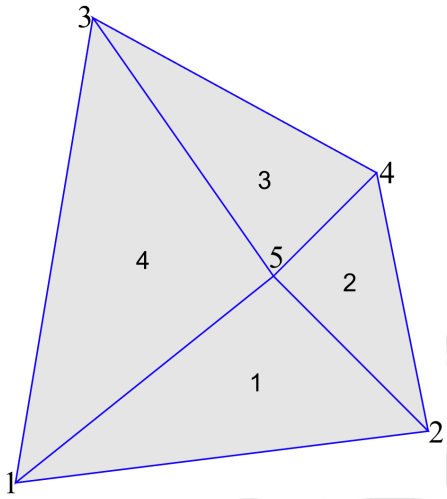
Global node	Total no. of connecting edges (nodeedgenum)	Associated edge and other node of the edge(nodeedge(1:8))							
		1^{st} edge	Other node	2^{nd} edge	Other node	3^{rd} edge	Other node	4^{th} edge	Other node
9	6	1	3	3	2	4	5	6	7
2	2	-2	3	-3	9	-	-	-	-
3	3	-1	9	2	2	-5	5	-	-
5	3	-4	9	5	3	-7	7	-	-
7	3	-6	9	7	5	-9	1	-	-
1	3	-8	9	9	7	-11	8	-	-
8	2	-10	9	11	1	-	-	-	-
-	-	-	-	-	-	-	-	-	-

Table 2.8: *nodeedgeexn* array of nodes after element loop of the fifth element

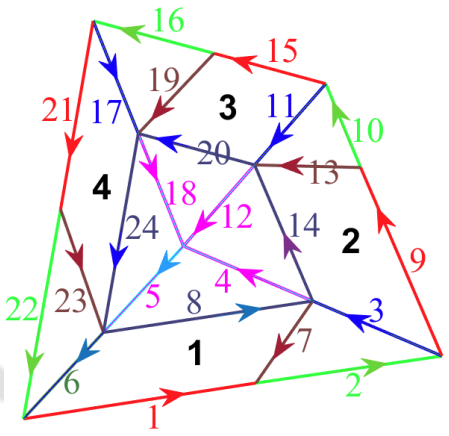
Global node	Associated edge and other node of the edge (nodeedgeexn(1:16))															
	5^{th} edge	other node	6^{th} edge	other node	7^{th} edge	other node	8^{th} edge	other node	9^{th} edge	other node	10^{th} edge	other node	11^{th} edge	other node	12^{th} edge	other node
9	8	1	10	8	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 2.9: *nodeedgeexn* array of nodes of the discretized domain after complete conversion.

Global node	Associated edge and other node of the edge (nodeedgeexn(1:16))															
	5^{th} edge	other node	6^{th} edge	other node	7^{th} edge	other node	8^{th} edge	other node	9^{th} edge	other node	10^{th} edge	other node	11^{th} edge	other node	12^{th} edge	other node
9	8	1	10	8	12	6	14	4	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-



(a) Domain meshed with 6-node triangular element.

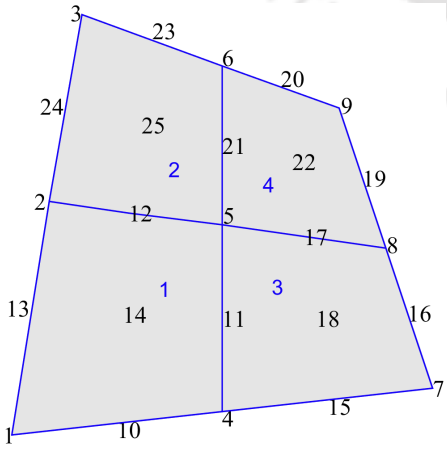


(b) Domain meshed with 8-edge triangular element.

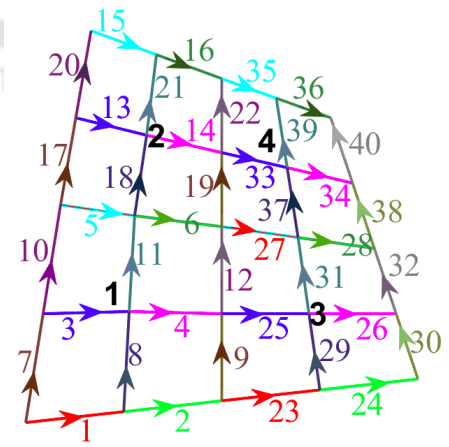
Figure 2.12: Meshed quadrilateral domain

2.4.4 Representative example of conversion in twelve edge quadrilateral

For the general domain shown in Fig. 2.13a which shows the domain meshed with four 9-node quadrilateral elements we have used the conversion algorithm to transform into the domain as shown in Fig. 2.13b. This generated domain is discretized with four twelve edge quadrilateral elements.



(a) Domain meshed with 9-node quadrilateral element.



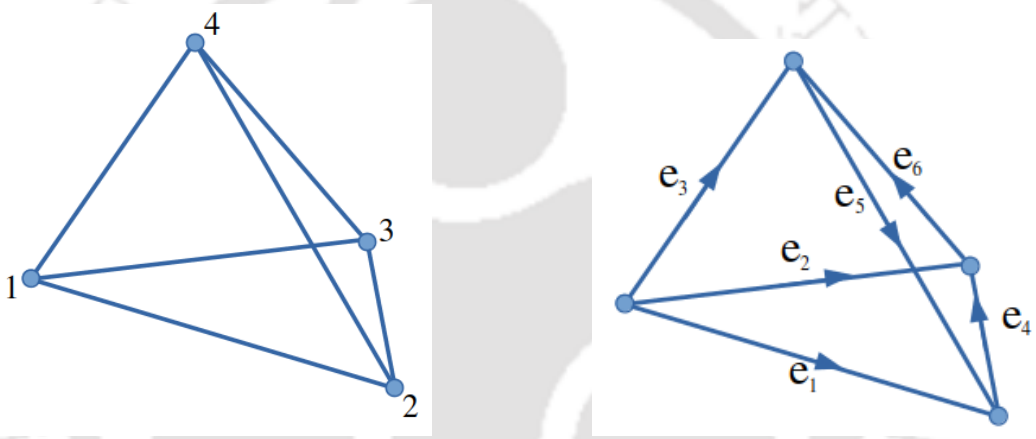
(b) Domain meshed with 12-edge quadrilateral element.

Figure 2.13: Discretized quadrilateral domain

2.5 Different edge elements in three dimensions

In this section, different three-dimensional edge elements adopted in our work is presented. Also, we have considered different three-dimensional nodal elements such as 8-node hexahedral and 10-node tetrahedral elements which are denoted as HexN8 and TetN10 respectively. These elements are well known and described in literature.

2.5.1 Six-edge tetrahedral element



(a) Element with four nodes. **(b)** Element with six edges.

Figure 2.14: Tetrahedral element

The six-edge tetrahedral element is formed from the four node tetrahedral element. Fig. 2.14a and Fig. 2.14b shows the local nodal connectivity and the desired local edge connectivity respectively of the tetrahedral element. For this edge element, with the node sets (1,2), (1,3), (1,4), (2,3), (4,2) and (3,4), six edges e_1, e_2, e_3, e_4, e_5 and e_6 respectively are formed. Six edge shape functions [1] of the element, are given as

$$
\begin{aligned}
\boldsymbol{v}_1 &= l_1(1-\eta-\zeta)\boldsymbol{\nabla}\xi + l_1\xi(\boldsymbol{\nabla}\eta + \boldsymbol{\nabla}\zeta),\\
\boldsymbol{v}_2 &= l_2(1-\xi-\eta)\boldsymbol{\nabla}\zeta + l_2\zeta(\boldsymbol{\nabla}\xi + \boldsymbol{\nabla}\eta),\\
\boldsymbol{v}_3 &= l_3(1-\xi-\zeta)\boldsymbol{\nabla}\eta + l_3\eta(\boldsymbol{\nabla}\xi + \boldsymbol{\nabla}\zeta),\\
\boldsymbol{v}_4 &= l_4(\xi\boldsymbol{\nabla}\zeta - \zeta\boldsymbol{\nabla}\xi),\\
\boldsymbol{v}_5 &= l_5(\eta\boldsymbol{\nabla}\xi - \xi\boldsymbol{\nabla}\eta), \text{and}\\
\boldsymbol{v}_6 &= l_6(\zeta\boldsymbol{\nabla}\eta - \eta\boldsymbol{\nabla}\zeta),
\end{aligned}
$$

where l_1, l_2, ..., l_6 are the lengths of the six edges of the tetrahedral element. We represented this element as TetE6.

By using the following relation we can calculate $\boldsymbol{\nabla} \times \boldsymbol{E}$,

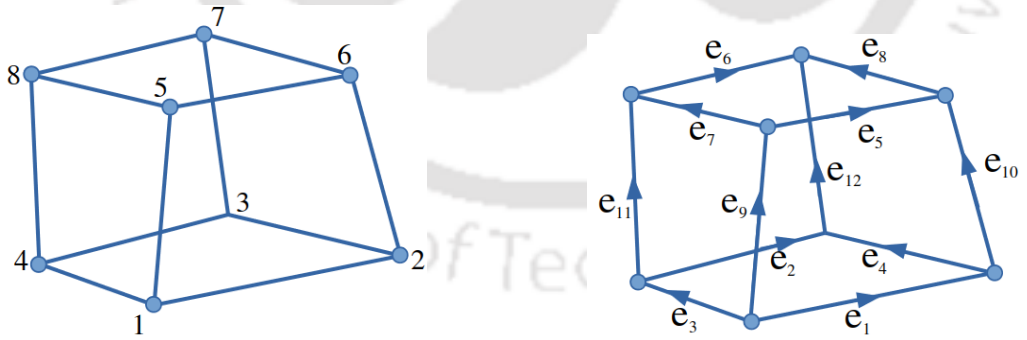
$$\boldsymbol{\nabla} \times \boldsymbol{E} = \begin{bmatrix} \frac{\partial v_{1z}}{\partial y} - \frac{\partial v_{1y}}{\partial z} & \frac{\partial v_{2z}}{\partial y} - \frac{\partial v_{2y}}{\partial z} & \cdots & \frac{\partial v_{5z}}{\partial y} - \frac{\partial v_{5y}}{\partial z} & \frac{\partial v_{6z}}{\partial y} - \frac{\partial v_{6y}}{\partial z} \\ \frac{\partial v_{1x}}{\partial z} - \frac{\partial v_{1z}}{\partial x} & \frac{\partial v_{2x}}{\partial z} - \frac{\partial v_{2z}}{\partial x} & \cdots & \frac{\partial v_{5x}}{\partial z} - \frac{\partial v_{5z}}{\partial x} & \frac{\partial v_{6x}}{\partial z} - \frac{\partial v_{6z}}{\partial x} \\ \frac{\partial v_{1y}}{\partial x} - \frac{\partial v_{1x}}{\partial y} & \frac{\partial v_{2y}}{\partial x} - \frac{\partial v_{2x}}{\partial y} & \cdots & \frac{\partial v_{5y}}{\partial x} - \frac{\partial v_{5x}}{\partial y} & \frac{\partial v_{6y}}{\partial x} - \frac{\partial v_{6x}}{\partial y} \end{bmatrix} \begin{Bmatrix} E_1 \\ E_2 \\ \vdots \\ E_5 \\ E_6 \end{Bmatrix} = \boldsymbol{B}\hat{\boldsymbol{E}},$$

where E_1, E_2, ..., E_6 are the tangential electric field components along edges e_1, e_2, ..., e_6 respectively. We can use various relations as given in Eq. (2.9), to obtain different constituent terms in the $\boldsymbol{B}$-matrix.

$$\begin{bmatrix} \frac{\partial v_{1x}}{\partial x} & \frac{\partial v_{1x}}{\partial y} & \frac{\partial v_{1x}}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{\partial v_{1x}}{\partial \xi} & \frac{\partial v_{1x}}{\partial \eta} & \frac{\partial v_{1x}}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \Gamma_{11} & \Gamma_{21} & \Gamma_{31} \\ \Gamma_{12} & \Gamma_{22} & \Gamma_{32} \\ \Gamma_{13} & \Gamma_{23} & \Gamma_{33} \end{bmatrix}, \tag{2.9}$$

where partial derivatives $\frac{\partial v_{1x}}{\partial \xi}$, $\frac{\partial v_{1x}}{\partial \eta}$, and $\frac{\partial v_{1x}}{\partial \zeta}$ can be obtained explicitly.

2.5.2 Twelve-edge hexahedral element



(a) Hexahedral element with eight nodes.

(b) Hexahedral element with twelve edges.

Figure 2.15: Hexahedral element

In three-dimensions, another important edge element is twelve-edge hexahedral element, which is denoted as HexE12 . In this work, we have generated twelve-edge element (Fig. 2.15b) from eight node finite element (Fig. 2.15a). The twelve edges, namely, e_1, e_2, e_3, ..., e_{12} are formed by connecting local node pairs of 8-node hexahedral element as (1,2), (4,3), (1,4), (2,3), (5,6), (8,7), (5,8), (6,7), (1,5), (2,6), (4,8) and (3,7) respectively.

Twelve edge shape functions [1] are

$$\boldsymbol{v}_1 = \frac{l_1}{8}(1-\eta)(1-\zeta)\boldsymbol{\nabla}\xi, \quad \boldsymbol{v}_2 = \frac{l_2}{8}(1+\eta)(1-\zeta)\boldsymbol{\nabla}\xi, \quad \boldsymbol{v}_3 = \frac{l_3}{8}(1-\xi)(1-\zeta)\boldsymbol{\nabla}\eta,$$
$$\boldsymbol{v}_4 = \frac{l_4}{8}(1+\xi)(1-\zeta)\boldsymbol{\nabla}\eta, \quad \boldsymbol{v}_5 = \frac{l_5}{8}(1-\eta)(1+\zeta)\boldsymbol{\nabla}\xi, \quad \boldsymbol{v}_6 = \frac{l_6}{8}(1+\eta)(1+\zeta)\boldsymbol{\nabla}\xi,$$
$$\boldsymbol{v}_7 = \frac{l_7}{8}(1-\xi)(1+\zeta)\boldsymbol{\nabla}\eta, \quad \boldsymbol{v}_8 = \frac{l_8}{8}(1+\xi)(1+\zeta)\boldsymbol{\nabla}\eta, \quad \boldsymbol{v}_9 = \frac{l_9}{8}(1-\xi)(1-\eta)\boldsymbol{\nabla}\zeta,$$
$$\boldsymbol{v}_{10} = \frac{l_{10}}{8}(1+\xi)(1-\eta)\boldsymbol{\nabla}\zeta, \quad \boldsymbol{v}_{11} = \frac{l_{11}}{8}(1-\xi)(1+\eta)\boldsymbol{\nabla}\zeta, \quad \text{and}$$
$$\boldsymbol{v}_{12} = \frac{l_{12}}{8}(1+\xi)(1+\eta)\boldsymbol{\nabla}\zeta,$$

where l_1, l_2, l_3, ... and l_{12} are lengths of the twelve edges of the element. $\boldsymbol{\nabla}\times\boldsymbol{E}$ can be calculated by using the following relation,

$$\boldsymbol{\nabla}\times\boldsymbol{E} = \begin{bmatrix} \frac{\partial v_{1z}}{\partial y} - \frac{\partial v_{1y}}{\partial z} & \frac{\partial v_{2z}}{\partial y} - \frac{\partial v_{2y}}{\partial z} & \cdots & \frac{\partial v_{11z}}{\partial y} - \frac{\partial v_{11y}}{\partial z} & \frac{\partial v_{12z}}{\partial y} - \frac{\partial v_{12y}}{\partial z} \\ \frac{\partial v_{1x}}{\partial z} - \frac{\partial v_{1z}}{\partial x} & \frac{\partial v_{2x}}{\partial z} - \frac{\partial v_{2z}}{\partial x} & \cdots & \frac{\partial v_{11x}}{\partial z} - \frac{\partial v_{11z}}{\partial x} & \frac{\partial v_{12x}}{\partial z} - \frac{\partial v_{12z}}{\partial x} \\ \frac{\partial v_{1y}}{\partial x} - \frac{\partial v_{1x}}{\partial y} & \frac{\partial v_{2y}}{\partial x} - \frac{\partial v_{2x}}{\partial y} & \cdots & \frac{\partial v_{11y}}{\partial x} - \frac{\partial v_{11x}}{\partial y} & \frac{\partial v_{12y}}{\partial x} - \frac{\partial v_{12x}}{\partial y} \end{bmatrix} \begin{Bmatrix} E_1 \\ E_2 \\ \vdots \\ E_{11} \\ E_{12} \end{Bmatrix} = \boldsymbol{B}\hat{\boldsymbol{E}},$$

where E_1, E_2, E_3, ... and E_{12} are tangential electric fields along the twelve edges of the hexahedral element. The different components of the $\boldsymbol{B}$-matrix are obtained combining various derivatives as follows:

$$\begin{bmatrix} \frac{\partial v_{1x}}{\partial x} & \frac{\partial v_{1x}}{\partial y} & \frac{\partial v_{1x}}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{\partial v_{1x}}{\partial \xi} & \frac{\partial v_{1x}}{\partial \eta} & \frac{\partial v_{1x}}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \Gamma_{11} & \Gamma_{21} & \Gamma_{31} \\ \Gamma_{12} & \Gamma_{22} & \Gamma_{32} \\ \Gamma_{13} & \Gamma_{23} & \Gamma_{33} \end{bmatrix}.$$

For the edge shape functions $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, ... , $\boldsymbol{v}_{12}$, partial derivatives with respect to ξ, η and ζ i.e. $\frac{\partial v_{1x}}{\partial \xi}$, $\frac{\partial v_{1x}}{\partial \eta}$, $\frac{\partial v_{1x}}{\partial \eta}$, ..., $\frac{\partial v_{12y}}{\partial \eta}$, $\frac{\partial v_{12z}}{\partial \zeta}$ are obtained by finite difference method.

2.6 Conversion technique in three-dimensions

This section discusses the conversion process involved in generating edge connectivity information from three dimensional conventional nodal elements.

2.6.1 Conversion technique with six edge tetrahedral element: one representative example

In this section, we outline the conversion programme flow with respect to one nodal tetrahedral mesh as shown in Fig. 2.16. For that triangular prismatic domain, Table 2.10 gives the information about the nodal connectivity of the element.

Table 2.10: Element nodal connectivity of discretized triangular prism.

Element number	Nodal connectivity (Global node no.)
1	1, 3, 2, 4
2	2, 5, 3, 4
3	3, 5, 6, 4

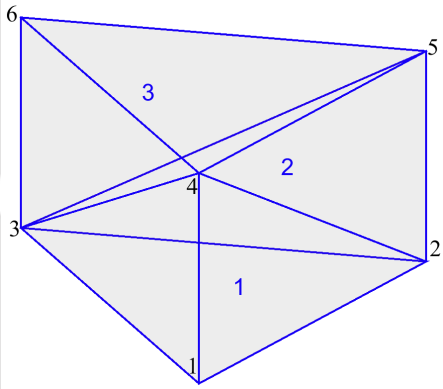


Figure 2.16: Domain discretized with 4-node tetrahedral element.

The flow contains two major loops such as the outer loop over element numbers and the inner loop over local edge numbers as shown in Fig. 2.9.

In the inner loop, the edge data structure is created by two subroutines, namely, *edgend* and *edgedata*. One counter variable e_l keeps track of creation of new edges in the flow.

In the inner loop, for each local edge, subroutine *edgend* (see flowchart 2.6) detects two end nodes ($nd1$ and $nd2$) of the edge for the element types, using the nodal connectivity of the element. For six edge tetrahedral element, we have followed local node to edge conversion as shown in Fig. 2.14. Table 2.11 consists of end nodes for all the six edges of the three elements, generated by this subroutine.

For the local edge i, end node informations ($nd1$ and $nd2$) are supplied to *edgedata* subroutine (Fig. 2.7). *edgedata* is the main subroutine in creating the edge data structure where different variables, like *nodeedgenum*, *nodeedge*, *nodeedgeexn*, *edgearr*, *nnedxn*, *ndexn* etc., are updated for both the end nodes.

Table 2.11: End nodes ($nd1$, $nd2$) for all the edges in the domain (Fig 2.16).

Local Edge number	**Element number**		
	1	**2**	**3**
1	1, 3	2, 5	3, 5
2	1, 2	2, 3	3, 6
3	1, 4	2, 4	3, 4
4	3, 2	5, 3	5, 6
5	4, 3	4, 5	4, 5
6	2, 4	3, 4	6, 4

Let us understand the flow in the *edgedata* subroutine for the second element when all the above variables have been updated for the first element as below.

Table 2.12: *nodeedge* array after the completion of first element loop

1^{st} edge	**Other node**	**2^{nd} edge**	**Other node**	**3^{rd} edge**	**Other node**	**4^{th} edge**	**Other node**
1	3	2	2	3	4	0	0
-2	1	-4	3	6	4	0	0
-1	1	4	2	-5	4	0	0
-3	1	5	3	-6	2	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(i) The first element has global nodes 1-4 (Fig. 2.16). After the loop for $ele = 1$, edge data structure have been updated for the first element. Hence, all these four nodes have been associated with three edges. Therefore, we have $nodeedgenum(1:4) = 3$ and remaining entries in *nodeedgenum* are 0.

(ii) Six edges have been constructed in the first loop. Hence, $e_l = 6$.

(iii) *nodeedge* array has the values as shown in Table 2.12 where i^{th}row is for the i^{th} global node.

(iv) After first element loop, no nodes are associated with more than 4 edges. So, *nodeedgeexn*, *nndexn*, *ndexn* arrays remain in initialized condition with zero entries.

(v) After the loop $ele = 1$, first row of the *edgecon* array have been updated as $1, 2, 3, 4, 5, 6$.

After various global variables are updated in the first element loop, the element loop runs for the second element and flow enters into the local edge loop. Now for the first local edge, different variables store the informations as follows.

(i) $nd1$ and $nd2$ from $edgend$: 2 & 5,

(ii) For $nd1$ (2),
- Existing associated edge : 2, 4 & 6 (odd columns of 2^{nd} row of Table 2.12).
- Respective other nodes are 1, 3 & 4 (even columns of 2^{nd} row of Table 2.12).

(iii) 1, 3 & 4 are not matching with $nd2 = 5$,

(iv) Therefore, there is no existing edge between $nd1 = 2$ and $nd2 = 5$.

(v) e_l is incremented from 6 to 7. So, edge 7 is created from node 2 to node 5.

(vi) So, $edgearr(1) = 7$.

(vii) $nodeedge(2, 7) = 7$, $nodeedge(2, 8) = 5$.

(viii) Consequently, for $nd2$ (5),
- $nodeedge(5, 1) = -7$, $nodeedge(5, 2) = 2$.
- $nodeedge(5, 1)$ is negative as for edge 7, node 5 is the end node.

(ix) Now, $nodeedgenum(2) = 4$, $nodeedgenum(5) = 1$.

(x) For the 7^{th} edge, $edgenode(7, 1) = 2$, $edgenode(7, 2) = 5$.

For the second local edge, global variables are updated as given below.

(i) $nd1$ and $nd2$ from $edgend$: 2 & 3,

(ii) For $nd1$ (2),
- Existing associated edge : 2, 4, 6 & 7 (odd columns of 2^{nd} row of Table 2.12).
- Respective other nodes are 1, 3, 4 & 5 (even columns of 2^{nd} row of Table 2.12).

(iii) From the other nodes, 3 is matching with $nd2 = 3$.

(iv) Therefore, there is an already existing edge between $nd1 = 2$ and $nd2 = 3$.

(v) So, $nodeedge(2, 3) = 4$ is assigned as global edge number to the 2^{nd} local edge.

(vi) Therefore, $edgearr(2) = 4$.

(vii) e_l is not incremented from previous value 7. Hence, $e_l = 7$.

For the third local edge, we have end nodes as $nd1 = 2$ & $nd2 = 4$ from $edgend$. We can see from $nodeedge$ array (Table 2.12) that already there exists edge 6 between $nd1 = 2$ & $nd2 = 4$. As the edge is repeating, we follow the similar arguments as discussed for the second local edge.

For the fourth local edge, we have end nodes as $nd1 = 5$ & $nd2 = 3$ from $edgend$. After observing existing $nodeeage$ array (Table 2.12), we can see that there is no edge between $nd1 = 5$ & $nd2 = 3$. Hence, one new edge and associated data structure will be created, following the similar arguments as discussed for the first local edge.

In the similar manner, we update different variables for remaining fifth and sixth local edges for element 2. After completing the inner local edge loop, ele is incremented to 3, and the flow enters into the third element loop. This procedure is repeated for all the discretized elements.

Let us understand the updation of the variables like $nodeedgeexn$, $nndexn$, $ndexn$ arrays, when any node has more than four connecting edges. We can see from Fig. 2.16

that global node numbers 3 and 4 are associated with more than four connecting edges. Fig. 2.16 also reveals that upto the second element loop, both node numbers 3 and 4 are connected with four edges. Table 2.13 represents the *nodeedge* array after completion of the second element loop.

Table 2.13: *nodeedge* array after completion of the second element loop

1st edge	**Other node**	**2nd edge**	**Other node**	**3rd edge**	**Other node**	**4th edge**	**Other node**
1	3	2	2	3	4	0	0
-2	1	-4	3	6	4	7	5
-1	1	4	2	-5	4	-8	5
-3	1	5	3	-6	2	9	5
-7	2	8	3	-9	4	0	0
0	0	0	0	0	0	0	0

For the third element, nodal connectivity is 3, 5, 6, 4 (from Table 2.10). From Fig. 2.14, we can see that node 3 (local node 1) is connected with the local edges e_1, e_2 and e_3 whereas node 4 (local node 4) is connected with the local edges e_3, e_5 and e_6.

Now, for local edge e_1,

(i) $nd1$ and $nd2$ from $edgend$: 3 & 5,

(ii) For $nd1$ (3),

- Existing associated edges are 1, 4, 5 & 8 (odd columns of 3$^{\text{rd}}$ row of Table 2.13).
- Respective other nodes are 1, 2, 4 & 5 (even columns of 3$^{\text{rd}}$ row of Table 2.13).

(iii) Between 3 & 5, edge 8 exists. There is no creation of any new edge.

(iv) So, $edgearr(1) = 8$.

(v) $nodeedge$, $nodeedgeexn$, $nndexn$, and $ndexn$ remain unchanged.

Now, for the second local edge e_2,

(i) $nd1$ and $nd2$ from $edgend$: 3 & 6,

(ii) For $nd1$ (3),

- Existing associated edges are 1, 4, 5 & 8 (odd columns of 3$^{\text{rd}}$ row of Table 2.13).
- Respective other nodes are 1, 2, 4 & 5 (even columns of 3$^{\text{rd}}$ row of Table 2.13).

(iii) 1, 2, 4 & 5 are not matching with $nd2 = 6$,

(iv) Therefore, there is no existing edge between $nd1 = 3$ and $nd2 = 6$.

(v) e_l is incremented from 9 to 10. So, edge 10 is created from node 3 to node 6.

(vi) So, $edgearr(2) = 10$.

(vii) $edgej = 5$ (>4), therefore, $nndexn = 1$ and $ndexn(1) = 3$.

(viii) For the fifth connecting edge of node 3, $nodeedgeexn(1,1) = 10$, $nodeedgeexn(1,2) = 6$.

(ix) For $nd2$ (6),

- $nodeedge(6,1)$ = -10, $nodeedge(6,2) = 3$.
- $nodeedge(6,1)$ is negative as for edge 10, node 6 is the end node.

(x) Now, $nodeedgenum(3) = 5$, and $nodeedgenum(6) = 1$.

(xi) For the new edge 10, $edgenode(10,1) = 3$, and $edgenode(10,2) = 6$.

Similarly, for the global node 4, in the case of sixth local edge loop, a new edge 12 is formed with the other node 6. This new edge 12 is the fifth connecting edge for the node 4. Therefore, *nndexn*, *ndexn*, and *nodeedgeexn* are further updated.

For this problem, final *nodedgeexn* array is shown in Table 2.14. Table 2.15 shows the final *nodeedge* array for all the global nodes after finishing all the element loops. Table 2.16 shows the edge connectivity array of entire domain. For all the global edges, corresponding end nodes are shown in Table 2.17. Fig. 2.17 represents the edge connectivities of the triangular prismatic domain.

Table 2.14: *nodeedgeexn* array for the triangular prismatic domain.

5th edge	**other node**	6th edge	**other node**	7th edge	**other node**	8th edge	**other node**	9th edge	**other node**	10th edge	**other node**	11th edge	**other node**	12th edge	**other node**
10	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2.15: *nodeedge* array for the meshed triangular prism.

1st edge	**Other node**	**2nd edge**	**Other node**	**3rd edge**	**Other node**	**4th edge**	**Other node**
1	3	2	2	3	4	0	0
-2	1	-4	3	6	4	7	5
-1	1	4	2	-5	4	-8	5
-3	1	5	3	-6	2	9	5
-7	2	8	3	-9	4	11	6
-10	3	-11	5	12	4	0	0

Table 2.16: Edge connectivity information of global elements of the finite element discretized domain.

Element number	Element to edge connectivity (Global edge no.)
1	1, 2, 3, 4, 5, 6
2	7, 4, 6, 8, 9, 5
3	8, 10, 5, 11, 9, 12

Table 2.17: *edgenode* array data of nodes of the discretized domain.

Global edge(i)	Starting node	End node
1	1	3
2	1	2
3	1	4
4	3	2
5	4	3
6	2	4
7	2	5
8	5	3
9	4	5
10	3	6
11	5	6
12	6	4

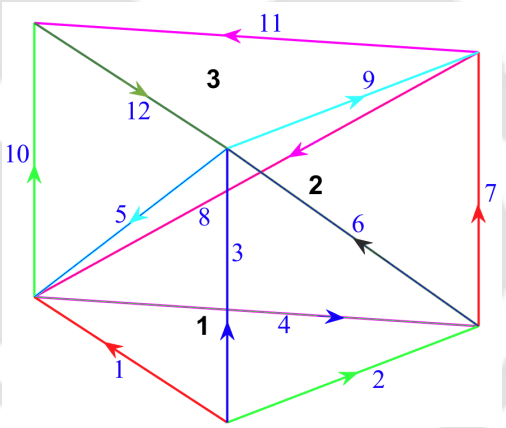


Figure 2.17: Domain discretized with 6-edge tetrahedral element.

2.6.2 Illustrative example of conversion from nodal hexahedral element to nodal-to-edge hexahedral element

An irregular hexahedral domain is discretized with two 8-node hexahedral elements, as shown in Fig. 2.18. For this domain, Table 2.18 gives the nodal connectivity information of the discretized elements. We have applied the conversion algorithm to transform this domain to the domain meshed with 12-edge hexahedral elements as shown in Fig. 2.19. For the 12-edge hexahedral element, we have followed the local nodal to edge conversion as shown in Fig. 2.15. Table 2.19 gives the edge connectivity array information for the transformed domain (shown in Fig. 2.19).

Table 2.18: Element nodal connectivity of discretized irregular hexahedral prism.

Element number	Nodal connectivity (Global node no.)
1	1, 2, 4, 3, 5, 6, 8, 7
2	5, 6, 8, 7, 9, 10, 12, 11

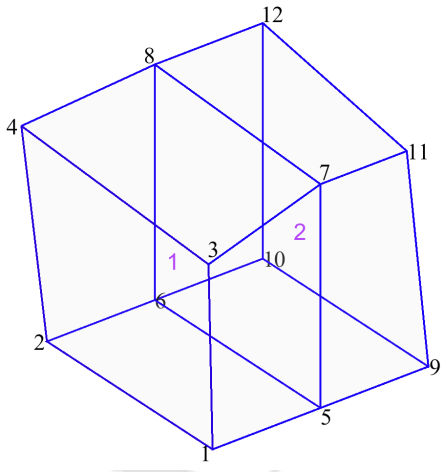


Figure 2.18: Discretized irregular hexahedral domain with 8-node hexahedral element.

Table 2.19: Edge connectivity information of global elements of the finite element discretized domain.

E*	Element to edge connectivity (Global edge no.)
1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
2	5, 6, 7, 8, 13, 14, 15, 16, 17, 18, 19, 20

E*-Element number

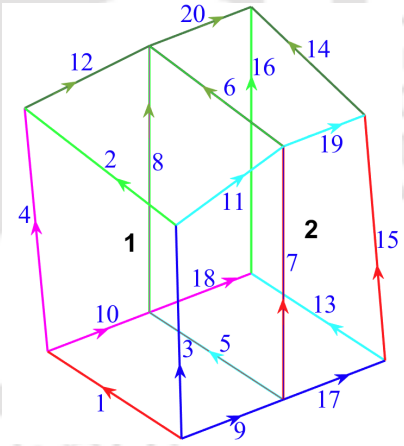


Figure 2.19: Discretized irregular hexahedral domain with 12-edge hexahedral element.

For the whole domain, for all the global nodes, Table 2.20 gives the final *nodeedge* array information, and Table 2.21 shows the *nodeedgenum* array data. Table 2.22 provides the *edgenode* array for all the global edges.

Table 2.20: *nodeedge* array for the meshed irregular hexahedral prism.

1^{st} edge	Other node	2^{nd} edge	Other node	3^{rd} edge	Other node	4^{th} edge	Other node
1	2	3	3	9	5	0	0
-1	1	4	4	10	6	0	0
2	4	-3	1	11	7	0	0
-2	3	-4	2	12	8	0	0
5	6	7	7	-9	1	17	9
-5	5	8	8	-10	2	18	10
6	8	-7	5	-11	3	19	11
-6	7	-8	6	-12	4	20	12
13	10	15	11	-17	5	0	0
-13	9	16	12	-18	6	0	0
14	12	-15	9	-19	7	0	0
-14	11	-16	10	-20	8	0	0

Table 2.21: *nodeedgenum* array data of nodes of the discretized domain.

Global node(i)	Number of connecting edges
1	3
2	3
3	3
4	3
5	4
6	4
7	4
8	4
9	3
10	3
11	3
12	3

Table 2.22: *edgenode* array data of nodes of the discretized domain.

Global edge(i)	Starting node	End node
1	1	2
2	3	4
3	1	3
4	2	4
5	5	6
6	7	8
7	5	7
8	6	8
9	1	5
10	2	6
11	3	7
12	4	8
13	9	10
14	11	12
15	9	11
16	10	12
17	5	9
18	6	10
19	7	11
20	8	12

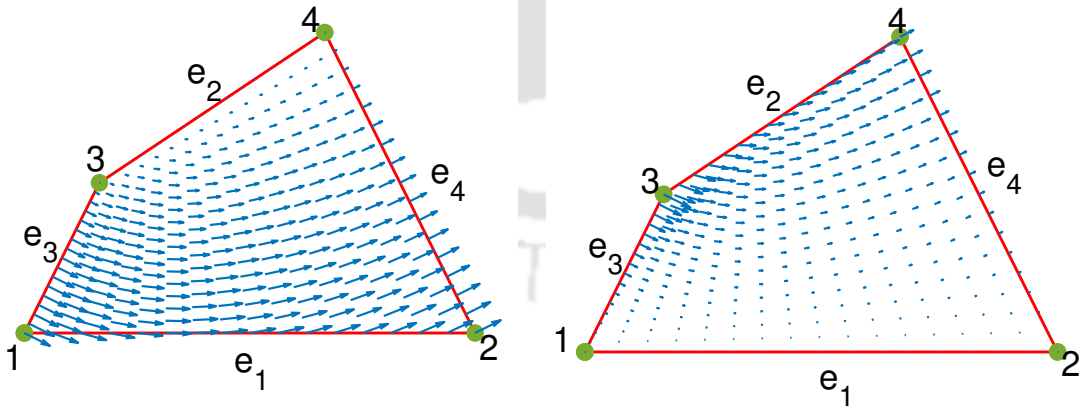
2.7 Inter-element continuity in edge elements

The shape functions of edge elements are constructed or developed in such a way that they impose tangential continuity condition across adjacent elements that share a com-

mon edge. This characteristic can be understood by examine the construction of these shape functions for both triangular and quadrilateral edge elements in 2D. For a 4-edge quadrilateral element, the edge shape functions of four edges, in terms of the local coordinates ξ and η, are given as (see [1])

$$\begin{Bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \boldsymbol{v}_3 \\ \boldsymbol{v}_4 \end{Bmatrix} = \begin{Bmatrix} \frac{l_1}{4}(1-\eta)\boldsymbol{\nabla}\xi \\ \frac{l_2}{4}(1+\eta)\boldsymbol{\nabla}\xi \\ \frac{l_3}{4}(1-\xi)\boldsymbol{\nabla}\eta \\ \frac{l_4}{4}(1+\xi)\boldsymbol{\nabla}\eta \end{Bmatrix},$$

where l_1, l_2, l_3, and l_4 are lengths of edges e_1, e_2, e_3, and e_4, respectively. The nature of these shape functions guarantee tangential continuity across inter-element boundaries, since they have tangential components only along the associated edges and none along the other edges. For example, the term involving $\boldsymbol{\nabla}\xi$ in the shape functions $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ ensures a tangential component only along edges e_1 and e_2 respectively. However, there is no tangential component of both $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ along the other edges e_3 and e_4, as ξ is constant along these edges. This behaviour is also visually represented through vector plots of the shape functions $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, as shown in Fig. 2.20a and Fig. 2.20b. These plots illustrate how a vector basis function maintains a tangential component along the associated edge and no tangential component along the remaining edges of the element.

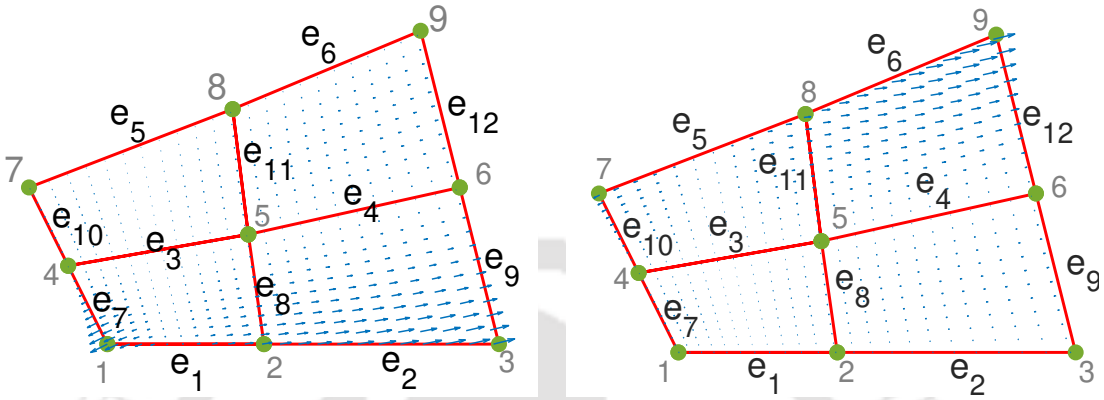


(a) Variation of $\boldsymbol{v}_1$ for the edge e_1. **(b)** Variation of $\boldsymbol{v}_2$ for the edge e_2.

Figure 2.20: Vector plots of edge shape functions for 4-edge quadrilateral element.

A similar interpretation applies to higher-order quadrilateral edge elements, as their shape functions are built on the principles established by lower-order vector basis functions. For example, for a 12-edge quadrilateral element, the edge shape functions for edges e_2 and e_6 are given as $\boldsymbol{v}_2 = \frac{l_2}{2}\eta(\eta-1)(\xi+0.5)\boldsymbol{\nabla}\xi$ and $\boldsymbol{v}_6 = \frac{l_6}{2}\eta(\eta+1)(\xi+0.5)\boldsymbol{\nabla}\xi$.

Fig. 2.21a and Fig. 2.21b illustrate the behaviour of the $\boldsymbol{v}_2$ and $\boldsymbol{v}_6$, showing how these basis functions ensure a tangential component only along the corresponding edges e_2 and e_6 respectively.



(a) Variation of $\boldsymbol{v}_2$ for the edge e_2. **(b)** Variation of $\boldsymbol{v}_6$ for the edge e_6.

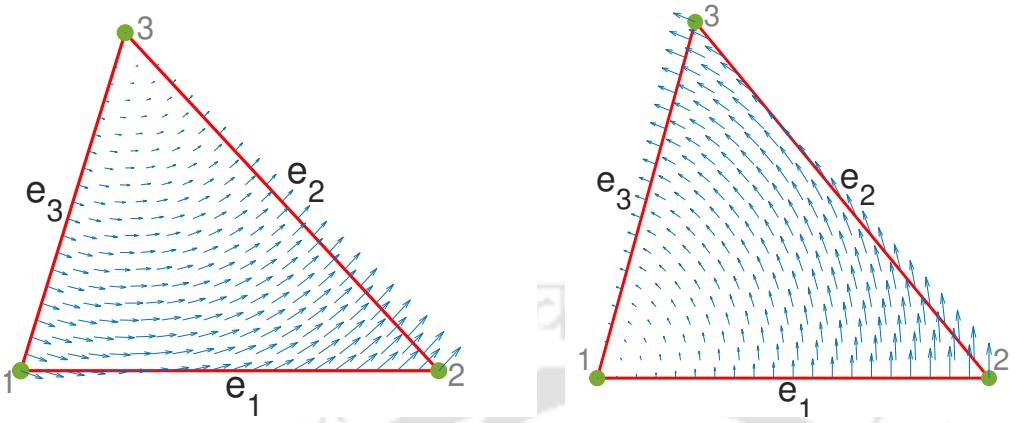
Figure 2.21: Vector plots of edge shape functions of 12-edge quadrilateral element.

Now, let us understand the behaviour of shape functions of triangular edge elements, focusing on a 3-edge triangular element with three edges e_1, e_2 and e_3. The edge shape functions $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$, expressed in natural coordinates ξ and η, are given as (see [1])

$$\begin{Bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \boldsymbol{v}_3 \end{Bmatrix} = \begin{Bmatrix} l_1(\xi\boldsymbol{\nabla}\eta - \eta\boldsymbol{\nabla}\xi) \\ l_2(-\eta\boldsymbol{\nabla}\xi - (1-\xi)\boldsymbol{\nabla}\eta) \\ l_3((1-\eta)\boldsymbol{\nabla}\xi + \xi\boldsymbol{\nabla}\eta) \end{Bmatrix},$$

where l_1, l_2 and l_3 are the edge lengths of three edges. Consider the unit vector $\boldsymbol{n}_1$ along the edge e_1, which is formed by joining the nodes 1 and 2, as shown in Fig. 2.22a. Using shape function $\boldsymbol{v}_1$, we find that $\boldsymbol{n}_1 \cdot \boldsymbol{\nabla}\xi = -1/l_1$ and $\boldsymbol{n}_1 \cdot \boldsymbol{\nabla}\eta = 1/l_1$, since ξ is a linear functions that varies from unity at node 1 and zero at node 2, and η is unity at node 2 and zero at node 1. Therefore, we have $\boldsymbol{n}_1.\boldsymbol{v}_1 = \xi + \eta = 1$ along the entire length of edge 1. This implies that $\boldsymbol{v}_1$ has a constant tangential component along edge 1. Moreover, since ξ vanishes along edge 2, $\boldsymbol{\nabla}\xi$ is normal to edge 2, η vanishes along edge 3, and $\boldsymbol{\nabla}\eta$ is normal to edge 3, $\boldsymbol{v}_1$ has no tangential components along these edges. Similar observations apply to $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$. Thus, tangential continuity is preserved across inter-element boundaries, but normal component is not. Fig. 2.22a and Fig. 2.22b illustrate the variation of the basis functions $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ for the edges e_1 and e_2, respectively. These vector plots make it easy to visualize how the constant tangential component of a specific shape function is maintained only along the associated edge.

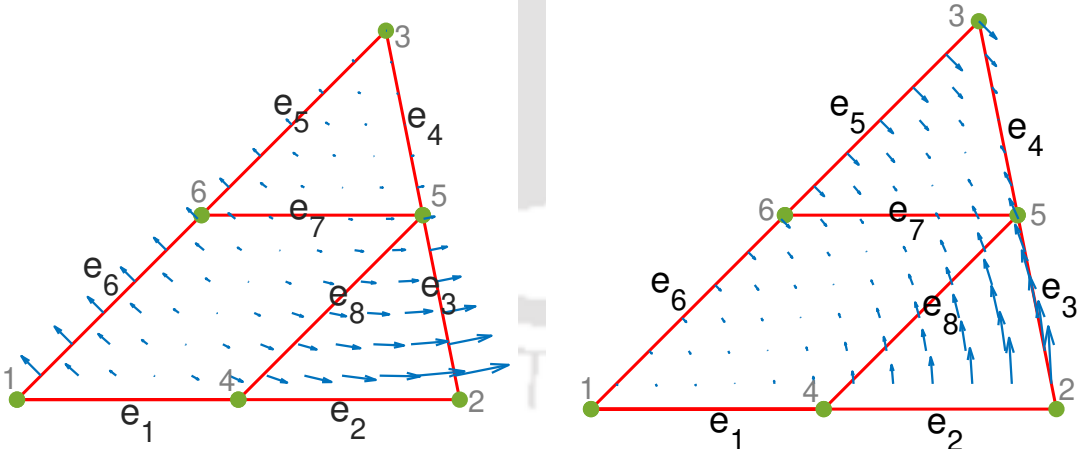
This same analysis applies to understand the behaviour of shape functions of higher order triangular edge elements. Because, the higher-order triangular shape functions are constructed by multiplying the interpolatory polynomial of higher order at the respective



(a) Variation of $\boldsymbol{v}_1$ for the edge e_1. **(b)** Variation of $\boldsymbol{v}_2$ for the edge e_2.

Figure 2.22: Vector plots of edge shape functions of 3-edge triangular element.

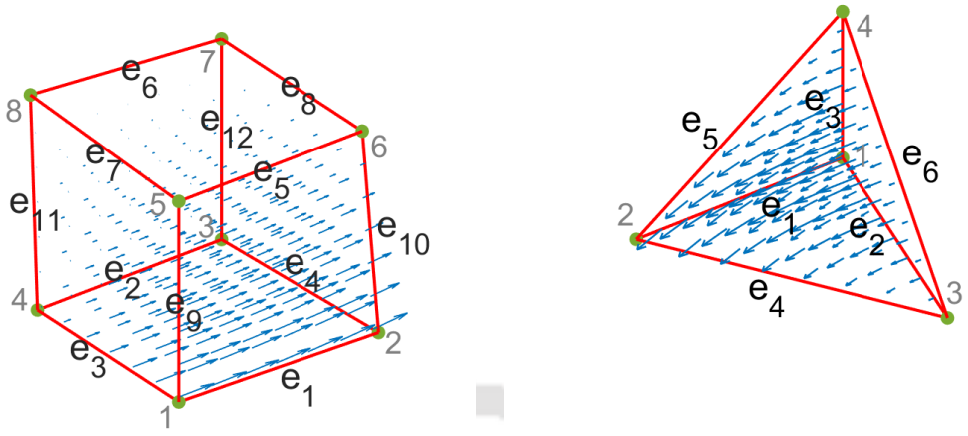
vertex with the lower-order triangle shape functions. For example, for an 8-edge triangle, the shape functions for edges e_2 and e_3 are $v_2 = l_2(4\eta - 1)(\xi\boldsymbol{\nabla}\eta - \eta\boldsymbol{\nabla}\xi)$ and $v_3 = l_3(4\eta - 1)(\eta\boldsymbol{\nabla}\alpha - \alpha\boldsymbol{\nabla}\eta)$, respectively. Fig. 2.23a and Fig. 2.23b show the variation of these shape functions for edges 2 and 3, respectively, illustrating the behaviour of $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$ at those edges.



(a) Variation of $\boldsymbol{v}_2$ for the edge e_2. **(b)** Variation of $\boldsymbol{v}_3$ for the edge e_3.

Figure 2.23: Vector plots of edge shape functions of 8-edge triangular element.

Similarly, we can explore the behaviour of edge shape functions for 3D edge elements, particularly for hexahedral and tetrahedral elements. The formulation used to derive the shape functions of 2D edge elements can be extended to 3D edge elements in a straightforward manner. These shape functions are constructed not only to guarantee tangential continuity across the edges, but also to guarantee the tangential continuity across the surfaces of the elements.



(a) For 12-edge hexahedral element, variation of $\boldsymbol{v}_1$ for the edge e_1.

(b) For 6-edge tetrahedra, variation of $\boldsymbol{v}_1$ for the edge e_1.

Figure 2.24: Vector plots of edge shape functions of 3D edge elements.

Let us consider the 12-edge hexahedral element which has 12 edges (see Fig. 2.24a). The edge shape functions of the twelve edges in natural coordinates ξ, η, and ζ are given as (see [1])

$$\boldsymbol{v}_1 = \frac{l_1}{8}(1-\eta)(1-\zeta)\boldsymbol{\nabla}\xi, \quad \boldsymbol{v}_2 = \frac{l_2}{8}(1+\eta)(1-\zeta)\boldsymbol{\nabla}\xi, \quad \boldsymbol{v}_3 = \frac{l_3}{8}(1-\xi)(1-\zeta)\boldsymbol{\nabla}\eta,$$
$$\boldsymbol{v}_4 = \frac{l_4}{8}(1+\xi)(1-\zeta)\boldsymbol{\nabla}\eta, \quad \boldsymbol{v}_5 = \frac{l_5}{8}(1-\eta)(1+\zeta)\boldsymbol{\nabla}\xi, \quad \boldsymbol{v}_6 = \frac{l_6}{8}(1+\eta)(1+\zeta)\boldsymbol{\nabla}\xi,$$
$$\boldsymbol{v}_7 = \frac{l_7}{8}(1-\xi)(1+\zeta)\boldsymbol{\nabla}\eta, \quad \boldsymbol{v}_8 = \frac{l_8}{8}(1+\xi)(1+\zeta)\boldsymbol{\nabla}\eta, \quad \boldsymbol{v}_9 = \frac{l_9}{8}(1-\xi)(1-\eta)\boldsymbol{\nabla}\zeta,$$
$$\boldsymbol{v}_{10} = \frac{l_{10}}{8}(1+\xi)(1-\eta)\boldsymbol{\nabla}\zeta, \quad \boldsymbol{v}_{11} = \frac{l_{11}}{8}(1-\xi)(1+\eta)\boldsymbol{\nabla}\zeta, \quad \text{and}$$
$$\boldsymbol{v}_{12} = \frac{l_{12}}{8}(1+\xi)(1+\eta)\boldsymbol{\nabla}\zeta,$$

where l_1, l_2, l_3, ... and l_{12} are lengths of the twelve edges of the element. To illustrate, let us observe the behaviour of $\boldsymbol{v}_1$. This vector basis function, defined by $\boldsymbol{\nabla}\xi$, has nonzero tangential component only along the edge 1, which is parallel to the ξ-axis. This is because ξ varies linearly along the edge 1 (which is parallel to the ξ-axis) and remains constant along the other edges. This means $\boldsymbol{v}_1$ maintains tangential continuity along the edge 1, as well as on the surfaces contained by edge 1. This behaviour can be illustrated from the vector plot of $\boldsymbol{v}_1$ as shown in Fig. 2.24a. Therefore, the edge shape functions constructed possess all the necessary properties that guarantee the tangential continuity across the element edges as well as across element surfaces.

For a 6-edge tetrahedral element which has 6 edges as shown in Fig. 2.24b. From the nodal shape functions of 4-node tetrahedra $N_1 = 1-\xi-\eta-\zeta$, $N_2 = \xi$, $N_3 = \zeta$ and $N_4 = \eta$ edge shape functions are derived. Then, the six edge shape functions expressed

in natural coordinates ξ, η and ζ are given as (see [1])

$$\boldsymbol{v}_1 = l_1(1-\eta-\zeta)\boldsymbol{\nabla}\xi + l_1\xi(\boldsymbol{\nabla}\eta + \boldsymbol{\nabla}\zeta), \qquad \boldsymbol{v}_2 = l_2(1-\xi-\eta)\boldsymbol{\nabla}\zeta + l_2\zeta(\boldsymbol{\nabla}\xi + \boldsymbol{\nabla}\eta),$$
$$\boldsymbol{v}_3 = l_3(1-\xi-\zeta)\boldsymbol{\nabla}\eta + l_3\eta(\boldsymbol{\nabla}\xi + \boldsymbol{\nabla}\zeta), \quad \boldsymbol{v}_4 = l_4(\xi\boldsymbol{\nabla}\zeta - \zeta\boldsymbol{\nabla}\xi), \quad \boldsymbol{v}_5 = l_5(\eta\boldsymbol{\nabla}\xi - \xi\boldsymbol{\nabla}\eta),$$
$$\text{and } \boldsymbol{v}_6 = l_6(\zeta\boldsymbol{\nabla}\eta - \eta\boldsymbol{\nabla}\zeta),$$

where l_1, l_2, ..., l_6 are the lengths of the six edges of the tetrahedral element. Similar to observation of shape functions for the triangular element, we can analyze the shape functions for tetrahedral element. The derivation of the shape functions for the tetrahedral element follows the same pattern as for triangular vector basis functions. Consider the shape function of edge 1, i.e., $\boldsymbol{v}_1$, which is derived from the nodal shape functions of node 1 and node 2. From the analysis of the lower-order triangle, by considering unit vector $\boldsymbol{n}_1$ pointing from node 1 and node 2, we can observe that $\boldsymbol{v}_1$ has only tangential component along edge 1. Moreover, since N_1 vanishes along edges e_4, e_5, and e_6 and N_2 vanishes along edges e_2, e_3 and e_6, $\boldsymbol{v}_1$ has no tangential component along these five edges. Furthermore, because N_1 vanishes on the element facet defined by (2,3,4) and N_2 vanishes on the element facet defined by (1,3,4), $\boldsymbol{v}_1$ has no tangential component on either of these faces. Its tangnetial component appears only on the element facets that contain edge e_1. This behavior can be easily visualized from the vector plot of $\boldsymbol{v}_1$, as shown in Fig. 2.24b.

2.8 Summary

We have presented a very useful conversion technique, which transform the nodal connectivities into edge connectivities for both two-dimensional and three-dimensional finite elements. Also, this conversion algorithm generates other necessary data structures in edge formulations like direction information of the edges, connecting nodes of a particular edge, associated edges of a particular node and respective other node of those edges. This algorithm converts 4-node quadrilateral into 4-edge quadrilateral, 3-node triangle into 3-edge triangle, 6-node triangle into 8-edge triangle, and 9-node quadrilateral into 12-edge quadrilateral in the case of two-dimensional finite elements. Whereas, for three-dimensional finite elements, this algorithm generates 6-edge tetrahedral elements from 4-node tetrahedral elements, and 12-edge hexahedral elements from 8-node hexahedral elements. Also, for each transformed edge element, detailed explantation of conversion algorithm is provided with relevant examples.

Chapter 3

Eigenvalue analysis and comparative performance of Nodal-to-Edge finite elements over conventional Nodal elements[†]

In electromagnetics, eigenvalue problems that are often encountered include those of cavity resonance and wave propagation in both closed and open structures such as waveguides, cavity resonators, and optical fibres. In these problems, one is interested in finding the resonant frequencies corresponding to eigenvalues and the associated resonant or propagation modes corresponding to eigenvectors.

To assess the numerical performance of the nodal-to-edge elements introduced in the previous chapter, several benchmark problems both in two-dimensions and three-dimensions have been thoroughly studied in this chapter. Eigen analysis is performed using the Eq. (1.5) for finding the square of the eigenvalues k_0^2. Additionally, the performance of the proposed elements is compared with that of the conventional nodal elements through a series of examples. These problems have been solved for meshes with successive refinement levels for both nodal and edge elements. The proposed two-dimensional nodal-to-edge elements consists of four edge quadrilateral (EQ4), three edge triangle (ET3), twelve edge quadrilateral (ET12), and eight edge triangle (ET8). Where as nodal elements include four node quadrilateral (Q4), nine node quadrilateral (Q9), and six node triangle (T6). In three-dimensional nodal elements, we have considered

[†]D. Kamireddy, S. Madhukar Chavan, and A. Nandy, "Comparative Performance of Novel Nodal-to-Edge finite elements over Conventional Nodal element for Electromagnetic Analysis", *Journal of Electromagnetic Waves and Applications* 37(1), 131−161 (2023). https://doi.org/10.1080/09205071.2022.2117102

ten node tetrahedra (TetN10), and eight node hexahedra (HexN8), while the transformed edge elements consist of six edge tetrahedra (TetE6), and twelve edge hexahedra (HexE12).

For all the problems discussed in this chapter, we have assumed $\epsilon_r = \mu_r = 1.0$. In the following examples, we have compared edge and nodal elements with respect to a certain error percentage with analytical benchmark values. In that bar-diagram comparison, each bar represents the minimum no. of required FDOF (free degrees of freedom) for that particular element to attain an error percentage less than the predefined scale.

There are black crosses in some bar diagrams which signify that it is not possible to generate less than 10% error with that element with available computational resources. We have mentioned best possible result for those cases. Such situations mostly occur with nodal element which further establish the better performance of edge elements. Also, the first non-zero eigenvalues of all the elements appeared after stating the number of zeros generated at the machine precision level. These zeros indicate the approximation of null space. The remaining chapter is arranged in the following manner: In section 3.1, the relative performance of two-dimensional nodal and transformed edge elements has been compared using several numerical examples. Thereafter, the effect of mesh distortion on edge elements along with the computational cost in numerical analysis has also been presented. And in section 3.2, we delve into the comprehensive details of the comparative performance study conducted between three-dimensional nodal and edge elements, accompanied by a series of numerical examples.

3.1 Numerical examples of eigenvalue problems in two-dimensions

In this section, we present the results of eigenvalue analysis conducted for two-dimensional domains with all possible complexities. These domains encompass shapes with regular boundaries, curved boundaries, sharp corners, non-convex shapes, and domains with non-homogeneous materials. In our simulations, we have modelled the computational domain using two-dimensional nodal finite elements, including Q4, T8, and Q8, as well as two-dimensional edge finite elements such as EQ4, ET3, EQ12 and ET8.

3.1.1 Square domain

A square domain with a side length of π is considered. The square domain's sides or boundaries are all perfectly conducting. Analysis data of different nodal and edge elements, like number of free degrees of freedom (FDOF) i.e. total number of equations, is presented in Table 3.1.

The square of eigenvalues for each type of element is given in Table 3.2 along with analytical results reported in [31]. All of these elements have produced correct eigenvalues along with the correct multiplicities.For all the elements the first non zero eigenvalue

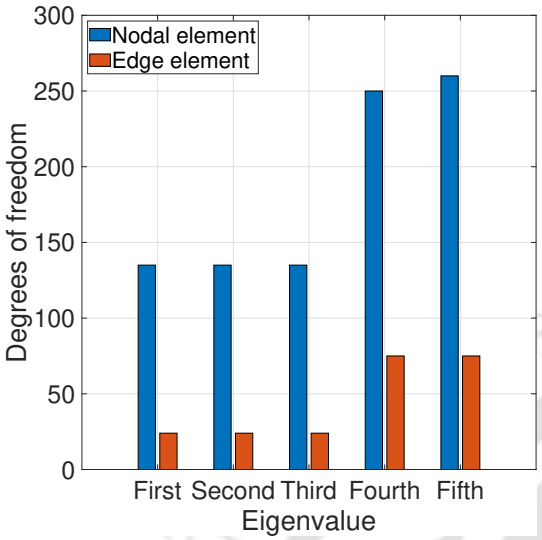
Table 3.1: Analysis data of different nodal and edge elements for the square domain problem.

Nodal element			Edge element		
Element Type	No. of elements	No. of equations/FDOF	Element Type	No. of elements	No. of equations/FDOF
Q4	64	231	EQ4	196	364
T6	98	663	ET8	112	530
Q9	64	855	EQ12	100	760

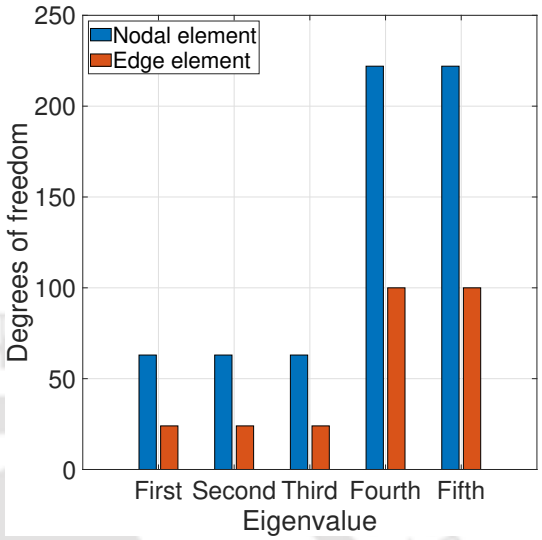
Table 3.2: k_0^2 on the square domain for different elements.

	Nodal element			Edge element		
Analytical	Q4	Q9	T6	EQ4	EQ12	ET8
1	1.012916	0.999843	0.999919	1.004203	1.000013	0.999990
1	1.012916	0.999843	0.999919	1.004203	1.000013	1.000015
2	2.025832	1.999264	2.000128	2.008407	2.000027	2.000150
4	4.209548	3.998554	4.000663	4.067583	4.000850	3.999844
4	4.209548	3.998554	4.001671	4.067583	4.000850	4.000469
5	5.222465	4.997901	5.003591	5.071786	5.000863	5.000285
5	5.222465	4.997901	5.007171	5.071786	5.000863	5.000285
8	8.419096	7.996779	8.033095	8.135166	8.001700	8.008889
9	10.080293	9.007072	9.024805	9.344778	9.009435	8.998040
9	10.080293	9.007072	9.024956	9.344778	9.009435	9.005165
10	11.093208	10.002567	10.038396	10.348982	10.009449	10.003717
10	11.093208	10.012254	10.048274	10.348982	10.009449	10.011721
13	14.289839	13.015767	13.088936	13.412361	13.010285	13.014842
13	14.289839	13.015767	13.179304	13.412361	13.010285	13.048523
16	19.453669	16.064853	16.145861	17.100354	16.051302	15.989497
16	19.453669	16.076531	16.158926	17.100354	16.051302	16.025185
Number of computed zeros						
-	77	245	221	45	121	65

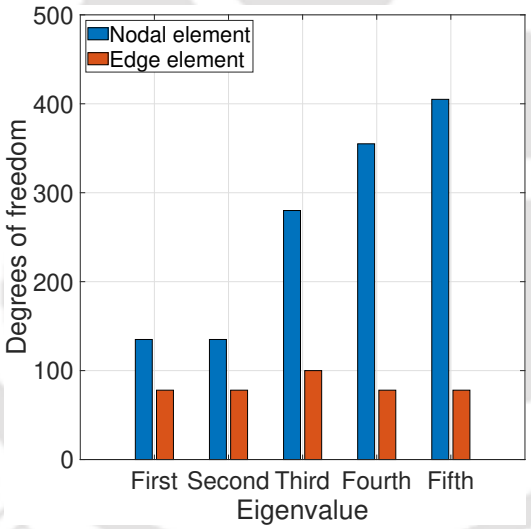
has occurred after a certain number of zero eigen values at the machine precision level which signifies the approximation of null space. We have noticed a significant difference between nodal and edge elements in the mesh convergence analysis. With edge elements, we have found a far better coarse mesh accuracy. This fact is presented in bar diagrams in Fig. 3.1. We have compared Q9 and EQ12 in Fig. 3.1a on a scale of less than 1% error. Q4 and EQ4 elements are compared for less than 6% error in Fig. 3.1b. A scale of less than 0.2% error is chosen in Fig. 3.1c to compare T6 and ET8 elements. From Fig. 3.1 we can see that we can attain the required level of accuracy for all five eigen values with a coarser edge element mesh than the required nodal element mesh for both quadrilateral and triangular elements.



(a) Minimum number of FDOF required for less than 1% error (Q9 vs EQ12).



(b) Minimum number of FDOF required for less than 6% error (Q4 vs EQ4).



(c) Minimum number of FDOF required for less than 0.2% error (T6 vs ET8).

Figure 3.1: Comparative study of numerical performance of nodal elements with edge elements in predicting eigenvalues for square domain.

3.1.2 Circular shape domain

Here, the domain is a circle of radius unity and Fig. 3.2 shows the discretized domain. The circular domain has a perfectly conducting boundary. No. of elements for different meshes of various element types are given in Table 3.3. The results are found with these meshes and they are closely matching with the benchmark values presented in [113, 114]. For all the elements squared of the obtained eigenvalues are listed in Table 3.4. The required degrees of freedom required to get less than 7% error for various elements has been presented in Fig. 3.3.

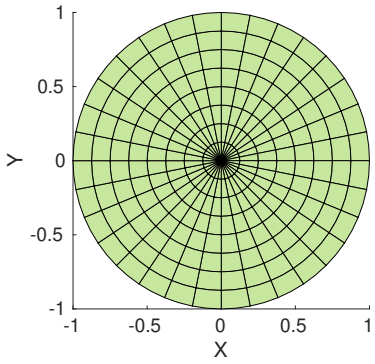


Figure 3.2: Discretized circular domain.

Table 3.3: Analysis data of different nodal and edge elements for the circular shape domain problem.

Nodal element			Edge element		
Element Type	No. of elements	No. of equations/ FDOF	Element Type	No. of elements	No. of equations/ FDOF
Q9 & T6	256	2979	EQ4 & ET3 EQ12 & ET8	1600 600	3160 4720

Table 3.4: k_0^2 on the circular domain for different elements (bracketed values show the multiplicity).

	Nodal element	Edge element	
Benchmark	Q9 & T6	EQ4 & ET3	EQ12 & ET8
3.391122 (2)	3.388867	3.410866	3.380563
9.329970 (2)	9.318499	9.443712	9.329356
14.680392 (1)	14.668347	14.768032	14.756723
17.652602 (2)	17.609686	18.043890	17.662206
28.275806 (2)	28.161350	28.586164	28.262108
28.419561 (2)	28.376474	29.300835	28.343555
41.158640 (2)	40.896578	43.394573	41.404578
44.970436 (2)	44.858738	45.351002	44.973322
49.224256 (1)	49.098151	49.598925	49.656871
56.272502 (2)	55.747995	60.581974	56.956837
64.240225 (2)	64.023638	65.163560	64.271251
Number of computed zeros			
-	992	401	450

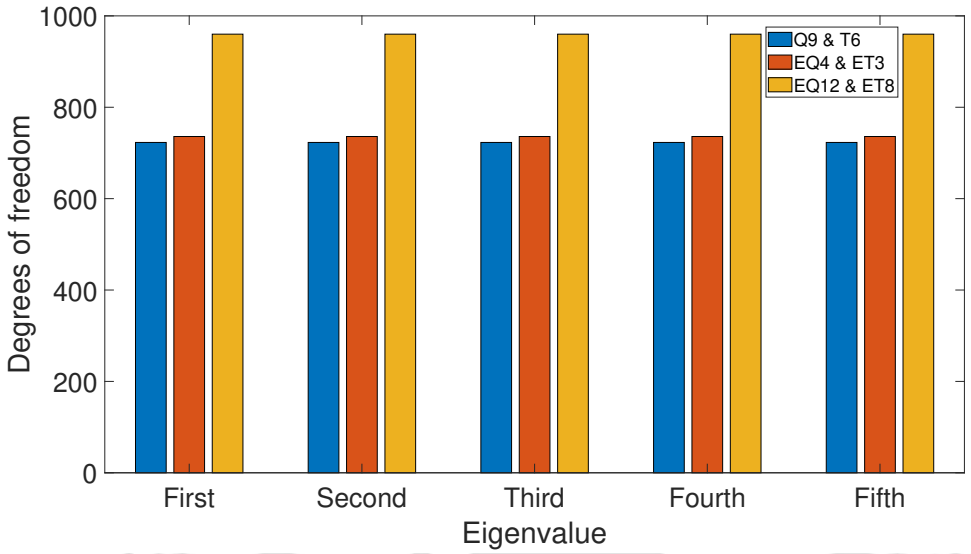


Figure 3.3: Minimum number of FDOF required for less than 7% error in predicting eigen values for the circular domain.

3.1.3 L-shaped domain

The L-shaped domain is obtained by deleting one quadrant from the square domain of side π which has been considered in the previous example (Section 3.1.1). Here, the domain is a non-convex domain with sharp corners. In numerical analysis, conducting boundary conditions are applied to all the boundaries of domain. The discretized L shape domain with Q9 elements is shown in Fig. 3.4. The L-shaped domain is discretized with different mesh sizes for different elements, which is given in Table 3.5. Square of the eigenvalues are listed in Table 3.6 for all the elements along with standard values reported in [31].

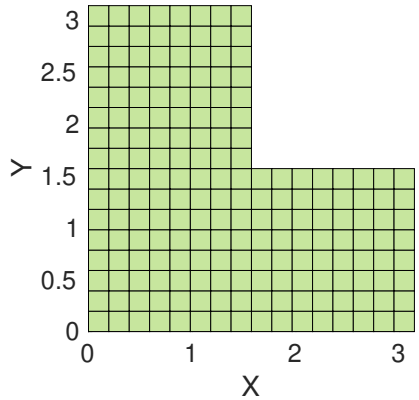


Figure 3.4: Discretized L shape domain.

As nodal elements can not capture the singular eigen value (0.591790), we have compared the accuracy of squared of the obtained eigenvalues in Fig. 3.5 from the

Table 3.5: Analysis data of different nodal and edge elements for the L shape domain problem.

Nodal element			Edge element		
Element Type	No. of elements	No. of equations/FDOF	Element Type	No. of elements	No. of equations/FDOF
Q4	768	2481	EQ4	432	805
T6	384	2481	ET8	384	1856
Q9	192	2481	EQ12	192	1457

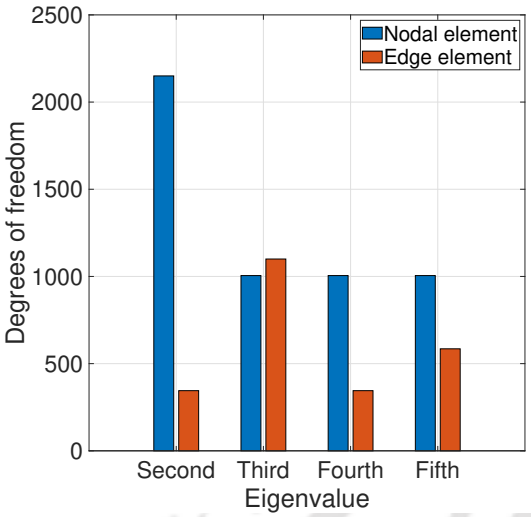
Table 3.6: k_0^2 on the L-shaped domain for different elements.

	Nodal element			Edge element		
Benchmark	Q4	Q9	T6	EQ4	EQ12	ET8
0.591790	-	-	-	0.596170	0.597191	0.596538
1.432320	1.479654	1.507641	1.481174	1.434491	1.432148	1.432253
-	1.620830	2.277630	1.623752	-	-	-
4.005540	4.012869	3.997995	3.998716	3.781632	3.815079	3.999995
4.005540	4.012869	3.998150	3.998852	4.022899	4.000132	4.000019
4.613200	4.649897	4.645948	4.636011	4.418905	4.427041	4.616022
5.067330	5.577478	5.960861	5.561194	4.946604	4.939201	5.090472
7.955130	8.025738	7.990559	7.995340	7.821276	7.806319	8.000462
8.647370	9.404155	9.563233	9.336718	8.731623	8.649600	8.671775
9.481660	9.597178	9.838923	9.528452	9.575278	9.457106	9.460954
11.426100	12.380478	13.563361	12.304280	11.421820	11.349739	11.534996
14.448600	14.808709	14.771906	14.700390	14.611556	14.451696	14.542718
16.086200	16.206662	15.974127	15.984640	16.368790	16.008200	16.000365
Number of computed zeros						
-	828	827	827	64	39	22

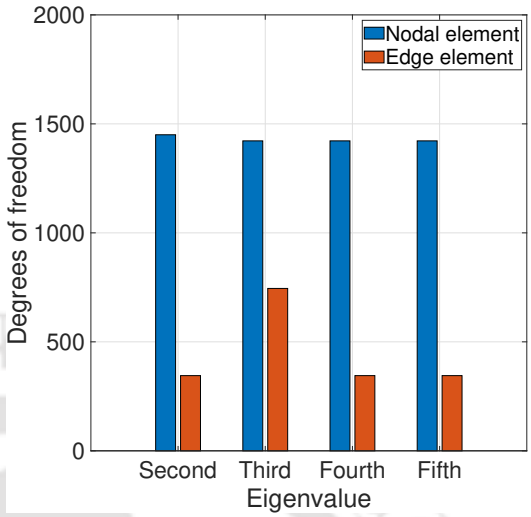
second eigen frequency. All the nodal elements are not able to capture the singular eigen value, as well as all of them, generate one spurious eigen value, whereas all the edge elements are able to predict the singular eigen value properly and they do not generate the spurious eigen value. We have found better coarse mesh accuracy with edge elements which is depicted in bar diagrams in Fig. 3.5. We have compared Q9 and EQ12 elements in Fig. 3.5a on a scale of less than 6% error. Q4 and EQ4 elements are compared on the scale of 6% error in Fig. 3.5b whereas T6 and ET8 elements are compared for less than 4% error in Fig. 3.5c.

3.1.4 Cracked circular domain

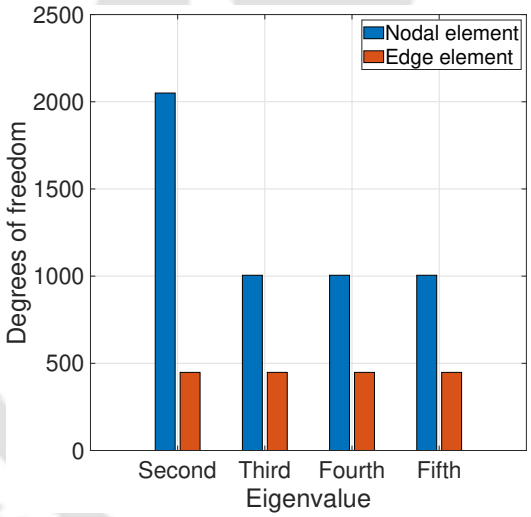
Here, the domain is a circle of radius unity but it is with a crack as shown in Fig. 3.6. All the boundaries of the computational domain are subjected to perfectly conducting boundary conditions.



(a) Minimum number of FDOF required for less than 6% error (Q9 vs EQ12).



(b) Minimum number of FDOF required for less than 6% error (Q4 vs EQ4).



(c) Minimum number of FDOF required for less than 5% error (T6 vs ET8).

Figure 3.5: Comparative study of numerical performance of nodal elements with edge elements in predicting eigenvalues for L shape domain.

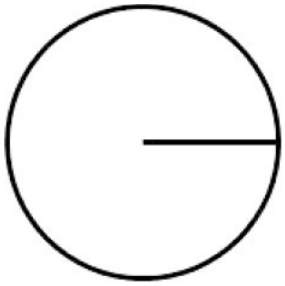


Figure 3.6: Cracked Circular domain.

Table 3.7: Analysis data of different nodal and edge elements for the cracked circular domain problem.

Nodal element			Edge element		
Element Type	No. of elements	No. of equations/ FDOF	Element Type	No. of elements	No. of equations/ FDOF
Q9 & T6	256	3030	EQ4 & ET3 EQ12 & ET8	384 400	723 3161

Table 3.8: k_0^2 for the circular domain with crack for different elements.

Analytical Benchmark	Nodal element	Edge element	
	Q9 & T6	EQ4 & ET3	EQ12 & ET8
1.358390	-	1.328769	1.243789
3.391122	3.732977	3.429584	3.368877
6.059858	6.059539	6.155568	6.051724
	7.907027	-	-
9.329970	9.315916	9.540422	9.329118
13.195056	13.171354	13.594326	13.201251
14.680392	14.664092	14.941521	14.763221
17.652602	17.601145	18.337851	17.662210
21.196816	22.594381	21.136960	20.034173
22.681406	28.140650	23.799963	22.709148
28.275806	28.589967	28.986891	28.067093
Number of computed zeros			
-	1009	40	8

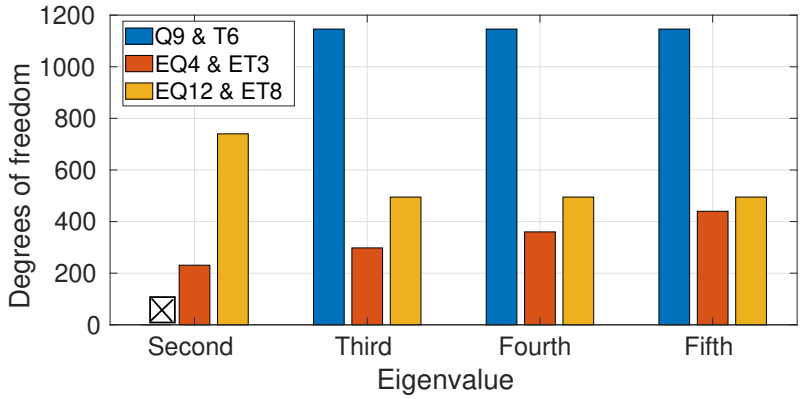


Figure 3.7: Minimum number of FDOF required for less than 4% error in predicting eigen values for the cracked circular domain.

The discretization of meshed domain with different types of elements is shown in Table 3.7. The results are found with these meshes and are listed in Table 3.8 along with the benchmark values from [113, 114]. Numerical results are compared and they

are closely matching with the benchmark values. As nodal elements can not capture the singular eigen value (1.358390), we have compared the accuracy of the obtained eigenvalues in Fig. 3.7 from the second eigen frequency. The degrees of freedom required to get an error less than 4% is shown in Fig. 3.7. With Q9 & T6 elements, we have obtained error of 10.1% with 3030 FDOF for second eigen value.

3.1.5 Curved L-shaped domain

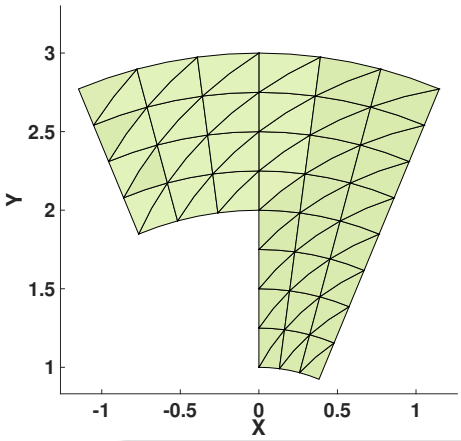


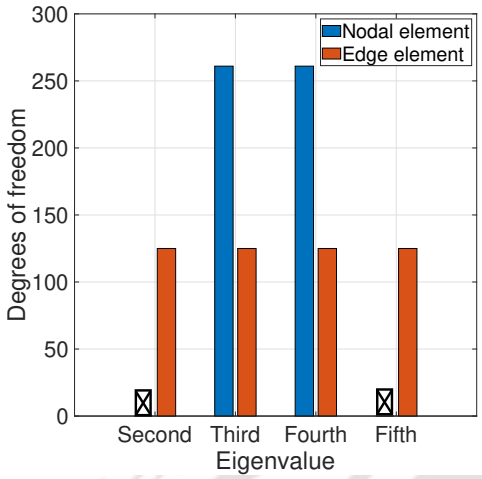
Figure 3.8: Discretized curved L shape domain.

Table 3.9: Analysis data of different nodal and edge elements for the curved L shape domain problem.

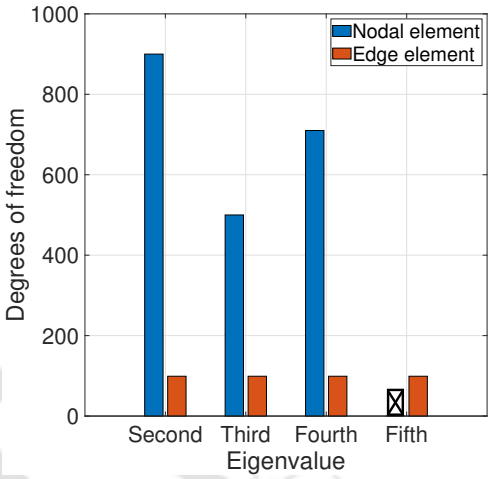
Nodal element			Edge element		
Element Type	No. of elements	No. of equations/FDOF	Element Type	No. of elements	No. of equations/FDOF
Q4	270	909	EQ4	200	480
T6	96	657	ET8	90	418
Q9	48	657	EQ12	84	790

This example is taken from [115]. Here, the domain has three straight and three circular sides of radii 1, 2, and 3, and Fig. 3.8 shows the discretized domain. All the sides of the domain are perfectly conducting. This is one of the challenging problem as the domain is curved, non-convex along with sharp corner. Details of the meshed domain for different nodal and edge elements are given in Table 3.9.

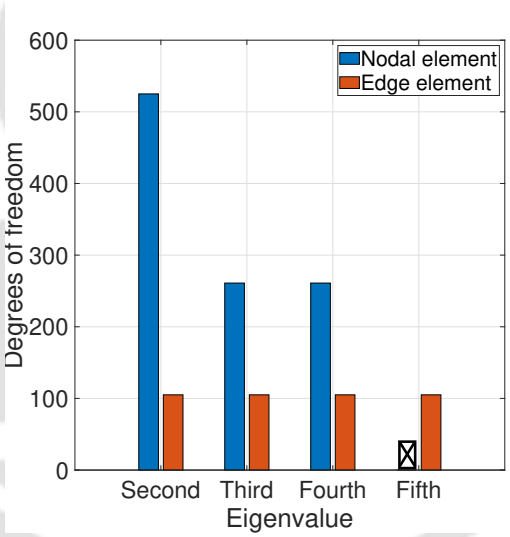
For all the elements, squared of the obtained eigenvalues are listed in Table 3.10 along with the benchmark values from [115]. All the nodal elements are not able to capture the singular eigen value (1.818571) (due to the presence of a sharp corner) as well as all of them generate one spurious eigen value. Whereas all the edge elements are able to predict the singular eigen value properly and they do not generate the spurious eigen



(a) Minimum number of FDOF required for less than 2.5% error (Q9 vs EQ12).



(b) Minimum number of FDOF required for less than 6.5% error (Q4 vs EQ4).



(c) Minimum number of FDOF required for less than 8% error (T6 vs ET8).

Figure 3.9: Comparative study of numerical performance of nodal elements with edge elements in predicting eigenvalues for curved L shape domain.

value. Therefore we have compared accuracy from the second eigenvalue in Fig. 3.9. We have found better coarse mesh accuracy with edge elements which is depicted in bar diagrams in Fig. 3.9. We have compared Q9 and EQ12 elements in Fig. 3.9a on a scale of less than 2.5% error. A scale of 6.5% error is chosen in Fig. 3.9b to compare Q4 and EQ4. T6 and ET8 elements are compared for less than 8% error in Fig. 3.9c. With 657 FDOF of Q9 element, we have obtained 10.4% error for the second eigen value and 21.7% error for the fifth eigen value. An error of 11.3% is obtained for Q4 element with 909 FDOF for the fifth eigen value. With T6 element for the fifth eigen value we get an error of 10.6% with 657 FDOF.

Table 3.10: k_0^2 on the curved L-shaped domain for different elements.

	Nodal element			Edge element		
Benchmark	Q4	Q9	T6	EQ4	EQ12	ET8
1.818571	-	-	-	1.809583	1.814099	1.804651
3.490576	3.689097	3.854593	3.718116	3.505945	3.490535	3.489638
-	5.033585	6.889177	5.070643	-	-	-
10.065602	10.153471	10.047738	10.056494	10.198870	10.066721	10.071826
10.111886	10.284415	10.252722	10.194525	10.241690	10.116060	10.109109
12.435537	13.845497	15.141449	13.759140	12.554637	12.427678	12.420649
Number of computed zeros						
-	303	219	219	184	173	149

3.1.6 Inhomogeneous L shape domain

In this example, the efficacy of the edge elements for the inhomogeneous domain becomes evident. This example is considered from [31]. Fig. 3.10 represents the domain where relative permittivities are 1 and 5 for grey and the light green regions respectively. Here, the walls of the domain are assumed to be perfectly conducting. Discretization details of the domain for various elements are presented in Table 3.11. For all the elements squared of the obtained eigenvalues are listed in Table 3.12. All the nodal elements are not able to capture the singular eigen value (0.175980) as well as all of them generate one spurious eigen value.

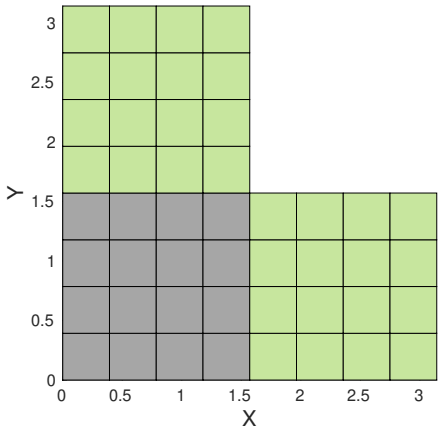


Figure 3.10: Inhomogeneous L shape domain.

Whereas all the edge elements are able to predict the singular eigen value properly and they do not generate the spurious eigen value, thus we have compared the accuracy of elements from the second eigen value in Fig. 3.11.We have found better coarse mesh accuracy with edge elements which is depicted in bar diagrams in Fig. 3.11. We have compared Q9 and EQ12 elements in Fig. 3.11a on a scale of less than 6% error. Q4 and

Table 3.11: Analysis data of different nodal and edge elements for the inhomogeneous L shape domain problem.

Nodal element			Edge element		
Element Type	No. of elements	No. of equations/FDOF	Element Type	No. of elements	No. of equations/FDOF
Q4	768	2481	EQ4	768	1457
T6	384	2481	ET8	384	1856
Q9	192	2481	EQ12	192	1457

Table 3.12: k_0^2 on the inhomogeneous L shape domain for different elements.

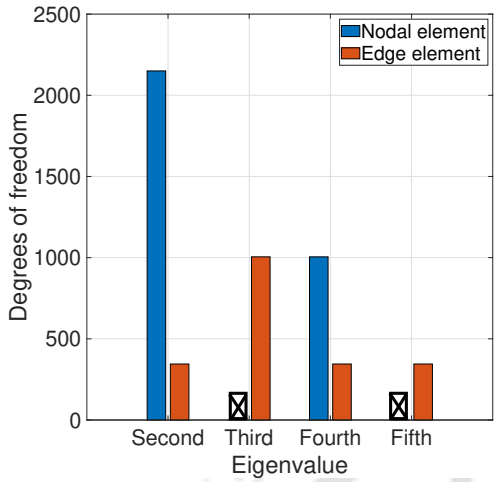
	Nodal element			Edge element		
Benchmark	Q4	Q9	T6	EQ4	EQ12	ET8
0.175980	-	-	-	0.176192	0.176090	0.176085
0.398080	0.410869	0.418399	0.411088	0.397985	0.397393	0.397395
-	0.410993	0.526242	0.411234	-	-	-
0.964840	0.973806	0.973085	0.970759	0.872179	0.853464	0.966204
0.978740	0.998372	1.003343	0.994894	0.976496	0.973764	0.980699
1.524310	1.785034	1.783645	1.775821	1.529238	1.521521	1.522974
1.765930	1.816265	1.973428	1.807250	1.768005	1.758150	1.761401
2.274180	2.306054	2.292383	2.293151	2.252808	2.233477	2.293116
2.389530	2.563009	2.694188	2.544549	2.395908	2.381229	2.412659
3.394090	3.428865	3.380034	3.382655	3.424851	3.385157	3.384739
3.397400	3.433245	3.384257	3.386683	3.427972	3.388137	3.388719
3.646940	3.695923	3.664321	3.663456	3.664173	3.630283	3.641584
3.664270	3.906853	4.096015	3.873663	3.686256	3.655079	3.658646
Number of computed zeros						
-	827	827	827	19	56	15

EQ4 elements are compared on a scale of 10% error in Fig. 3.11b whereas T6 and ET8 elements are compared for less than 5% error in Fig. 3.11c.

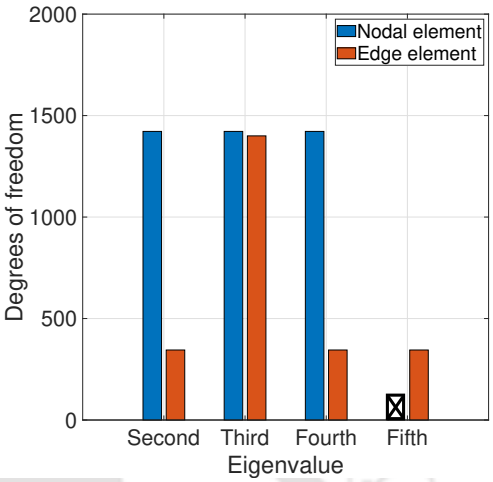
With Q9 elements for 2481 FDOF there is 17% error for the fifth eigen value. For EQ12 elements we obtained error of 11.8% with 1457 FDOF for the third eigen value while for Q4 element we get 17.1% error for 2481 FDOF with the fifth eigen value. Error of 16.5% is obtained with T6 element for 2481 FDOF for the fifth eigen value.

3.1.7 Comparative study of mesh convergence analysis with L_2-norm error between nodal and edge elements

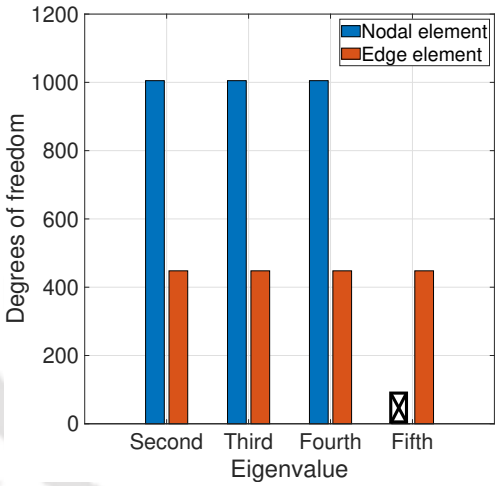
In this section, we have conducted a comparative analysis of numerical accuracy in terms of L_2 error norm with mesh convergence, between nodal and edge elements in 2D. For this study, we have chosen four different numerical examples such as square, curved L shape, L shape and inhomogeneous L shape geometries. For each problem,



(a) Minimum number of FDOF required for less than 6% error (Q9 vs EQ12).



(b) Minimum number of FDOF required for less than 10% error (Q4 vs EQ4).



(c) Minimum number of FDOF required for less than 5% error (T6 vs ET8).

Figure 3.11: Comparative study of numerical performance of nodal elements with edge elements in predicting eigenvalues for inhomogeneous L shape domain.

five different meshes of nodal and edge elements are used to perform the convergence analysis. Table 3.13 shows the detailed mesh analysis data for all four domains. L_2-norm error is calculated between numerical and analytical benchmark values as

$$L_2\text{-norm error} = \sqrt{\frac{\sum_{i=1}^{N}\Big[(\text{FEM value})_i - (\text{Analytical value})_i\Big]^2 \times m_i}{\sum_{i=1}^{N}\Big[(\text{Analytical value})_i\Big]^2 \times m_i}}, \tag{3.1}$$

where, N is maximum no. of non singular eigen values obtained with the coarsest mesh, m_i is multiplicity of one particular eigen value.

Table 3.13: Analysis data of different nodal and edge elements for L_2-norm error convergence analysis.

Square domain

Total no. of elements	No. of equations/ FDOF		Total no. of elements	No. of equations/ FDOF		Total no. of elements	No. of equations/ FDOF	
	Q4	EQ4		Q9	EQ12		T6	ET8
16	63	24	16	135	112	18	135	120
36	96	60	25	351	180	32	231	196
64	135	112	36	495	264	50	351	264
144	260	180	49	663	480	72	495	380
256	480	231	64	855	760	98	663	560

Curved L shape domain

	Q4	EQ4		Q9	EQ12		T6	ET8
24	98	93	27	177	84	24	177	144
60	219	163	36	261	112	36	288	261
96	717	295	48	381	264	54	432	381
168	813	367	84	501	480	72	501	448
270	909	560	108	760	657	96	680	657

L shape domain

	Q4	EQ4		Q9	EQ12		T6	ET8
192	483	345	48	657	345	96	657	448
300	962	551	75	1005	551	150	1005	710
432	1425	805	108	1425	805	216	1425	1032
588	1817	1107	147	1917	1107	294	1917	1414
768	2081	1457	192	2381	1457	384	2381	1856

Inhomogeneous L shape domain

	Q4	EQ4		Q9	EQ12		T6	ET8
192	483	345	48	657	345	96	657	448
300	962	551	75	1005	551	150	1005	710
432	1425	805	108	1425	805	216	1425	1032
588	1817	1107	147	1917	1107	294	1917	1414
768	2081	1457	192	2381	1457	384	2381	1856

L_2-norm error values of both nodal and edge elements are plotted along with free degrees of freedom (FDOF) for all the problems, as shown in Fig. 3.12. The error values are compared between nodal and edge elements for each problem. For example, in a square domain problem with 32 elements (196 equations), ET8 edge element predicts eigenvalues with 0.025% error, whereas T6 nodal element achieve 0.045% error with the

finest mesh we have chosen i.e., of 98 elements (663 equations). In the case of Q12 element, eigen frequencies are predicted with 0.26 % error by using 36 elements (180 equations). But, for the same number of nodal elements 495 equations are solved to achieve 0.23 % of error prediction. For 256 Q4 and EQ4 elements, numerical results are predicted by EQ4 elements with more accurately (2.7 % error) than Q4 elements (4.8 % error).

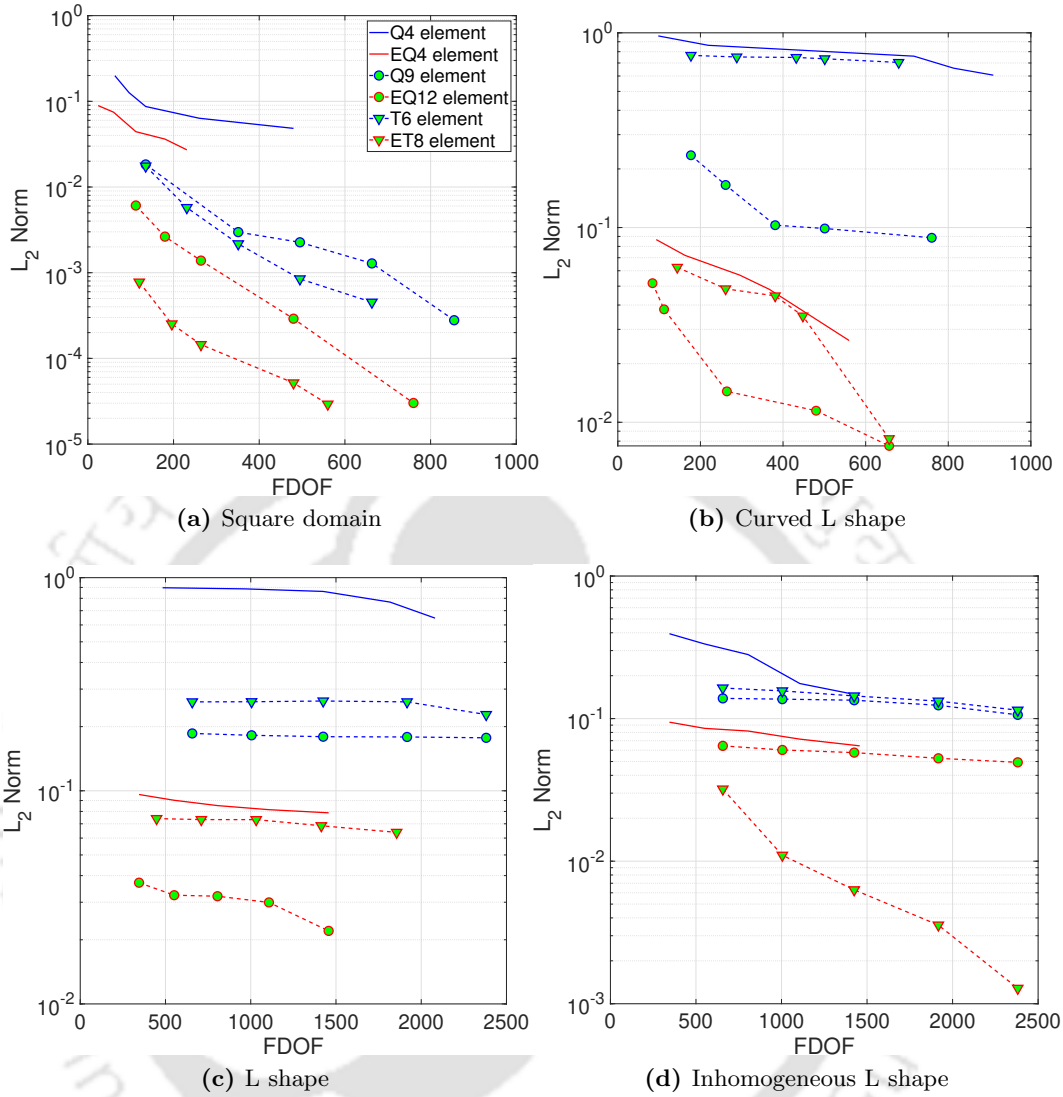
We want to assess the robustness of edge and nodal elements in predicting eigenvalues for the curved objects with sharp edges and corners. For that reason, convergence analysis has been carried out for curved L shape domain. From Fig. 3.12b, when comparing EQ12 with Q9, for almost the same degrees of freedom i.e., 264 FDOF (Q9) and 261 FDOF (EQ12), Q9 element predict eigenvalues with an error of 16.54 %, whereas EQ12 element achieve almost more than ten times more accuracy with 1.4 % error. Similarly, among ET8 and T6, ET8 is more accurate for almost equivalent FDOF whereas EQ4 has better accuracy than Q4 for meshes with close FDOF.

We have observed similar trends for other two examples, namely, homogeneous L shape domain (Fig. 3.12c) and inhomogeneous L shape domain (Fig. 3.12d). For all three types of elements, namely, linear quadrilateral, quadratic quadrilateral, and quadratic triangular elements; the edge elements have almost one order lower error norm than that of corresponding nodal elements having equivalent FDOF. Here FDOF is same as the no. of equations, hence it can be taken as an equivalent measure of computational cost.

3.1.8 Performance analysis for distorted mesh

To analyze the performance of the distorted mesh we have considered the standard curved L shape problem. To obtain the eigenvalues, the curved L shape domain is discretized using a uniform and deformed mesh made of 198 Q4 and EQ4 elements, 75 Q9 and EQ12 elements and 72 T6 and ET8 elements. The distorted meshes of different elements are shown in Figs. 3.13a, 3.13b and 3.13c. All of the elements' results are presented in Table 3.14. The results of a distorted mesh of all the elements are almost the same as the results of a uniform mesh.

Now, we want to assess the influence of mesh distortion on numerical performance of finite elements quantitatively. For this, we are defining one metric M_d, a measure of distortion in the entire domain, as $M_d = n_s \times \frac{p_s}{100}$, where p_s is % of shift of the nodes to introduce distortion, n_s is the total no. of shifted nodes in the entire domain. In our numerical experimentation, four different cases are considered which are presented in Table 3.15 along with the corresponding values of M_d. Among total no. of shifted nodes n_s, half of the nodes are shifted in radial direction and the remaining points are shifted in θ direction. We have chosen curved L shape domain for the analysis and the domain is discretized with 336 Q4 and EQ4 elements. Fig. 3.14 represents distorted meshes for all four cases mentioned in Table 3.15. For all the meshes, 628 (EQ4 elements) and 1125



(a) Square domain **(b)** Curved L shape

(c) L shape **(d)** Inhomogeneous L shape

Figure 3.12: Convergence plots in L_2-norm with increasing mesh refinement in predicting eigenvalues for four different domains.

(Q4 elements) FDOF/equations are solved for the eigen frequencies. In each case, for first four nonsingular eigenvalues, the percentage of error is calculated with benchmark values available from [115]. For each eigenvalue, the percentage of error values is plotted in Fig. 3.15 with the distortion metric parameters for both nodal and edge elements. Fig. 3.15 shows that the numerical performance of both nodal and edge elements is not affected due to mesh distortion.

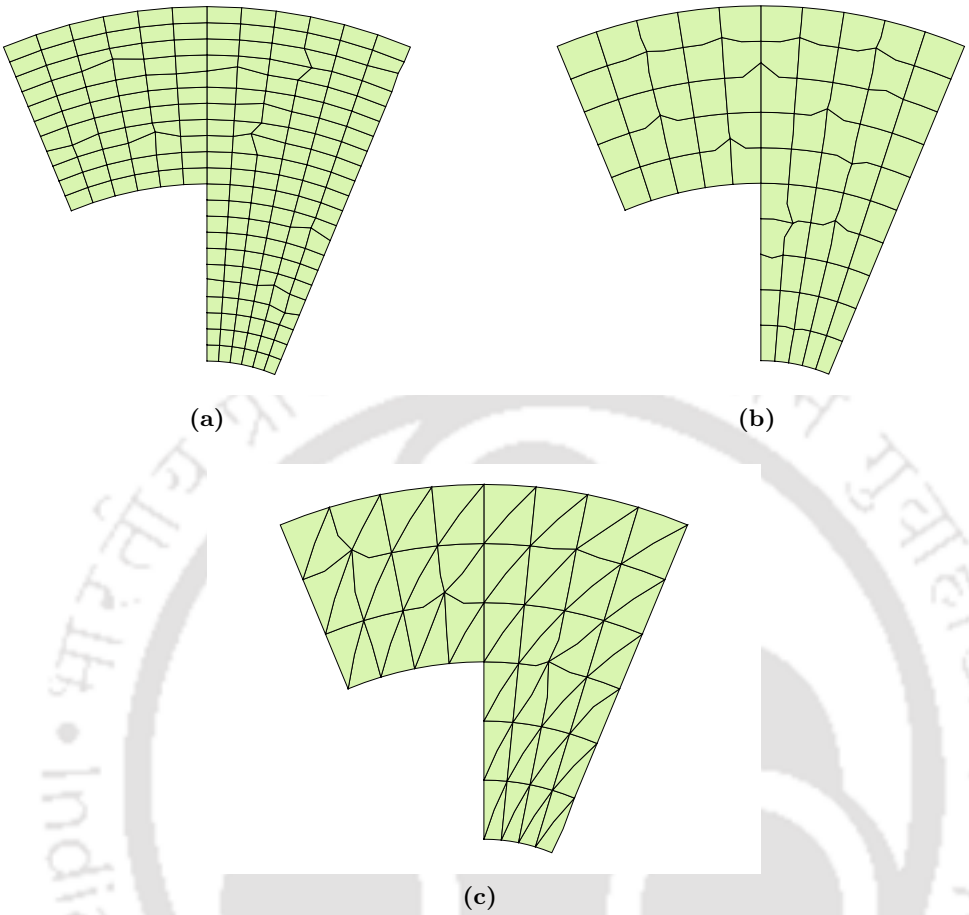


Figure 3.13: Curved L-shaped domain discretized with (a) distorted Q4 elements, (b) distorted Q9 elements and (c) distorted T6 elements.

3.1.9 Computational cost analysis

In this section, we have compared the required computational cost along with no. of required free degree of freedoms and L_2 error norm for Q9 and EQ12 elements while solving the problem of curved L shape domain. Different mesh refinements such as 27, 36, 48, 84, and 108 Q9 and EQ12 elements are used to discretize the computational domain in our numerical analysis and Table 4.2 represents respective no. of free degree of freedoms. Our computational codes are run on a Dell vostro model, using Intel Core i3 4.1 GHz processor with 4 GB RAM. We have plotted the computational time in seconds required to achieve the certain L_2-norm error along with total free degrees of freedom as shown in Fig. 4.5. We can see that with EQ12 we achieve 0.02 L_2 error norm with around 670 FDOF in 1.3 seconds, whereas with almost same no. of FDOF in Q9 we have achieved L_2 error norm as 0.17 in 3.2 seconds. Also from Fig. 4.5, we can notice that with Q9 element we can achieve L_2 norm of 0.15 in computational time

Table 3.14: k_0^2 on the curved L shape domain for normal and distorted meshes of lower order higher order of nodal and edge elements.

	Q4: 198 elements (681 FDOF)		EQ4: 198 elements (362 FDOF)	
Benchmark	Nodal element (Q4)		Edge element (EQ4)	
-	Normal	Distorted	Normal	Distorted
1.818571	-	-	1.810926	1.810926
3.490576	3.725217	3.729079	3.508771	3.508770
-	5.084991	5.085667	-	-
10.065602	10.145899	10.148647	10.141604	10.141603
10.111886	10.386313	10.407295	10.348577	10.348577
12.435537	13.897028	13.902881	12.501734	12.501734
Number of computed zeros				
-	65	65	227	227

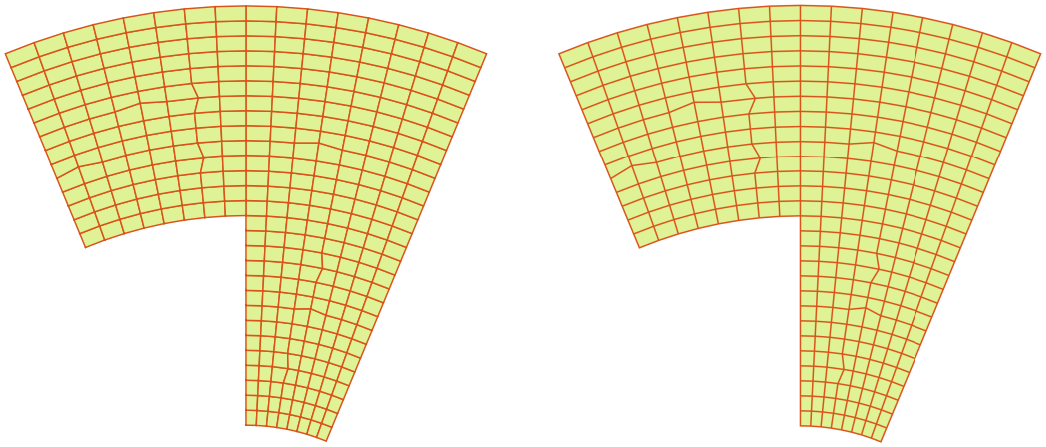
	Q9: 75 elements (1005 FDOF)		EQ12: 75 elements (560 FDOF)	
Benchmark	Nodal element (Q9)		Edge element (EQ12)	
-	Normal	Distorted	Normal	Distorted
1.818571	-	-	1.813849	1.816383
3.490576	3.798913	3.799074	3.490505	3.495839
-	6.828452	6.843561	-	-
10.065602	10.049142	10.058195	10.067992	10.088585
10.111886	10.242052	10.245292	10.113374	10.140025
12.435537	15.071954	15.085305	12.429564	12.443714
Number of computed zeros				
-	335	335	261	261

	T6: 72 elements (501 FDOF)		ET8: 72 elements (332 FDOF)	
Benchmark	Nodal element (Q9)		Edge element (EQ12)	
-	Normal	Distorted	Normal	Distorted
1.818571	-	-	1.803444	1.799088
3.490576	3.753233	3.751064	3.489667	3.484439
-	5.118928	5.109720	-	-
10.065602	10.063793	10.063864	10.070665	10.059423
10.111886	10.204527	10.212374	10.110606	10.110596
12.435537	13.806202	13.824826	12.414018	12.407989
Number of computed zeros				
-	167	167	117	117

of 4.2 seconds whereas with EQ12 with in computational time of 0.4 seconds we can achieve L_2 norm of 0.04.

Table 3.15: Mesh distortion metrics for different cases.

	% of shift (p_s)	No. of nodes shifted (n_s)	$\frac{p_s}{100}$	Distortion metric, $M_d{=}n_s \times \frac{p_s}{100}$
Case -I	20	8	0.2	1.6
Case -II	30	8	0.3	2.4
Case -III	20	16	0.2	3.2
Case -IV	30	16	0.3	4.8



(a) mesh with 1.6 distortion metric

(b) mesh with 2.4 distortion metric

(c) mesh with 3.2 distortion metric

(d) mesh with 4.8 distortion metric

Figure 3.14: Discretized Curved L-shaped domains with different distortion metrics.

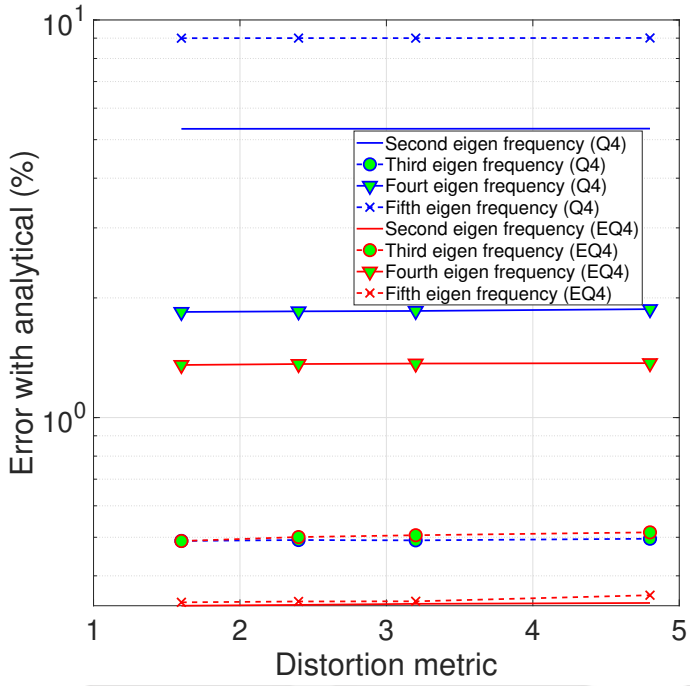


Figure 3.15: Comparative study of convergence analysis with different distortion metrics for curved L shape domain meshed with Q4 and EQ4 elements.

Table 3.16: Analysis data of Q9 and EQ12 elements for curved L shape problem.

Total no. of elements	No. of Equations /FDOF	
	Q9	EQ12
27	381	192
36	501	260
48	657	352
84	1125	628
108	1425	816

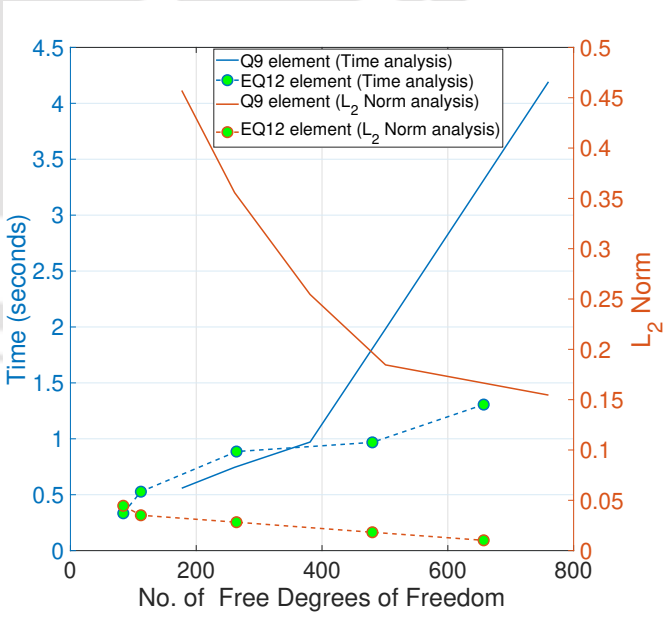


Figure 3.16: Computational cost and convergence analysis of curved L shape domain meshed with Q9 and EQ12 elements.

3.2 Numerical examples of eigenvalue problems in three-dimensions

In this section, we provide detailed insights into the numerical examples examined in our comparative performance analysis between three-dimensional nodal and edge elements. Our selection encompasses the domains with regular shapes, sharp corners and also non-convex configurations to ensure a comprehensive evaluation. For these analyses, we employed TetN10, HexN8, TetE6, and HexE12 finite elements to represent the computational domains.

3.2.1 Cubical domain with perfectly conducting surfaces

In the first example, we have considered a cubical domain with perfectly conducting walls. This is one convex domain. A cube of dimension $\pi \times \pi \times \pi$, as shown in Fig. 4.2, is discretized with HexE12, TetE6, HexN8, and TetN10 finite elements. Detailed numerical analysis data for various elements, is shown in Table 3.17. In Table 3.18, k_0^2 values are listed for all four types of meshes along with the analytical benchmark values [115]. Besides the analytical values, we can see the multiplicity of each eigen value in the bracket.

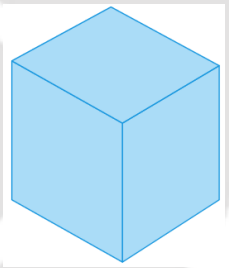


Figure 3.17: Cube with perfectly conducting walls.

Table 3.17: Details of numerical analysis of various elements for the cubical domain problem.

Nodal element			Edge element		
Element Type	No. of elements	No. of equations /FDOF	Element Type	No. of elements	No. of equations /FDOF
HexN8	481	2922	TetE6	481	376
TetN10	343	1800	HexE12	343	756

Numerical results of both nodal and nodal-to-edge elements closely match with the analytical values. We have also observed that the multiplicity of the eigenvalues also match. The last row of the Table 3.18 represents the number of computed zero eigen values (in machine precision) for different meshes with different nodal and edge elements. We can observe from Table 3.17 that in case of the proposed nodal-to-edge elements,

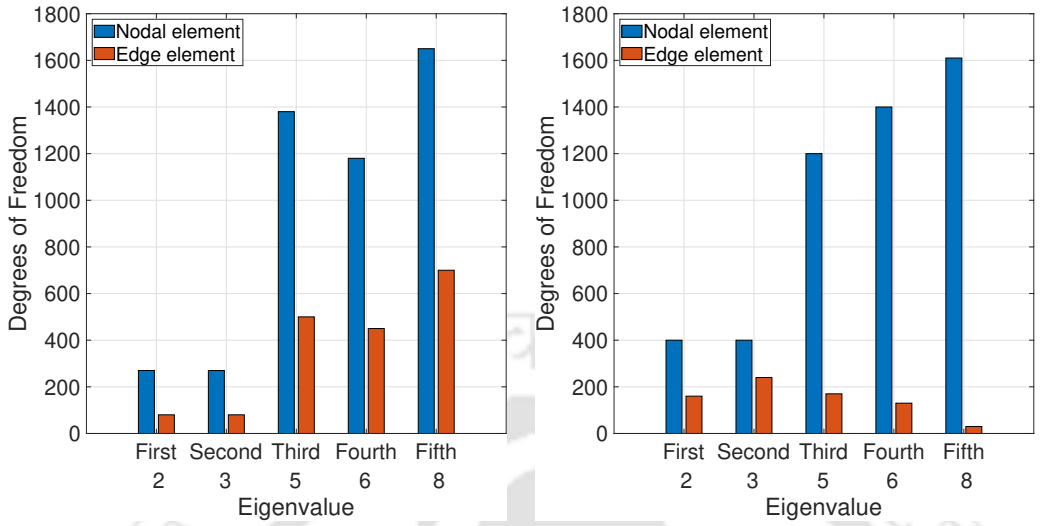
Table 3.18: k_0^2 values for the cubical domain for different elements.

Analytical	Nodal element		Edge element	
	TetN10	HexN8	TetE6	HexE12
2(3)	2.003500	2.033791	1.960873	2.033791
-	2.003612	2.033791	1.970371	2.033791
-	2.003769	2.033791	1.985529	2.033791
3(2)	3.011475	3.050687	2.916051	3.050687
-	3.011652	3.050687	2.941088	3.050687
5(6)	5.042407	5.291983	4.726775	5.291983
-	5.044139	5.291983	4.781211	5.291983
-	5.046123	5.291984	4.831787	5.291984
-	5.049152	5.291984	4.881658	5.291984
-	5.049566	5.291984	4.914506	5.291984
-	5.054583	5.291984	4.925642	5.291984
6(6)	6.074598	6.308879	5.479826	6.308879
-	6.078698	6.308879	5.590799	6.308879
-	6.088814	6.308879	5.632758	6.308879
-	6.089015	6.308879	5.670128	6.308879
-	6.092767	6.308880	5.753275	6.308880
-	6.093264	6.308880	5.868806	6.308880
8(3)	8.147867	8.550176	7.692972	8.550176
-	8.172563	8.550176	8.023065	8.550176
-	8.188397	8.550176	8.043613	8.550176
Number of computed zero eigen values				
-	950	551	427	319

the required numerical results are generated solving less no. of equations as compared to that with nodal elements.

For the first five eigenvalues, we have presented a comparative study of convergence analysis between nodal and edge elements. For HexN8 and HexE12 elements, a scale of 7.5% error is selected. Among successive refined meshes, the limiting mesh is decided for each eigen frequency in such a way that the obtained percentage error for that mesh is less than the 7.5% scale. Fig. 3.18a depicts no. of FDOF of all such limiting meshes, for both HexN8 and HexE12, and for all five eigen frequencies.

Similarly, an error value of 5% is chosen as the scale to compare the tetrahedral elements TetN10 and TetE6. Fig. 3.18b represents the no. of FDOF of the limiting meshes to have percentage error which is less than that 5% scale. From our study, it is evident that both the edge elements are converging faster than respective nodal elements.



(a) Minimum numbers of FDOF necessary to achieve less than 7.5% error (HexN8 vs HexE12).

(b) Minimum numbers of FDOF necessary to achieve less than 5% error (TetN10 vs TetE6).

Figure 3.18: A comparison of the numerical performance between nodal and nodal-to-edge elements in predicting k_0^2 values for cubical domain.

3.2.2 3D-L shaped domain with perfectly conducting walls

We have selected this 3D-L shaped domain [115] to test the effectiveness of the transformed nodal-to-edge elements for non convex domains with sharp edges. The 3D-L shaped domain as shown in Fig. 3.19, is generated by removing the cube $[0,1] \times [0,1] \times [0,1]$ from the rectangular parallelepiped $[-1,1] \times [0,1] \times [-1,1]$. All the boundaries of the domain are perfectly conducting.

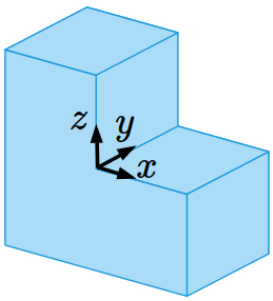


Figure 3.19: 3D-L shaped domain.

Table 3.19 shows the mesh details for the 3D-L shaped domain with TetN10, HexN8, TetE6 and HexE12 finite elements. k_0^2 values generated from the numerical analysis with these elements, are tabulated in Table 3.20 along with the benchmark values from [115]. Both the nodal elements, namely, TetN10 and HexN8 elements, could not predict the second eigen frequency 11.345226. Detailed observation on Table 3.20 reveals that both TetE6 and HexE12 generate more accurate eigen values as compared to TetN10 and

Table 3.19: Details of numerical study of various elements for the 3D-L shaped domain.

Nodal element			Edge element		
Element Type	No. of elements	No. of equations /FDOF	Element Type	No. of elements	No. of equations /FDOF
TetN10	544	3256	TetE6	1172	999
HexN8	375	1640	HexE12	375	800

Table 3.20: k_0^2 values for the 3D-L shaped domain for different elements.

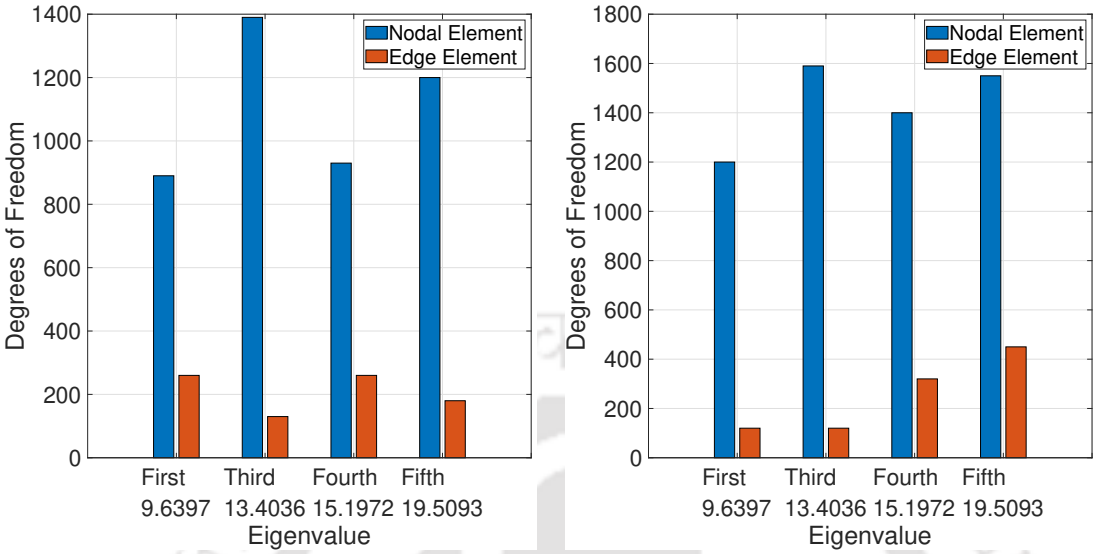
Benchmark	Nodal element		Edge element	
	TetN10	HexN8	TetE6	HexE12
9.639724	9.725360	9.985664	9.681325	9.985664
11.345226	-	-	11.040918	11.663421
13.403636	13.695116	14.038256	13.150709	13.766370
15.197252	14.057556	14.411929	14.909535	15.697204
19.509328	15.252165	15.697204	19.194623	20.184054
19.739209	19.841714	-	19.304015	20.396780
19.739209	19.849318	20.396779	19.508339	20.396780
19.739209	19.855788	20.396779	19.666336	20.396780
21.259084	20.943397	20.396780	20.944727	21.924222
Number of computed zero eigen values				
-	1205	951	857	780

HexN8 elements. Mesh details in Table 3.20 further illustrate that the better accuracy of both the edge elements, is obtained solving less than half of the equations solved for respective nodal elements.

Fig.3.20a represent the minimum number of FDOF of the limiting meshes for HexN8 and HexE12, to attain a percentage error of less than 9% for different eigen frequencies. We can see that solving less than 300 equations, all four eigen frequencies can be accurately predicted by HexE12 elements, whereas more than 1400 equations are required to solve with HexN8 elements in predicting the third eigen frequency.

Similarly, 5% error scale is selected to compare TetN10 and TetE6 elements. Fig. 3.20b depict the relative performance for four eigen values. Here, minimum 1200 equations are required to solve with TetN10 to achieve required accuracy for all the eigen frequencies, whereas TetE6 element require to solve only 400 equations.

Thus, it is evident from both the comparisons that even in the case of non convex domains with sharp edges, both the edge elements are performing far better in predicting the benchmark eigen frequencies as compared to the respective nodal elements.



(a) Minimum numbers of FDOF necessary to achieve less than 9% error (HexN8 vs HexE12).

(b) Minimum number of FDOF necessary to achieve less than 5% error (TetN10 vs TetE6).

Figure 3.20: A comparison of the numerical performance between nodal and nodal-to-edge elements in predicting k_0^2 values for 3D L-shaped domain.

3.2.3 Fichera corner domain with perfectly conducting boundaries

This is another example [115] of a non-convex domain with sharp edges and corners. Fichera corner domain, as shown in Fig. 3.21, is obtained by removing the cube $[0,1]\times[0,1]\times[0,1]$ from the cube $[-1,1]\times[-1,1]\times[-1,1]$. All the boundaries of the volume are treated as perfectly conducting. Table 3.21 represent the mesh details with various nodal and edge elements in terms of no. of elements and total no. of equations . Table 3.22 contains the obtained natural frequencies with various nodal and edge elements along with the analytical benchmark values reported in [35, 115].

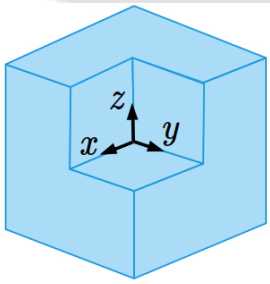


Figure 3.21: Domain of Fichera corner problem.

Both the nodal elements, namely, TetN10 and HexN8 elements, fail to capture the first singular eigenvalue which is due to the sharp corner in the non convex domain. But the nodal-to-edge elements, TetE6 and HexE12, are successful in predicting that eigenvalue. It is evident from the Tables 3.21 and 3.22 that both the edge elements

Table 3.21: Details of numerical analysis with various elements for the Fichera corner problem.

Nodal element			Edge element		
Element Type	No. of elements	No. of equations /FDOF	Element Type	No. of elements	No. of equations /FDOF
TetN10	671	4609	TetE6	2969	2789
HexN8	875	3794	HexE12	1008	2363

Table 3.22: k_0^2 values for the Fichera corner domain for different elements.

Benchmark	Nodal element		Edge element	
	TetN10	HexN8	TetE6	HexE12
3.234320	-	-	3.022205	3.170671
5.882670	6.49579776	6.504939	5.858868	5.927803
5.883710	6.49937643	6.523407	5.864368	5.940017
10.678900	7.27994948	7.328486	10.651293	10.098595
10.683200	10.86519883	11.055402	10.797808	10.386231
10.694500	10.87897155	11.197529	10.821931	10.874614
12.365300	12.67583357	12.940076	12.078230	12.050320
12.372300	-	-	12.264810	12.514143
Number of computed zero eigen values				
-	1050	851	657	540

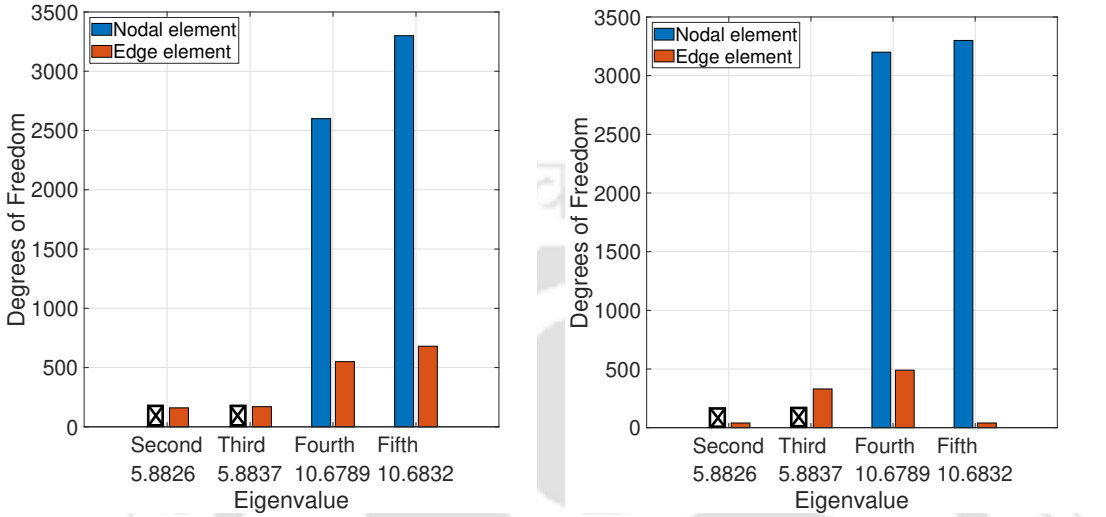
are predicting the benchmark eigenvalues more accurately than the nodal elements.And also, the edge elements are solving around 35% less equations than respective nodal elements.

For the comparative study between HexN8 and HexE12 elements, a scale of 7.5% error is chosen. Fig.3.22a shows that it is required to solve around 700 equations with HexE12 elements to achieve the selected accuracy level for all the frequencies. Whereas, in the case of HexN8 element, to predict second and third eigen frequencies with less than 12.5% error, minimum 3800 FDOF are required. The black crosses are representing that we can not find a mesh with HexN8 elements within our computational capacity, which can predict second and third eigen frequencies within 7.5% error limit.

Fig. 3.22b depicts the comparative performance between TetN10 and TetE6 elements with respect to the scale of 5% error. TetE6 elements are capable to predict all the eigen frequencies solving maximum 500 equations. Whereas TetN10 element requires minimum 4000 FDOF to predict second and third eigen values with an error of less than 11%. Again, the black crosses are representing the same meaning for TetN10 elements for the error limit of 5%.

In this example, edge elements have shown far better performance as compared to the nodal elements. Both the edge elements can predict all five eigen frequencies within

the desired accuracy whereas the respective nodal elements fail to deliver the similar performance even with five times more number of equations.



(a) Minimum number of FDOF necessary to achieve less than 7.5% error (HexN8 vs HexE12).

(b) Minimum number of FDOF necessary to achieve less than 5% error (TetN10 vs TetE6).

Figure 3.22: A comparison of the numerical performance between nodal and nodal-to-edge elements in predicting k_0^2 values for Fichera corner domain.

3.3 Order of convergence for nodal and edge elements

To find the order of convergence for both nodal and edge elements in terms of number of free degrees of freedom (N), we have chosen the findings/results presented in section 3.1.7. In that section, we have conducted a comparative analysis of numerical accuracy in terms of L_2 error norm with mesh convergence, between nodal and edge elements in 2D. For this study, we have chosen four different numerical examples such as square, curved L shape, L shape and inhomogeneous L shape geometries. For each problem, five different meshes of nodal and edge elements are used to perform the convergence analysis. Nodal elements include Q4, Q8, and T6, whereas edge elements include EQ4, EQ12, and ET6. Table 3.23 shows the detailed mesh analysis data for all four domains. L_2-norm error is calculated between numerical and analytical benchmark values as

$$L_2\text{-norm error} = \sqrt{\frac{\sum_{i=1}^{K}\left[(\text{FEM value})_i - (\text{Analytical value})_i\right]^2 \times m_i}{\sum_{i=1}^{K}\left[(\text{Analytical value})_i\right]^2 \times m_i}},$$

where, K is maximum no. of non singular eigen values obtained with the coarsest mesh, m_i is multiplicity of one particular eigen value.

Table 3.23: Analysis data of different nodal and edge elements for L_2-norm error convergence analysis.

Square domain

Total no. of elements	No. of equations/ FDOF		Total no. of elements	No. of equations/ FDOF		Total no. of elements	No. of equations/ FDOF	
	Q4	EQ4		Q9	EQ12		T6	ET8
16	63	24	16	135	112	18	135	120
36	96	60	25	351	180	32	231	196
64	135	112	36	495	264	50	351	264
144	260	180	49	663	480	72	495	380
256	480	231	64	855	760	98	663	560

Curved L shape domain

	Q4	EQ4		Q9	EQ12		T6	ET8
24	98	93	27	177	84	24	177	144
60	219	163	36	261	112	36	288	261
96	717	295	48	381	264	54	432	381
168	813	367	84	501	480	72	501	448
270	909	560	108	760	657	96	680	657

L shape domain

	Q4	EQ4		Q9	EQ12		T6	ET8
192	483	345	48	657	345	96	657	448
300	962	551	75	1005	551	150	1005	710
432	1425	805	108	1425	805	216	1425	1032
588	1817	1107	147	1917	1107	294	1917	1414
768	2081	1457	192	2381	1457	384	2381	1856

Inhomogeneous L shape domain

	Q4	EQ4		Q9	EQ12		T6	ET8
192	483	345	48	657	345	96	657	448
300	962	551	75	1005	551	150	1005	710
432	1425	805	108	1425	805	216	1425	1032
588	1817	1107	147	1917	1107	294	1917	1414
768	2081	1457	192	2381	1457	384	2381	1856

In the order of convergence formula, $L_2^u \leq CN^\gamma$ where L_2^u represents the L_2- norm error, N is the degrees of freedom, C is a constant, and γ is the rate of convergence. After transforming this equation into a logarithmic form:

$$log(L_2^u) \leq log(C) + \gamma log(N).$$

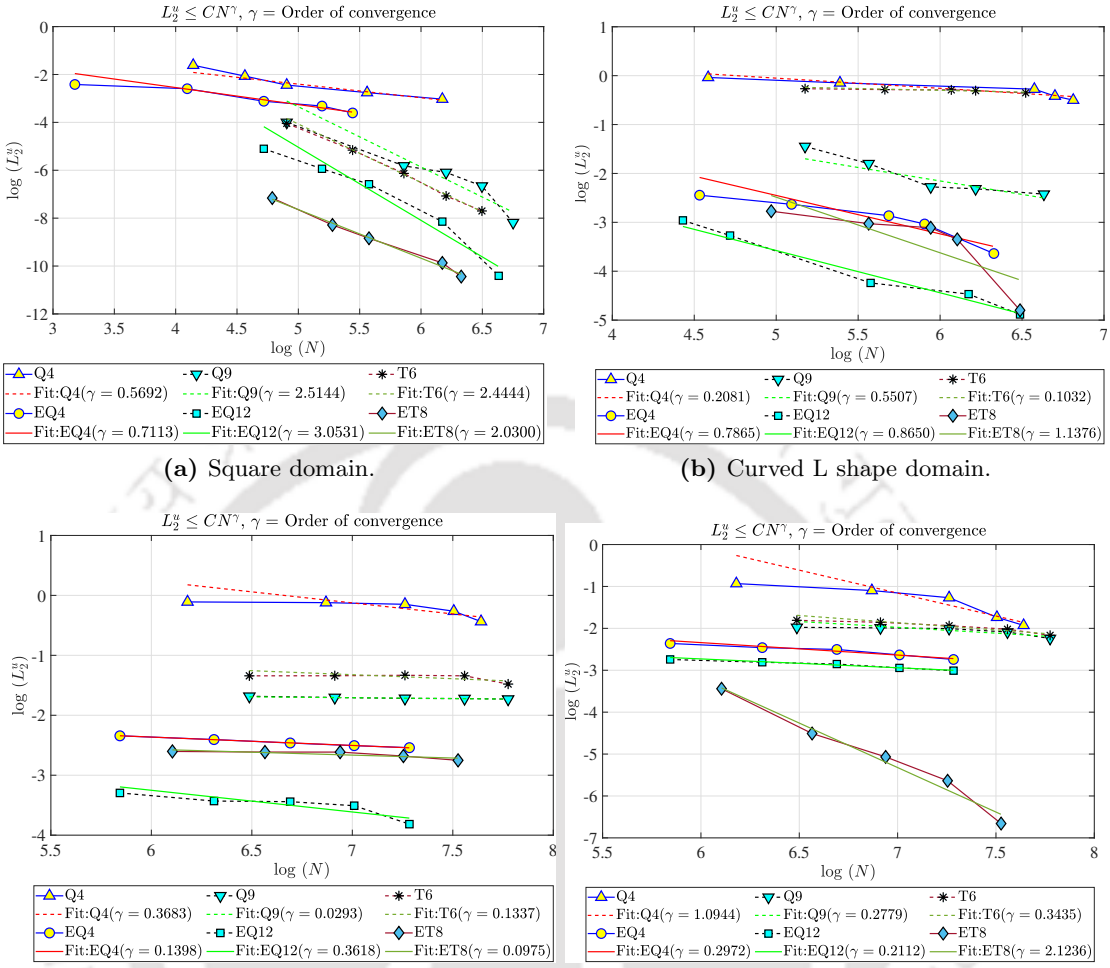
In this expression, the term $log(C)$ is arbitrary and included for mathematical completeness, representing the intercept at the L_2^u axis in the $log - log$ plot. The parameter γ is slope of the line in a $log - log$ plot, denoting order of convergence. $Log(N)$ represents the logarithm of the total degrees of freedom. In the $log - log$ plot, $Log(N)$ is plotted on the $x-$axis, and provides insight into how the error decreases as the number of degrees of freedom (i.e. mesh refinement) increases. For the nodal finite element formulation, for each node, two degrees of freedom are assigned in case of 2D, whereas in 3D, three degrees of freedom are assigned. In edge finite element formulation, one degree of freedom is assigned to each edge, both in 2D and 3D.

To conduct the order of convergence study, for each problem, the logarithm of the L_2 norm error and degrees of freedom for both 2D nodal and edge elements are calculated. These values are then plotted, as shown in Fig. 3.23. The error values are compared between nodal and edge elements for each problem. Fig. 3.23a represents the rate of convergence of both 2D nodal and edge elements for square domain problem. Lower order edge element exhibit an order of convergence value (γ) of approximately 1, indicating a linear rate of convergence with the mesh refinement. For higher order edge elements, the γ value is 3, indicating a cubic rate of error reduction with refinement. This figure demonstrates the fundamental convergence behaviour of nodal and edge in a simple domain.

For curved objects with sharp edges and corners such as curved L shape domain, Fig. 3.23b illustrates the convergence rate for different elements. For other two examples such as homogeneous L shape domain and Inhomogeneous L shape domain, Fig. 3.23c and Fig. 3.23d represent the convergence plots for different nodal and edge elements. Although a similar trend is observed for nodal elements, edge elements provide accurate results compared to nodal elements. It is also noted that the rate of convergence varies for both nodal and edge elements depending on the complexity of the problem, as shown in Fig. 3.23b, Fig. 3.23c, and Fig. 3.23d.

Also, in our study, we have analysed the order of convergence for various 3D nodal and edge elements for three different problems such as cube, Fichera and 3D L shape domains, as discussed in Section 3.2. The nodal elements inlcude HexN8 and TetN10, whereas the edge elements are HexE12 and TetE6. Table 3.24 presents the mesh details along with N used in the study. Fig. 3.24 illustrates the rate of convergence for both 3D nodal and edge elements across the three problems: cube, Fichera and 3D L shape domains.

For a cube with conducting walls problem, Fig. 3.24a represents the rate of convergence for 3D nodal and edge elements. With mesh refinement, both nodal and edge elements predict results with a linear rate of convergence, having a value of γ nearly 1. Fig. 3.24b and Fig. 3.24c are the convergence plots for Fichera and 3D L shape domain problems, respectively. Similar to 2D edge elements, 3D edge elements provide more accurate results than 3D nodal elements with mesh refinement.



(a) Square domain. **(b)** Curved L shape domain.

(c) L shape domain. **(d)** Inhomogenous L shape doamin.

Figure 3.23: Order of convergence for different 2D edge elements.

3.4 Summary

A detailed performance comparison between conventional nodal and transformed edge finite elements has been presented. Various solved examples in two-dimensions include all possible complexities like curved boundaries, non-convex domains, sharp corners, distorted meshes, and non-homogeneous domains. In every case, edge elements have shown better coarse mesh accuracy than nodal elements. We have observed in many cases that, in order to achieve the same level of accuracy, the required no. of equations with edge element is less than half of that with nodal elements. For the non-convex domains with sharp corners, nodal elements can not predict the singular eigen value which is well predicted by all the edge elements. In addition, for such domains, nodal elements predict one additional spurious eigen value which is not present with edge elements. Also, we have observed that mesh distortion does not affect the performance of both nodal and edge elements. Edge elements are performing better with less computational cost than

Table 3.24: Analysis data of different 3D nodal and edge elements for L_2-norm error convergence analysis.

Cube

Total no. of elements	No. of equations/ N		Total no. of elements	No. of equations/ N	Total no. of elements	No. of equations/ N
	HexN8	HexE12		TetN10		TetE6
27	168	36	17	81	17	14
64	378	108	59	338	59	38
125	704	240	263	1586	263	199
216	1170	450	481	2922	481	376
343	1800	756	960	5638	960	856

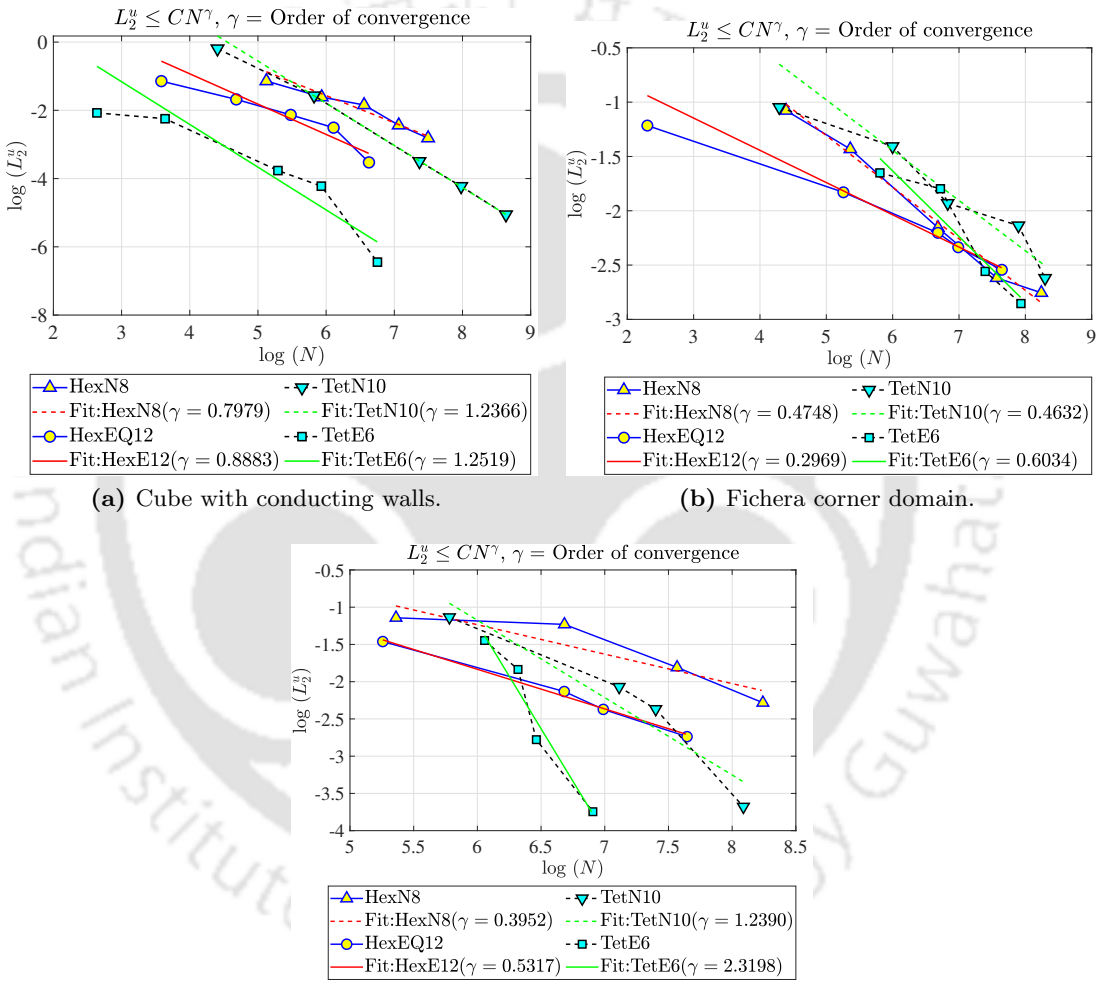
Fichera

	HexN8	HexE12		TetN10		TetE6
25	81	40	18	73	69	40
56	213	192	72	403	445	333
189	799	798	154	925	982	829
449	1935	1084	450	2687	1806	1627
875	3794	2092	671	4013	2969	2789

3D L shape

	HexN8	HexE12		TetN10		TetE6
24	80	26	24	120	120	173
96	332	160	55	324	324	427
192	829	372	206	1228	1228	555
375	1640	800	266	1636	1636	641
768	3621	1760	544	3256	3256	999

nodal elements. We have also carried out detailed performance comparison between three-dimensional edge elements and respective nodal elements. We have observed that edge elements can predict the singular eigenvalues for non-convex domains with sharp corners unlike the nodal elements. Moreover, in all the examples, edge elements have far better coarse mesh accuracy as compared to the nodal elements.



(a) Cube with conducting walls.

(b) Fichera corner domain.

(c) 3D Lshape domain.

Figure 3.24: Order of convergence for different 3D edge elements.

Chapter 4

Exact Conservation in Transient Electromagnetics using Nodal-to-Edge Elements[‡]

In chapter 3, we have conducted electromagnetic eigenvalue analysis using the proposed conversion algorithm. In this chapter, we aim to further extend the application of the conversion technique in electromagnetic transient analysis. We present the variational statement and edge based FEM implementation to solve electromagnetic radiation and scattering problems in time domain. In section 4.1, we summarize the state of art in transient electromagnetics. Then the mathematical formulation is presented in section 5.1, followed by standard benchmark examples in section 4.3.

4.1 Literature survey

In computational electromagnetics, different numerical methods to solve Maxwell's equations in time domain can be broadly classified as Finite Difference Time Domain (FDTD) method [116] and Time Domain FEM (TDFEM) method [95]. FDTD method is simpler to implement whereas TDFEM method offers the flexibility to use unstructured grid [90–96, 102]. In TDFEM method, either one second order equation of a single variable is solved [94–96] or two coupled first order Maxwell's equations are simultaneously solved [90, 92, 93]. In TDFEM, field variables can be interpolated with nodal basis functions [24, 117, 118] or edge elements can be used for discretizations [90–96, 102, 119]. Material discontinuities can be taken care more efficiently with edge elements.

Finite Element Boundary Integral method (FEBI) is also used to solve the time domain electromagnetic radiation and scattering problems in unbounded domains [94, 100–102] where the infinite solution domain is divided into interior and exterior regions

[‡]D. Kamireddy, and A. Nandy, "Exact Conservation in Transient Electromagnetics using edge Elements", *International Journal for Computational Methods in Engineering Science and Mechanics*, 1−15 (2024). . Available at: https://doi.org/10.1080/15502287.2024.2372813

by an artificial boundary. The exterior domain is modelled by boundary integral method whereas FEM is implemented in the interior domain. Two methods are interlinked by field continuity on the artificial boundary. In [96, 120], perfectly matched layer (PML) is used in time domain finite element analysis of radiation and scattering problems.

While solving transient problems, it is crucial to ensure stability of the time-marching scheme to obtain bounded and reliable solutions. Stability depends on the finite element mesh size as well as on the time step used in time-marching [106]. In the FDTD method, proposed by Yee [116], time step has to satisfy the Courant-Friedrich-Levy (CFL) condition to ensure stability. It results in a significant increase in computational time, specially for high spatial mesh resolutions. To bypass the CFL constraint, several new methods [121–124] based on alternating direction implicit (ADI) operator splitting scheme have been proposed. These methods are unconditionally stable while solving a tridiagonal system of equations at each time step. Furthermore, there exist other methods, including locally-one dimensional FDTD (LOD-FDTD) [125, 126], the split-step FDTD (SS-FDTD) method [127], the Crank-Nicholson based factorization splitting scheme (CN-FDTD) [128, 129], and the Leapfrog ADI-FDTD methods [130–132], in which time steps remain independent of mesh sizes.

For a non-homogeneous media or dispersive media, Kole et. al. [133] presented a generalized unconditionally stable algorithm. They achieved this through orthogonal transformations of the skew-symmetric time evolution operator for the electromagnetic fields. Two other FDTD methods in dispersive media are based on evaluation of convolution integrals [134, 135], and on the solution of auxiliary differential equations [136, 137].

Issue of numerical stability happens while solving coupled first order Maxwell's equations with Leapfrog scheme [103, 104, 117, 138] in TDFEM approach. While solving second-order wave equation in TDFEM, time-discretization is done by forward difference, backward difference [139], central difference [91, 140], Newmark methods [141, 142] etc. Although, both Newmark method and central difference have second-order accuracy, the former one is preferred for its unconditional stability [1]. In [143], a general approach based on the theory of discrete finite system analysis was proposed to investigate the stability behavior of TDFEMs for electromagnetic simulations in dispersive media. This approach was later employed to various methods like evolution of convolution integral [144], FEBI [94], and orthogonal TDFEM [105] for stability analysis.

There is an intimate relation between the unconditional stability of an algorithm and the conservation or dissipation of energy in the physical model [106]. Poynting's theorem [107] states that in a lossless medium without sources, the electromagnetic energy of a wave remains constant over time. Any numerical scheme should not violate this principle to attain unconditional stability. Rieben et. al. [103] presented a symplectic method, which maintains the conservation of total electromagnetic energy. In a relavent study [145], authors proposed efficient energy-conserved splitting schemes for three-dimensional Maxwell's equations. In [24], trapezoidal rule was applied to solve the

interior and exterior domain electromagnetic radiation and scattering problems within the nodal framework. The authors also presented conservation of different quantities in continuum framework and they validated that trapezoidal rule mimic these conservation characteristics both algorithmically and in a series of numerical examples. Hence, the time stepping strategy was unconditionally stable, having no limitation on time step from stability viewpoint.

In the present work, the time-stepping strategy presented in [24] is extended into the edge element framework to solve radiation and scattering problems. In edge based TDFEM approach, we have considered the conversion technique discussed in [146] to obtain edge data structure from nodal mesh informations. Here, we have derived one quantity in continuum framework, which should conserve under certain boundary conditions. Furthermore, we have algorithmically shown that the applied time-stepping strategy conserves the quantity. This also has been demonstrated through a series of numerical examples.

The remaining chapter is structured in the following manner. At first, in section 4.2, we will present the variational formulation followed by the time-stepping strategy and edge finite element formulation. After that, we will derive one quantity and prove its conservation in continuum framework as well as algorithmically for the adopted time stepping strategy. Finally, in section 4.3, present strategy is used to solve various standard benchmark examples. Under proper boundary conditions, conservation of the derived continuum quantity is evident in some examples. Accurate numerical simulation for the transient electric fields in various examples exhibit the efficacy of the proposed numerical strategy.

4.2 Mathematical Formulation

In this section, we develop the finite element equation for Maxwell's equation in edge element framework. Also, one quantity is derived in continuum framework which conserves under certain boundary conditions. The developed FEM strategy mimics this conservation characteristic.

4.2.1 Variational statement

Electromagnetic radiation and scattering is governed by following Maxwell's electromagnetic wave equation

$$\epsilon \frac{\partial^2 \boldsymbol{E}}{\partial t^2} + \frac{\partial \boldsymbol{j}}{\partial t} + \boldsymbol{\nabla} \times \left(\frac{1}{\mu} \boldsymbol{\nabla} \times \boldsymbol{E} \right) = \boldsymbol{0}, \tag{4.1}$$

where ϵ and μ are electric permittivity and magnetic permeability of the space, $\boldsymbol{j}$ is the current density.

The variational statement of Eq. (4.1) can be obtained by taking its dot product with the variation of electric field $\boldsymbol{E}_\delta$ and performing integration by parts as

$$\int_\Omega \boldsymbol{E}_\delta \cdot \left[\epsilon \frac{\partial^2 \boldsymbol{E}}{\partial t^2} + \boldsymbol{\nabla} \times \left(\frac{1}{\mu} \boldsymbol{\nabla} \times \boldsymbol{E} \right) + \frac{\partial \boldsymbol{j}}{\partial t} \right] d\Omega = \boldsymbol{0}. \tag{4.2}$$

After using the vector identity $\boldsymbol{a} \cdot (\boldsymbol{\nabla} \times \boldsymbol{b}) = (\boldsymbol{\nabla} \times \boldsymbol{a}) \cdot \boldsymbol{b} - \boldsymbol{\nabla} \cdot (\boldsymbol{a} \times \boldsymbol{b})$, Eq. (4.2) can be expressed as

$$\int_\Omega \left[(\boldsymbol{\nabla} \times \boldsymbol{E}_\delta) \cdot (\frac{1}{\mu} \boldsymbol{\nabla} \times \boldsymbol{E}) - \boldsymbol{\nabla} \cdot \left(\boldsymbol{E}_\delta \times \frac{1}{\mu} \boldsymbol{\nabla} \times \boldsymbol{E} \right) + \boldsymbol{E}_\delta \cdot \left(\epsilon \frac{\partial^2 \boldsymbol{E}}{\partial t^2} + \frac{\partial \boldsymbol{j}}{\partial t} \right) \right] d\Omega = \boldsymbol{0}. \tag{4.3}$$

Using Gauss-divergence theorem and the following vector identity

$$\boldsymbol{a} \cdot (\boldsymbol{b} \times \boldsymbol{c}) = \boldsymbol{b} \cdot (\boldsymbol{c} \times \boldsymbol{a}) = \boldsymbol{c} \cdot (\boldsymbol{a} \times \boldsymbol{b}),$$

Eq. (4.3) can be rewritten as

$$\frac{1}{\mu} \int_\Omega (\boldsymbol{\nabla} \times \boldsymbol{E}_\delta) \cdot (\boldsymbol{\nabla} \times \boldsymbol{E})\, d\Omega + \frac{1}{\mu} \int_\Gamma \boldsymbol{E}_\delta \cdot (\boldsymbol{n} \times \boldsymbol{\nabla} \times \boldsymbol{E})\, d\Gamma + \int_\Omega \epsilon \boldsymbol{E}_\delta \cdot \frac{\partial^2 \boldsymbol{E}}{\partial t^2}\, d\Omega + \int_\Omega \boldsymbol{E}_\delta \cdot \frac{\partial \boldsymbol{j}}{\partial t}\, d\Omega = \boldsymbol{0}. \tag{4.4}$$

Here surface, Γ is $\Gamma_e \cup \Gamma_h \cup \Gamma_\infty$. We specify $\boldsymbol{E} \times \boldsymbol{n} = \boldsymbol{0}$ on perfectly conducting surface Γ_e and magnetic field $\boldsymbol{H} = -\frac{1}{\mu} \boldsymbol{\nabla} \times \boldsymbol{E}$ is prescribed on Γ_h. For exterior domain problems, on the spherical truncation surface Γ_∞, the following first order absorbing condition

$$\boldsymbol{n} \times (\boldsymbol{\nabla} \times \boldsymbol{E}) + \sqrt{\epsilon \mu} \boldsymbol{n} \times \left(\boldsymbol{n} \times \frac{\partial \boldsymbol{E}}{\partial t} \right) = \boldsymbol{0} \tag{4.5}$$

is applied.

Applying above boundary conditions, Eq. 4.4 become

$$\begin{aligned} \int_\Omega \frac{1}{\mu} (\boldsymbol{\nabla} \times \boldsymbol{E}_\delta) \cdot (\boldsymbol{\nabla} \times \boldsymbol{E})\, d\Omega + \int_\Omega \epsilon \boldsymbol{E}_\delta \cdot \frac{\partial^2 \boldsymbol{E}}{\partial t^2}\, d\Omega + \int_{\Gamma_\infty} \sqrt{\frac{\epsilon}{\mu}} (\boldsymbol{E}_\delta \times \boldsymbol{n}) \cdot \left(\frac{\partial \boldsymbol{E}}{\partial t} \times \boldsymbol{n} \right) d\Gamma \\ = - \int_\Omega \boldsymbol{E}_\delta \cdot \frac{\partial \boldsymbol{j}}{\partial t}\, d\Omega + \int_{\Gamma_h} (\boldsymbol{E}_\delta \times \boldsymbol{n}) \cdot \left(\frac{\partial \boldsymbol{H}}{\partial t} \right) d\Gamma. \end{aligned} \tag{4.6}$$

4.2.2 Time-stepping strategy and Finite element equation

Following time-stepping strategies over a time interval $[t_n, t_{n+1}]$ with $t_\Delta := t_{n+1} - t_n$ have been used.

$$\frac{\boldsymbol{v}_n + \boldsymbol{v}_{n+1}}{2} = \frac{\boldsymbol{E}_{n+1} - \boldsymbol{E}_n}{t_\Delta}, \qquad \boldsymbol{E} = \frac{\boldsymbol{E}_n + \boldsymbol{E}_{n+1}}{2}, \tag{4.7a}$$

$$\frac{\partial \boldsymbol{E}}{\partial t} = \frac{\boldsymbol{E}_{n+1} - \boldsymbol{E}_n}{t_\Delta}, \qquad \frac{\partial^2 \boldsymbol{E}}{\partial t^2} = \frac{\boldsymbol{v}_{n+1} - \boldsymbol{v}_n}{t_\Delta}. \tag{4.7b}$$

Then, Eq.(4.6) can be written as

$$\int_{\Omega} \frac{1}{\mu}(\boldsymbol{\nabla} \times \boldsymbol{E}_{\delta}) \cdot \left[\frac{\boldsymbol{\nabla} \times \boldsymbol{E}_{n} + \boldsymbol{\nabla} \times \boldsymbol{E}_{n+1}}{2}\right] d\Omega + \int_{\Omega} \epsilon \boldsymbol{E}_{\delta} \cdot \left(\frac{\boldsymbol{v}_{n+1} - \boldsymbol{v}_{n}}{t_{\Delta}}\right) d\Omega + \int_{\Gamma_{\infty}} \sqrt{\frac{\epsilon}{\mu}}(\boldsymbol{E}_{\delta} \times \boldsymbol{n}) \cdot \left(\frac{\boldsymbol{E}_{n+1} - \boldsymbol{E}_{n}}{t_{\Delta}} \times \boldsymbol{n}\right) d\Gamma = -\int_{\Omega} \boldsymbol{E}_{\delta} \cdot \left(\frac{\boldsymbol{j}_{n+1} - \boldsymbol{j}_{n}}{t_{\Delta}}\right) d\Omega + \int_{\Gamma_{h}} (\boldsymbol{E}_{\delta} \times \boldsymbol{n}) \cdot \left(\frac{\partial \boldsymbol{H}}{\partial t}\right) d\Gamma. \tag{4.8}$$

We discretize the fields and their variations in Eq.(4.7a) as

$$\begin{aligned} \boldsymbol{E} &= \boldsymbol{V}\hat{\boldsymbol{E}}, & \boldsymbol{E}_{\delta} &= \boldsymbol{V}\hat{\boldsymbol{E}}_{\delta}, \\ \boldsymbol{\nabla} \times \boldsymbol{E} &= \boldsymbol{B}\hat{\boldsymbol{E}}, & \boldsymbol{\nabla} \times \boldsymbol{E}_{\delta} &= \boldsymbol{B}\hat{\boldsymbol{E}}_{\delta}, \\ \boldsymbol{E} \times \boldsymbol{n} &= \boldsymbol{n}_{mat}^{T}\boldsymbol{V}\hat{\boldsymbol{E}}, & \boldsymbol{E}_{\delta} \times \boldsymbol{n} &= \boldsymbol{n}_{mat}^{T}\boldsymbol{V}\hat{\boldsymbol{E}}_{\delta}, \end{aligned}$$

where $\hat{\boldsymbol{E}}$ is the edge values of $\boldsymbol{E}$, $\hat{\boldsymbol{E}}_{\delta}$ denote the respective variation of $\hat{\boldsymbol{E}}$ and

$$\boldsymbol{V} = \begin{bmatrix} v_{1x} & v_{2x} & \dots \\ v_{1y} & v_{2y} & \dots \\ v_{1z} & v_{2z} & \dots \end{bmatrix}$$

is the element edge shape functions matrix, v_x, v_y and v_z are the x, y, z components of each edge shape function. $\boldsymbol{B}$-matrix is given as

$$\boldsymbol{B} = \begin{bmatrix} \frac{\partial v_{1z}}{\partial y} - \frac{\partial v_{1y}}{\partial z} & \frac{\partial v_{2z}}{\partial y} - \frac{\partial v_{2y}}{\partial z} & \dots \\ \frac{\partial v_{1x}}{\partial z} - \frac{\partial v_{1z}}{\partial x} & \frac{\partial v_{2x}}{\partial z} - \frac{\partial v_{2z}}{\partial x} & \dots \\ \frac{\partial v_{1y}}{\partial x} - \frac{\partial v_{1x}}{\partial y} & \frac{\partial v_{2y}}{\partial x} - \frac{\partial v_{2x}}{\partial y} & \dots \end{bmatrix} \text{ and } \boldsymbol{n}_{mat} = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix}$$

is the skew-symmetric matrix whose axial vector is $\boldsymbol{n}$ and n_1, n_2 and n_3 are the components of the vector $\boldsymbol{n}$. After substituting the above discretizations into Eq. (4.8), and eliminating $\boldsymbol{v}_{n+1}$ we get

$$\boldsymbol{K}\hat{\boldsymbol{E}}_{n+1} = \boldsymbol{f}, \tag{4.9}$$

where

$$\begin{aligned} \boldsymbol{K} &= \int_{\Omega} \left[\frac{1}{2\mu}\boldsymbol{B}^{T}\boldsymbol{B} + \frac{2\epsilon}{t_{\Delta}^{2}}\boldsymbol{V}^{T}\boldsymbol{V}\right] d\Omega + \int_{\Gamma_{\infty}} \frac{1}{t_{\Delta}}\sqrt{\frac{\epsilon}{\mu}}\boldsymbol{V}^{T}\boldsymbol{n}_{mat}\boldsymbol{n}_{mat}^{T}\boldsymbol{V}\, d\Gamma, \\ \boldsymbol{f} &= \int_{\Omega} \boldsymbol{V}^{T}\left[-\left(\frac{\boldsymbol{j}_{n+1} - \boldsymbol{j}_{n}}{t_{\Delta}}\right) + \frac{2\epsilon}{t_{\Delta}^{2}}\boldsymbol{E}_{n} + \frac{2\epsilon}{t_{\Delta}}\boldsymbol{v}_{n}\right] d\Omega + \int_{\Gamma_{h}} \boldsymbol{V}^{T}\boldsymbol{n}_{mat}\left(\frac{\partial \boldsymbol{H}}{\partial t}\right) d\Gamma_{h} \\ &\quad - \int_{\Omega} \frac{1}{2\mu}\boldsymbol{B}^{T}(\boldsymbol{\nabla} \times \boldsymbol{E}_{n})\, d\Omega + \int_{\Gamma_{\infty}} \frac{1}{t_{\Delta}}\sqrt{\frac{\epsilon}{\mu}}\boldsymbol{V}^{T}\boldsymbol{n}_{mat}\boldsymbol{n}_{mat}^{T}\boldsymbol{E}_{n}\, d\Gamma. \end{aligned}$$

After solving for $\hat{\boldsymbol{E}}_{n+1}$, we can obtain $\boldsymbol{H}_{n+1}$ by using the following relation

$$\boldsymbol{H}_{n+1} = \boldsymbol{H}_n - \frac{t_\Delta}{2\mu}(\boldsymbol{\nabla} \times \boldsymbol{E}_n + \boldsymbol{\nabla} \times \boldsymbol{E}_{n+1}).$$

4.2.3 Conservation of continuum quantity

Here, we derive a quantity in continuum framework which conserves under certain boundary conditions and in the absence of loading. Taking dot product of Eq. (4.1) with $\dot{\boldsymbol{E}}$ and performing integration by parts we get

$$\int_\Omega \left[\dot{\boldsymbol{E}} \cdot \epsilon \frac{\partial^2 \boldsymbol{E}}{\partial t^2} + \dot{\boldsymbol{E}} \cdot \frac{\partial \boldsymbol{j}}{\partial t} + \dot{\boldsymbol{E}} \cdot \boldsymbol{\nabla} \times \left(\frac{1}{\mu}\boldsymbol{\nabla} \times \boldsymbol{E}\right)\right] d\Omega = \boldsymbol{0}. \tag{4.10}$$

Using the relation $\frac{d}{dt}\left[\int_\Omega \frac{\epsilon}{2}\left(\dot{\boldsymbol{E}} \cdot \dot{\boldsymbol{E}}\right) d\Omega\right] = \epsilon \int_\Omega \dot{\boldsymbol{E}} \cdot \frac{\partial^2 \boldsymbol{E}}{\partial t^2}\, d\Omega$ and the vector identity $\boldsymbol{a} \cdot (\boldsymbol{\nabla} \times \boldsymbol{b}) = (\boldsymbol{\nabla} \times \boldsymbol{a}) \cdot \boldsymbol{b} - \boldsymbol{\nabla} \cdot (\boldsymbol{a} \times \boldsymbol{b})$ Eq. (4.10) can be expressed as

$$\begin{aligned}\frac{d}{dt}\left[\int_\Omega \frac{\epsilon}{2}\dot{\boldsymbol{E}} \cdot \dot{\boldsymbol{E}}\, d\Omega\right] + \int_\Omega \dot{\boldsymbol{E}} \cdot \frac{\partial \boldsymbol{j}}{\partial t}\, d\Omega + \int_\Omega \left(\boldsymbol{\nabla} \times \dot{\boldsymbol{E}}\right) \cdot \left(\frac{1}{\mu}\boldsymbol{\nabla} \times \boldsymbol{E}\right) d\Omega - \\ \int_\Omega \boldsymbol{\nabla} \cdot \dot{\boldsymbol{E}} \times \left(\frac{1}{\mu}\boldsymbol{\nabla} \times \boldsymbol{E}\right) d\Omega = \boldsymbol{0}.\end{aligned} \tag{4.11}$$

Applying Gauss-divergence theorem and including the different surface boundary conditions, Eq.(4.11) is given as

$$\begin{aligned}&\frac{d}{dt}\left[\int_\Omega \frac{\epsilon}{2}\dot{\boldsymbol{E}} \cdot \dot{\boldsymbol{E}}\, d\Omega\right] + \int_\Omega \dot{\boldsymbol{E}} \cdot \frac{\partial \boldsymbol{j}}{\partial t}\, d\Omega + \int_\Omega \left(\boldsymbol{\nabla} \times \dot{\boldsymbol{E}}\right) \cdot \left(\frac{1}{\mu}\boldsymbol{\nabla} \times \boldsymbol{E}\right) d\Omega - \\ &\int_{\Gamma_h} \dot{\boldsymbol{H}} \cdot \left(\dot{\boldsymbol{E}} \times \boldsymbol{n}\right) d\Gamma + \int_{\Gamma_e} \left(\frac{1}{\mu}\boldsymbol{\nabla} \times \boldsymbol{E}\right) \cdot \left(\dot{\boldsymbol{E}} \times \boldsymbol{n}\right) d\Gamma + \\ &\int_{\Gamma_\infty} \sqrt{\frac{\epsilon}{\mu}} \left(\dot{\boldsymbol{E}} \times \boldsymbol{n}\right) \cdot \left(\dot{\boldsymbol{E}} \times \boldsymbol{n}\right) d\Gamma = \boldsymbol{0}.\end{aligned} \tag{4.12}$$

Under no loading i.e., $\boldsymbol{j} = \boldsymbol{0}$, and if $\boldsymbol{E} \times \boldsymbol{n} = \boldsymbol{0}$ on Γ_e and $\boldsymbol{H} \times \boldsymbol{n} = \boldsymbol{0}$ on Γ_h, and considering interior domain problems we can write Eq. (4.12) as

$$\frac{1}{2}\frac{d}{dt}\left[\int_\Omega \epsilon\left(\dot{\boldsymbol{E}} \cdot \dot{\boldsymbol{E}}\right) + \int_\Omega \left(\frac{1}{\mu}\boldsymbol{\nabla} \times \boldsymbol{E}\right) \cdot (\boldsymbol{\nabla} \times \boldsymbol{E})\right] d\Omega = \boldsymbol{0}. \tag{4.13}$$

Finally, we get as

$$\frac{1}{2}\int_\Omega \left[\epsilon\left(\dot{\boldsymbol{E}} \cdot \dot{\boldsymbol{E}}\right) + \left(\frac{1}{\mu}\boldsymbol{\nabla} \times \boldsymbol{E}\right) \cdot (\boldsymbol{\nabla} \times \boldsymbol{E})\right] d\Omega = \text{constant}. \tag{4.14}$$

i.e., $\bar{E} = \frac{1}{2}\int_\Omega \left[\epsilon\left(\dot{\boldsymbol{E}} \cdot \dot{\boldsymbol{E}}\right) + \left(\frac{1}{\mu}\boldsymbol{\nabla} \times \boldsymbol{E}\right) \cdot (\boldsymbol{\nabla} \times \boldsymbol{E})\right] d\Omega$ is conserved.

Now, we algorithmically prove that the applied time stepping strategy conserves the derived quantity $\bar{E}$. The semi discrete form of the FEM equation can be written from

Eq. (4.6) as

$$\boldsymbol{M}\ddot{\hat{\boldsymbol{E}}} + \boldsymbol{C}\dot{\hat{\boldsymbol{E}}} + \boldsymbol{K}\hat{\boldsymbol{E}} = \boldsymbol{f}, \tag{4.15}$$

where

$$\boldsymbol{M} = \int_{\Omega} \epsilon \boldsymbol{V}^T \boldsymbol{V}\, d\Omega, \qquad \boldsymbol{C} = \int_{\Gamma_\infty} \sqrt{\frac{\epsilon}{\mu}} \boldsymbol{V}^T \boldsymbol{n}_{mat} \boldsymbol{n}^T{}_{mat} \boldsymbol{V}\, d\Gamma,$$
$$\boldsymbol{K} = \int_{\Omega} \frac{1}{\mu} \boldsymbol{B}^T \boldsymbol{B}\, d\Omega, \qquad \boldsymbol{f} = -\int_{\Omega} \boldsymbol{V}^T \frac{\partial \boldsymbol{j}}{\partial t}\, d\Omega + \int_{\Gamma_h} \boldsymbol{V}^T \boldsymbol{n}_{mat} \frac{\partial \boldsymbol{H}}{\partial t}\, d\Gamma_h.$$

For interior domain problems without loading (i.e., $\boldsymbol{j} = \boldsymbol{0}$) and under specific boundary conditions i.e., $\boldsymbol{E} \times \boldsymbol{n} = \boldsymbol{0}$ on Γ_e and $\boldsymbol{H} \times \boldsymbol{n} = \boldsymbol{0}$ on Γ_h, Eq. (4.15) takes the form as

$$\boldsymbol{M}\ddot{\hat{\boldsymbol{E}}} + \boldsymbol{K}\hat{\boldsymbol{E}} = \boldsymbol{0}. \tag{4.16}$$

After applying the time stepping strategy from Eq. (4.7), Eq. (4.16) is expressed as

$$\boldsymbol{M}\left(\frac{\dot{\hat{\boldsymbol{E}}}_{n+1} - \dot{\hat{\boldsymbol{E}}}_n}{t_\Delta}\right) + \boldsymbol{K}\left(\frac{\hat{\boldsymbol{E}}_{n+1} + \hat{\boldsymbol{E}}_n}{2}\right) = \boldsymbol{0}. \tag{4.17}$$

Now we premultiply the term $\frac{t_\Delta}{2}\left(\dot{\hat{\boldsymbol{E}}}_{n+1} + \dot{\hat{\boldsymbol{E}}}_n\right)^T$ to Eq. (4.17). Then using the relation $\frac{1}{t_\Delta}\left(\hat{\boldsymbol{E}}_{n+1} - \hat{\boldsymbol{E}}_n\right) = \frac{1}{2}\left(\dot{\hat{\boldsymbol{E}}}_{n+1} + \dot{\hat{\boldsymbol{E}}}_n\right)$, Eq. (4.17) can be written as

$$\frac{1}{2}\left(\dot{\hat{\boldsymbol{E}}}_{n+1} + \dot{\hat{\boldsymbol{E}}}_n\right)^T \boldsymbol{M}\left(\dot{\hat{\boldsymbol{E}}}_{n+1} - \dot{\hat{\boldsymbol{E}}}_n\right) + \frac{1}{2}\left(\hat{\boldsymbol{E}}_{n+1} - \hat{\boldsymbol{E}}_n\right)^T \boldsymbol{K}\left(\hat{\boldsymbol{E}}_{n+1} + \hat{\boldsymbol{E}}_n\right) = \boldsymbol{0}. \tag{4.18}$$

Finally using symmetry of $\boldsymbol{K}$ and $\boldsymbol{M}$, Eq.(4.18) can be expressed as

$$\frac{1}{2}\dot{\hat{\boldsymbol{E}}}_{n+1}^T \boldsymbol{M}\dot{\hat{\boldsymbol{E}}}_{n+1} + \frac{1}{2}\hat{\boldsymbol{E}}_{n+1}^T \boldsymbol{K}\hat{\boldsymbol{E}}_{n+1} = \frac{1}{2}\dot{\hat{\boldsymbol{E}}}_n^T \boldsymbol{M}\dot{\hat{\boldsymbol{E}}}_n + \frac{1}{2}\hat{\boldsymbol{E}}_n^T \boldsymbol{K}\hat{\boldsymbol{E}}_n. \tag{4.19}$$

Hence, we have demonstrated algorithmically that the quantity $\bar{E}$ is conserved, irrespective of the choice of time-step t_Δ. This ensures that our proposed time stepping strategy is unconditionally stable from physical argument. A mathematical justification of unconditional stability has been proved in following section 4.2.4.

4.2.4 Stability analysis of the time marching scheme

In this section, we want to present the details of the stability analysis performed for the presented time marching scheme using Z-transform. From Section 4.2, consider the

variational statement given by Eq. 4.6:

$$\int_{\Omega} \frac{1}{\mu}(\boldsymbol{\nabla} \times \boldsymbol{E}_{\delta}) \cdot (\boldsymbol{\nabla} \times \boldsymbol{E})\, d\Omega + \int_{\Omega} \epsilon \boldsymbol{E}_{\delta} \cdot \frac{\partial^2 \boldsymbol{E}}{\partial t^2}\, d\Omega + \int_{\Gamma_{\infty}} \sqrt{\frac{\epsilon}{\mu}} (\boldsymbol{E}_{\delta} \times \boldsymbol{n}) \cdot \left(\frac{\partial \boldsymbol{E}}{\partial t} \times \boldsymbol{n} \right) d\Gamma = -\int_{\Omega} \boldsymbol{E}_{\delta} \cdot \frac{\partial \boldsymbol{j}}{\partial t}\, d\Omega + \int_{\Gamma_h} (\boldsymbol{E}_{\delta} \times \boldsymbol{n}) \cdot \left(\frac{\partial \boldsymbol{H}}{\partial t} \right) d\Gamma. \tag{4.20}$$

We discretize the fields and their variations in Eq. (4.20)

$$\begin{aligned} \boldsymbol{E} &= \boldsymbol{V}\hat{\boldsymbol{E}}, & \boldsymbol{E}_{\delta} &= \boldsymbol{V}\hat{\boldsymbol{E}}_{\delta}, \\ \boldsymbol{\nabla} \times \boldsymbol{E} &= \boldsymbol{B}\hat{\boldsymbol{E}}, & \boldsymbol{\nabla} \times \boldsymbol{E}_{\delta} &= \boldsymbol{B}\hat{\boldsymbol{E}}_{\delta}, \\ \boldsymbol{E} \times \boldsymbol{n} &= \boldsymbol{n}_{mat}^{T}\boldsymbol{V}\hat{\boldsymbol{E}}, & \boldsymbol{E}_{\delta} \times \boldsymbol{n} &= \boldsymbol{n}_{mat}^{T}\boldsymbol{V}\hat{\boldsymbol{E}}_{\delta}, \end{aligned}$$

where $\hat{\boldsymbol{E}}$ is the edge values of $\boldsymbol{E}$, $\hat{\boldsymbol{E}}_{\delta}$ denote the respective variation of $\hat{\boldsymbol{E}}$ and

$$\boldsymbol{V} = \begin{bmatrix} v_{1x} & v_{2x} & \dots \\ v_{1y} & v_{2y} & \dots \\ v_{1z} & v_{2z} & \dots \end{bmatrix}$$

is the element edge shape functions matrix, v_x, v_y and v_z are the x, y, z components of each edge shape function. $\boldsymbol{B}$-matrix is given as

$$\boldsymbol{B} = \begin{bmatrix} \frac{\partial v_{1z}}{\partial y} - \frac{\partial v_{1y}}{\partial z} & \frac{\partial v_{2z}}{\partial y} - \frac{\partial v_{2y}}{\partial z} & \dots \\ \frac{\partial v_{1x}}{\partial z} - \frac{\partial v_{1z}}{\partial x} & \frac{\partial v_{2x}}{\partial z} - \frac{\partial v_{2z}}{\partial x} & \dots \\ \frac{\partial v_{1y}}{\partial x} - \frac{\partial v_{1x}}{\partial y} & \frac{\partial v_{2y}}{\partial x} - \frac{\partial v_{2x}}{\partial y} & \dots \end{bmatrix} \text{ and } \boldsymbol{n}_{mat} = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix}$$

is the skew-symmetric matrix whose axial vector is $\boldsymbol{n}$ and n_1, n_2 and n_3 are the components of the vector $\boldsymbol{n}$.

After substituting above into Eq. (4.20), we can obtain the ordinary differential equation in the time domain as

$$\boldsymbol{M}\frac{d^2\hat{\boldsymbol{E}}}{dt^2} + \boldsymbol{C}\frac{d\hat{\boldsymbol{E}}}{dt} + \boldsymbol{K}\hat{\boldsymbol{E}} = \boldsymbol{f}, \tag{4.21}$$

where

$$\boldsymbol{M} = \int_{\Omega} \epsilon \boldsymbol{V}^T \boldsymbol{V}\, d\Omega, \qquad \boldsymbol{C} = \int_{\Gamma_{\infty}} \sqrt{\frac{\epsilon}{\mu}} \boldsymbol{V}^T \boldsymbol{n}_{mat} \boldsymbol{n}^T{}_{mat} \boldsymbol{V}\, d\Gamma,$$
$$\boldsymbol{K} = \int_{\Omega} \frac{1}{\mu} \boldsymbol{B}^T \boldsymbol{B}\, d\Omega, \qquad \boldsymbol{f} = -\int_{\Omega} \boldsymbol{V}^T \frac{\partial \boldsymbol{j}}{\partial t}\, d\Omega + \int_{\Gamma_h} \boldsymbol{V}^T \boldsymbol{n}_{mat} \frac{\partial \boldsymbol{H}}{\partial t}\, d\Gamma_h.$$

Using Newmark's method with $\beta = 0.25$ and $\gamma = 0.5$ values, we can obtain the following relations:

$$\frac{\boldsymbol{v}_n + \boldsymbol{v}_{n+1}}{2} = \frac{\boldsymbol{E}_{n+1} - \boldsymbol{E}_n}{t_\Delta}, \qquad \boldsymbol{E} = \frac{\boldsymbol{E}_n + \boldsymbol{E}_{n+1}}{2}, \tag{4.22a}$$

$$\frac{\partial \boldsymbol{E}}{\partial t} = \frac{\boldsymbol{E}_{n+1} - \boldsymbol{E}_n}{t_\Delta}, \qquad \frac{\partial^2 \boldsymbol{E}}{\partial t^2} = \frac{\boldsymbol{v}_{n+1} - \boldsymbol{v}_n}{t_\Delta}, \tag{4.22b}$$

$$\frac{\boldsymbol{v}_{n+1} - \boldsymbol{v}_n}{t_\Delta} = \frac{\dot{\boldsymbol{v}}_{n+1} + \dot{\boldsymbol{v}}_n}{2}, \tag{4.22c}$$

where t_Δ is the time-step size. Applying the above relations to discretize Eq. 4.21, we obtain

$$\left[\frac{2\boldsymbol{M}}{t_\Delta^2} + \frac{\boldsymbol{C}}{t_\Delta} + \frac{\boldsymbol{K}}{2}\right]\hat{\boldsymbol{E}}_{n+1} = \left[\frac{2\boldsymbol{M}}{t_\Delta^2} - \frac{\boldsymbol{K}}{2}\right]\hat{\boldsymbol{E}}_n + \frac{2\boldsymbol{M}}{t_\Delta}\hat{\boldsymbol{v}}_n + \frac{\boldsymbol{C}}{t_\Delta}\hat{\boldsymbol{E}}_n + \frac{\boldsymbol{f}_n + \boldsymbol{f}_{n+1}}{2}. \tag{4.23}$$

With the following relation:

$$\boldsymbol{v}_n = \frac{\boldsymbol{E}_n - \boldsymbol{E}_{n-1}}{t_\Delta} + \frac{t_\Delta}{2}\left(\frac{\dot{\boldsymbol{v}}_n + \dot{\boldsymbol{v}}_{n-1}}{2}\right), \qquad \frac{\boldsymbol{v}_{n-1} + \boldsymbol{v}_n}{2} = \frac{\boldsymbol{E}_n - \boldsymbol{E}_{n-1}}{t_\Delta}, \tag{4.24a}$$

Eq. 4.23 becomes as

$$\left[\frac{\boldsymbol{M}}{t_\Delta^2} + \frac{\boldsymbol{C}}{2t_\Delta} + \frac{\boldsymbol{K}}{4}\right]\hat{\boldsymbol{E}}_{n+1} = \left[\frac{2\boldsymbol{M}}{t_\Delta^2} - \frac{\boldsymbol{K}}{2}\right]\hat{\boldsymbol{E}}_n - \left[\frac{\boldsymbol{M}}{t_\Delta^2} - \frac{\boldsymbol{C}}{2t_\Delta} + \frac{\boldsymbol{K}}{4}\right]\hat{\boldsymbol{E}}_{n-1} + \bar{\boldsymbol{f}}_n, \tag{4.25}$$

where $\bar{\boldsymbol{f}}_n = \frac{\boldsymbol{f}_{n-1} + 2\boldsymbol{f}_n + \boldsymbol{f}_{n+1}}{8}$.

4.2.4.1 Stability analysis using Z-transforms

For the system derived, to analyse the stability of the time-marching process, we employ the method proposed in [143]. For a lossless media such that $\boldsymbol{C} = 0$, Eq. 4.25 can be written as

$$\left[\frac{\boldsymbol{M}}{t_\Delta^2} + \frac{\boldsymbol{K}}{4}\right]\hat{\boldsymbol{E}}_{n+2} - \left[\frac{2\boldsymbol{M}}{t_\Delta^2} - \frac{\boldsymbol{K}}{2}\right]\hat{\boldsymbol{E}}_{n+1} + \left[\frac{\boldsymbol{M}}{t_\Delta^2} + \frac{\boldsymbol{K}}{4}\right]\hat{\boldsymbol{E}}_n = \bar{\boldsymbol{f}}_{n+1}. \tag{4.26}$$

Performing Z-transform to the left hand side of the Eq. 4.26, we obtain as

$$\frac{1}{t_\Delta^2}\boldsymbol{M}z^2\{\tilde{E}\} + \frac{1}{4}\boldsymbol{K}z^2\{\tilde{E}\} - \frac{2}{t_\Delta^2}\boldsymbol{M}z\{\tilde{E}\} + \frac{1}{2}\boldsymbol{K}z\{\tilde{E}\} + \frac{1}{t_\Delta^2}\boldsymbol{M}\{\tilde{E}\} + \frac{1}{4}\boldsymbol{K}z\{\tilde{E}\} = 0, \tag{4.27}$$

where $\{\tilde{E}\}$ denotes the Z-transform of $\hat{\boldsymbol{E}}_n$. Finally, we can express the Eq. 4.27 as

$$\frac{-4(z-1)^2}{(z+1)^2}\{\tilde{E}\} = t_\Delta^2\boldsymbol{M}^{-1}\boldsymbol{K}\{\tilde{E}\}. \tag{4.28}$$

Clearly, $\frac{-4(z-1)^2}{(z+1)^2}$ is the eigenvalue of matrix $t_\Delta^2 \boldsymbol{M}^{-1}\boldsymbol{K}$. To obtain the condition on stability of the TDFEM numerical schemes, the magnitude of z should be bounded by one or inside the unit circle in the complex z plane. Assuming the maximum eigenvalue of the matrix $t_\Delta^2 \boldsymbol{M}^{-1}\boldsymbol{K}$ to be λ_{max}, then the condition on the time-step t_Δ is given as

$$t_\Delta \leq \frac{\sqrt{\lambda_{max}}}{\sqrt{\rho(\boldsymbol{M}^{-1}\boldsymbol{K})}}, \tag{4.29}$$

where $\rho(\boldsymbol{M}^{-1}\boldsymbol{K})$ denotes the spectral radius of matrix $\boldsymbol{M}^{-1}\boldsymbol{K}$. To determine λ_{max}, we can seek the roots of the following characteristic equation:

$$(z-1)^2 + \frac{\lambda}{4}(z+1)^2 = 0, \tag{4.30}$$

by varying λ value from zero to λ_{max}. The roots of Eq. 4.30 can be found as

$$z = \frac{1 - \frac{\lambda}{4} \pm 4\sqrt{-\lambda}}{1 + \frac{\lambda}{4}} \tag{4.31}$$

in a lossless media. For some λ value, if the roots of the Eq. 4.31 go beyond the unit circle in the complex plane, corresponding λ gives some limit on t_Δ. In our formulation, for any value of λ, the roots of the Eq. 4.31 always on the unit circle, indicating that the derived time marching scheme is unconditionally stable. To illustrate this, $|z|$ values are calculated for λ values ranging from 0 to 1×10^6 with an increment of 0.1. The results are plotted in Fig. 4.1, confirming the stability of the scheme as $|z|$ remains constant at 1.

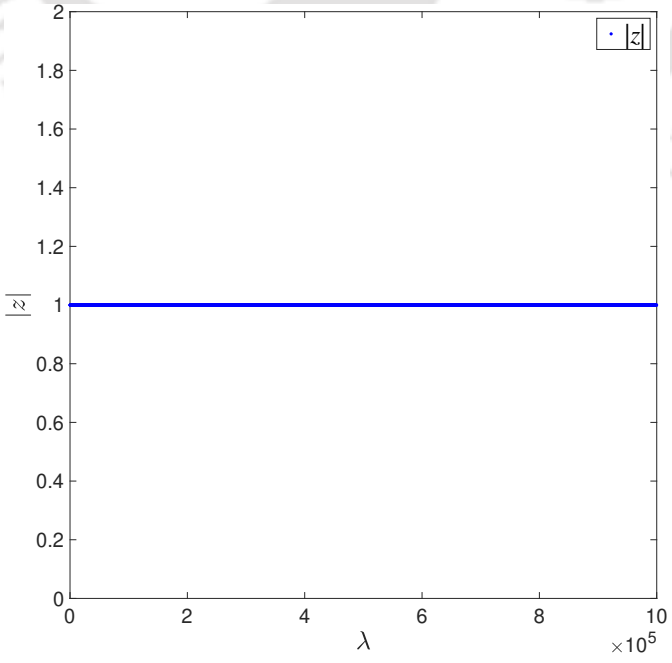


Figure 4.1: Plot showing $|z|$ values for different λ values.

4.2.5 Scattering from conducting and dielectric bodies

In this section, we describe additional mathematical formulation for scattering from conducting and dielectric bodies. Let us consider an incident wave $\boldsymbol{E}_{\text{inc}}$ impinges on a conducting or dielectric body and it is known that total field, $\boldsymbol{E} = \boldsymbol{E}_{\text{scat}} + \boldsymbol{E}_{\text{inc}}$. In the scattering analysis, the scattered field also satisfies the radiation absorbing boundary condition (Eq. 4.5) on the truncation boundary Γ_∞ as

$$\boldsymbol{n} \times (\boldsymbol{\nabla} \times \boldsymbol{E}_{\text{scat}}) + \sqrt{\epsilon\mu}\boldsymbol{n} \times \left(\boldsymbol{n} \times \frac{\partial \boldsymbol{E}_{\text{scat}}}{\partial t}\right) = \boldsymbol{0}. \tag{4.32}$$

It can be rewritten in terms of the total field $\boldsymbol{E}$ as

$$\boldsymbol{n} \times (\boldsymbol{\nabla} \times \boldsymbol{E}) = -\sqrt{\epsilon\mu}\boldsymbol{n} \times \left(\boldsymbol{n} \times \frac{\partial \boldsymbol{E}}{\partial t}\right) + \sqrt{\epsilon\mu}\boldsymbol{n} \times \left(\boldsymbol{n} \times \frac{\partial \boldsymbol{E}_{\text{inc}}}{\partial t}\right) - \boldsymbol{n} \times \mu \frac{\partial \boldsymbol{H}_{\text{inc}}}{\partial t}. \tag{4.33}$$

Hence, 2$^{\text{nd}}$ term in left hand side (LHS) of Eq. (4.4) can be expressed as

$$\begin{aligned}\frac{1}{\mu}\int_\Gamma \boldsymbol{E}_\delta \cdot (\boldsymbol{n} \times \boldsymbol{\nabla} \times \boldsymbol{E})\, d\Gamma &= -\sqrt{\frac{\epsilon}{\mu}}\int_{\Gamma_\infty} \boldsymbol{E}_\delta \cdot \left[\boldsymbol{n} \times \left(\boldsymbol{n} \times \frac{\partial \boldsymbol{E}}{\partial t}\right)\right] d\Gamma \\ &+ \sqrt{\frac{\epsilon}{\mu}}\int_{\Gamma_\infty} \boldsymbol{E}_\delta \cdot \left[\boldsymbol{n} \times \left(\boldsymbol{n} \times \frac{\partial \boldsymbol{E}_{\text{inc}}}{\partial t}\right)\right] d\Gamma - \int_{\Gamma_\infty} (\boldsymbol{E}_\delta \times \boldsymbol{n}) \cdot \left(\frac{\partial \boldsymbol{H}_{\text{inc}}}{\partial t}\right) d\Gamma.\end{aligned}$$

Using the proposed time stepping strategy, we can rewrite the above term as

$$\begin{aligned}\frac{1}{\mu}\int_\Gamma \boldsymbol{E}_\delta \cdot (\boldsymbol{n} \times \boldsymbol{\nabla} \times \boldsymbol{E})\, d\Gamma &= \int_\Gamma \sqrt{\frac{\epsilon}{\mu}} (\boldsymbol{E}_\delta \times \boldsymbol{n}) \cdot \left[\left(\frac{\boldsymbol{E}_{n+1} - \boldsymbol{E}_n}{t_\Delta}\right) \times \boldsymbol{n}\right] d\Gamma \\ &- \int_{\Gamma_\infty} \sqrt{\frac{\epsilon}{\mu}} (\boldsymbol{E}_\delta \times \boldsymbol{n}) \cdot \left(\frac{\partial \boldsymbol{E}_{\text{inc}}}{\partial t} \times \boldsymbol{n}\right) d\Gamma - \int_{\Gamma_\infty} (\boldsymbol{E}_\delta \times \boldsymbol{n}) \cdot \left(\frac{\partial \boldsymbol{H}_{\text{inc}}}{\partial t}\right) d\Gamma.\end{aligned}$$

Therefore, for scattering analysis, load vector $\boldsymbol{f}$ from Eq. (4.9) is modified to $\bar{\boldsymbol{f}}$ with additional terms as

$$\bar{\boldsymbol{f}} = \boldsymbol{f} + \int_{\Gamma_\infty} \sqrt{\frac{\epsilon}{\mu}} (\boldsymbol{E}_\delta \times \boldsymbol{n}) \cdot \left(\frac{\partial \boldsymbol{E}_{\text{inc}}}{\partial t} \times \boldsymbol{n}\right) d\Gamma + \int_{\Gamma_\infty} (\boldsymbol{E}_\delta \times \boldsymbol{n}) \cdot \left(\frac{\partial \boldsymbol{H}_{\text{inc}}}{\partial t}\right) d\Gamma.$$

4.3 Numerical examples

To validate the proposed formulation, we have conducted transient analysis for different benchmark numerical examples including electromagnetic radiating and scattering analysis of both conducting and dielectric bodies. In our simulation, 6-edge tetrahedral element (TetE6) is used to mesh the computational domain. To ensure consistently accurate and reliable results, we have chosen two different mesh sizes for each problem. And also, we choose speed of the light $c = 3{\times}10^8/\sqrt{\epsilon_r\mu_r}$ m/s.

4.3.1 Cube with conducting walls and internal electromagnetic fields and currents

A cube of dimension $\pi \times \pi \times \pi$ as shown in Fig. 4.2 is considered. All the sides of the cube are treated as perfectly conducting. Here, conducting boundary condition is implemented by applying $\boldsymbol{E} \times \boldsymbol{n} = \boldsymbol{0}$ on the elements of the boundary surfaces. Analytical solution [27] which is shown below, exactly satisfies the governing equations and boundary conditions.

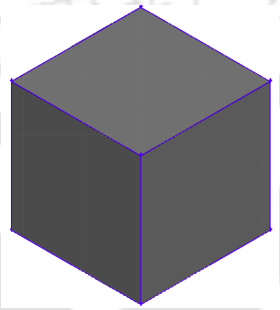


Figure 4.2: Cube

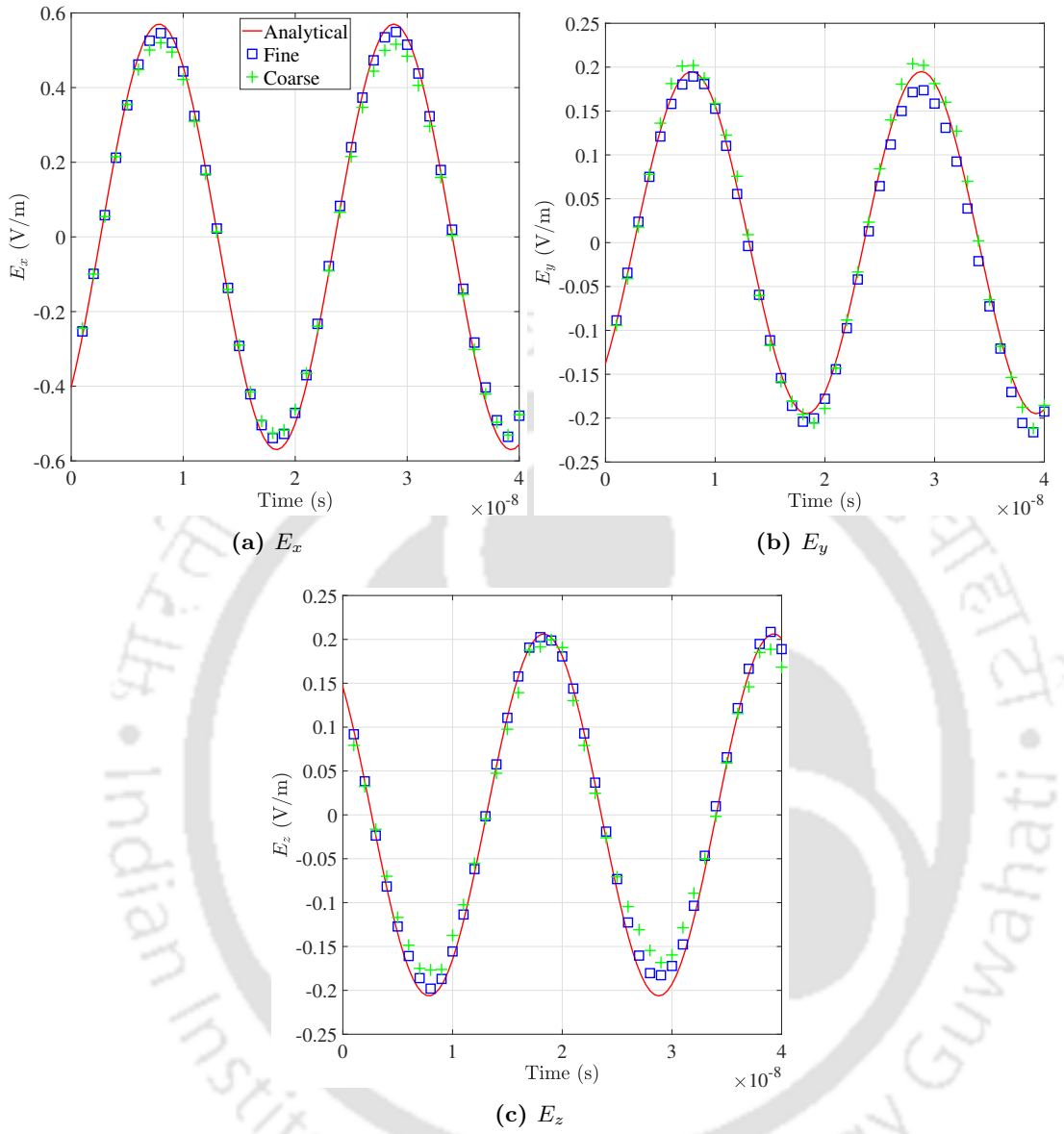
$$
\begin{aligned}
E_x &= 2\cos(x)\sin(y)\sin(z)\left[\cos(\omega t) - \sin(\omega t)\right], \\
E_y &= \sin(x)\cos(y)\sin(z)\left[\sin(\omega t) - \cos(\omega t)\right], \\
E_z &= \sin(x)\sin(y)\cos(z)\left[\sin(\omega t) - \cos(\omega t)\right], H_x = 0, \\
H_y &= \frac{-3}{\mu\omega}\cos(x)\sin(y)\cos(z)\left[\cos(\omega t) + \sin(\omega t)\right], \\
H_z &= \frac{3}{\mu\omega}\cos(x)\cos(y)\sin(z)\left[\cos(\omega t) + \sin(\omega t)\right], \\
j_x &= \left(\frac{2\epsilon\mu\omega^2 - 6}{\mu\omega}\right)\cos(x)\sin(y)\sin(z)\left[\cos(\omega t) + \sin(\omega t)\right], \\
j_y &= \left(\frac{3 - \epsilon\mu\omega^2}{\mu\omega}\right)\sin(x)\cos(y)\sin(z)\left[\cos(\omega t) + \sin(\omega t)\right], \\
j_z &= \left(\frac{3 - \epsilon\mu\omega^2}{\mu\omega}\right)\sin(x)\sin(y)\cos(z)\left[\cos(\omega t) + \sin(\omega t)\right],
\end{aligned}
\tag{4.34}
$$

where E_x, E_y, E_z are the components of electric field in x, y and z directions, respectively. Similarly, H_x, H_y, H_z are the components of magnetic field, whereas j_x, j_y, and j_z are the components of current density.

Table 4.1: Mesh and equation details of a cube with conducting walls and supply of $\boldsymbol{j}$.

	No. of elements	No. of Equations
Coarse mesh	908	798
Fine mesh	2648	2586

For transient analysis, the cube is modelled with both coarse and fine meshes, detailed in Table 4.1. Throughout our simulation, we have supplied an external current density $\boldsymbol{j}$ as shown by Eq. 4.34 to the domain. Additionally, initial electric field values i.e., $\boldsymbol{E}$ at time $t = 0$ seconds, obtained from the Eq. 4.34, are applied. With the parameters, $\omega = 3 \times 10^8$ rad/s, $\epsilon_r = \mu_r = 1$, and time-step t_Δ of 10^{-9} second, we have simulated electric field upto 4×10^{-8} seconds. We have compared FEM results with the analytical counterpart for the transient electric fields at $(2.78, 0.49, 2.67)$ in Fig. 5.9. Both the coarse and fine mesh results are perfectly matching with the analytical one. Fig. 4.4 shows the time response of the derived mathematical quantity $\bar{E}$ based on FEM data. For the current numerical example, supply of current density $\boldsymbol{j}$ is stopped after 4 $\times 10^{-8}$ seconds. Therefore, as discussed in section 4.2.3, quantity $\bar{E}$ should be conserved after this point of time which is accurately reflected in Fig. 4.4.



(a) E_x **(b)** E_y

(c) E_z

Figure 4.3: Comparison of transient electric fields at $(x, y, z) = (2.78, 0.49, 2.67)$ for the cube with conducting walls problem.

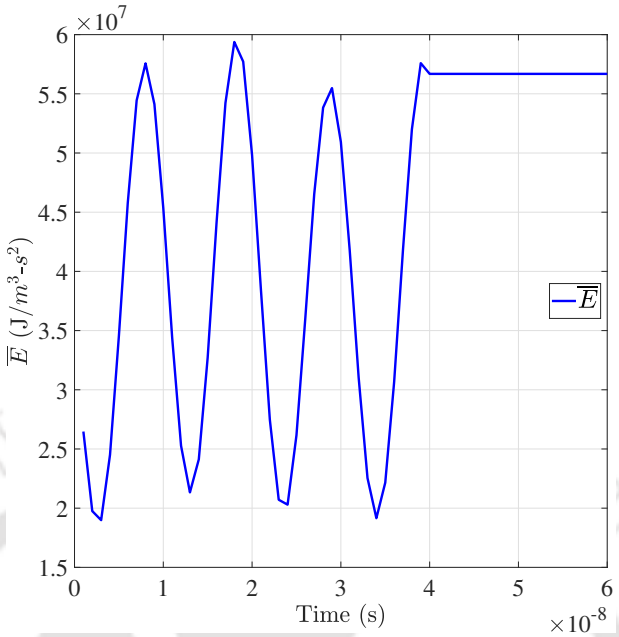


Figure 4.4: Conservation of $\bar{E}$ in the absence of $\boldsymbol{j}$ and with $\boldsymbol{E} \times \boldsymbol{n} = \boldsymbol{0}$ on boundary of the domain.

To understand the effectiveness of the transformed edge elements over nodal elements in time domain problems, computational cost analysis is performed. We have compared the required computational cost along with number of required free degrees of freedom (FDOF) and L_2 error norm for HexN8 and HexE12 elements. Different mesh refinements such as 64, 125, 216, 343, and 512 HexN8 and HexE12 elements are used to discretize the computational domain in our numerical analysis. Table 4.2 represents respective no. of free degrees of freedom. Throughout our simulation, we have supplied an external current density $\boldsymbol{j}$ as shown by Eq. 4.34 to the domain. Additionally, initial electric field values i.e., $\boldsymbol{E}$ at time $t = 0$ seconds, obtained from the Eq. 4.34, are applied. With the parameters, $\omega = 3 \times 10^8$ rad/s, $\epsilon_r = \mu_r = 1$, and time-step t_Δ of 10^{-9} second, we have simulated electric field upto 4×10^{-8} seconds. Our computational codes are run on a Dell vostro model, using Intel Core i3 4.1 GHz processor with 4 GB RAM. We have calculated the L_2-norm error between the simulation results ($\boldsymbol{E}_{\text{FEM}}$) obtained from both Nodal and Edge finite elements and analytical benchmark values ($\boldsymbol{E}_{\text{A}}$) as

$$L_2\text{-norm error} = \frac{1}{N}\sqrt{\sum_{i=1}^{N}\left[\frac{\sum_{k=x,y,z}\left(E_{k(\text{FEM})_i} - E_{k(\text{A})_i}\right)^2}{(E_{x(\text{A})_i})^2 + (E_{y(\text{A})_i})^2 + (E_{z(\text{A})_i})^2}\right]}$$

where N is the total number of nodes of the discretized domain.

We have plotted the computational time in seconds required to achieve the certain L_2-norm error along with total free degrees of freedom as shown in Fig. 4.5. We can see

Table 4.2: Analysis data of HexN8 and HexE12 elements for radiation analysis of a cube with conducting walls problem.

Total no. of elements	No. of Equations /FDOF	
	HexN8	HexE12
64	192	162
125	352	240
216	650	352
343	1080	628
512	1666	816

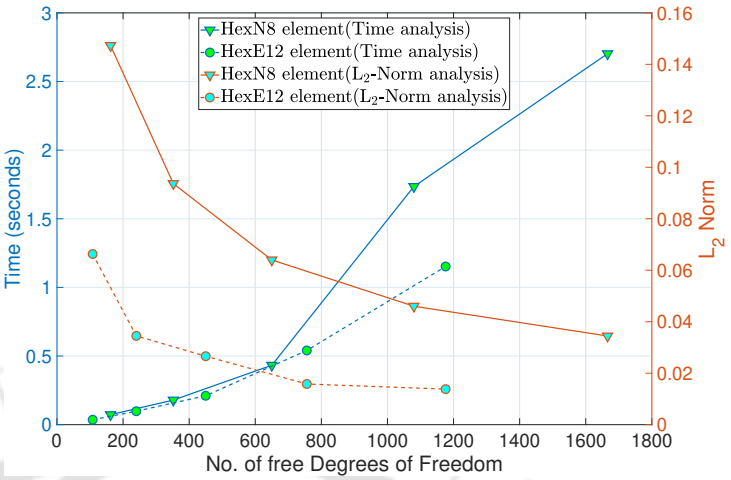


Figure 4.5: Computational cost and convergence analysis of cube with conducting walls meshed with HexN8 and HexE12 elements.

that with HexE12 we achieve 0.015 average L_2 error norm with around 756 FDOF in aroud 1.73 seconds, whereas with almost same no. of FDOF in HexN8 we have achieved average L_2 error norm as 0.06 in 0.9 seconds. Also from Fig. 4.5, we can notice that with HexN8 element we can achieve average L_2 norm of 0.035 in computational time of 2.7 seconds whereas with HexE12 with in computational time of 1.15 seconds we can achieve average L_2 norm of 0.013. Overall computational efficacy of edge elements is clearly visible. It will be scaled up to a significant value for realistic large problem domains.

4.3.2 Cube with prescribed magnetic field on the boundary

This example is chosen from [139], and the analytical solution is given as

$$\begin{aligned}
&E_x = \sin(\omega t - kz), \quad E_y = \sin(\omega t - kx), \quad E_z = \sin(\omega t - ky),\\
&H_x = \frac{k}{\mu\omega}\sin(\omega t - ky), \quad H_y = \frac{k}{\mu\omega}\sin(\omega t - kz), \quad H_z = \frac{k}{\mu\omega}\sin(\omega t - kx),\\
&j_x = \left(\frac{k^2}{\mu\omega} - \omega\epsilon\right)\cos(\omega t - kz), \quad j_y = \left(\frac{k^2}{\mu\omega} - \omega\epsilon\right)\cos(\omega t - kx), \quad \text{and}\\
&j_z = \left(\frac{k^2}{\mu\omega} - \omega\epsilon\right)\cos(\omega t - ky).
\end{aligned} \tag{4.35}$$

A cube having each side of dimension 2 is modelled with properties $\omega = 2$, $k = 3$, $\mu = 30/8$ and $\epsilon = 0.2$. Table 4.3 provides mesh and equation details about the numerical analysis. Using Eq. 4.35 current density $\boldsymbol{j}$ is supplied as loading condition, electric field at time $t = 0$ seconds is supplied as initial condition and the magnetic field $\boldsymbol{H}$ is also supplied to all the boundary walls of the cube as the boundary condition. With a time step of 0.1 second, the simulation is conducted upto 4 seconds.

Table 4.3: Details of the numerical analysis of cube problem with magnetic loading on walls.

	No. of elements	No. of Equations
Coarse mesh	4217	5765
Fine mesh	7302	9572

Fig. 4.6 shows the time response of FEM results from both meshes along with analytical values at a point (1.01, 0.16, 0.98) inside the cube. The plot demonstrates a good match between the FEM results and the analytical values. We set $\boldsymbol{H} \times \boldsymbol{n}$ to zero after 4 seconds, and after 5 seconds $\boldsymbol{j}$ is set to zero. Fig. 4.7 shows the time variation of $\bar{E}$ as per FEM data. After 5 seconds there is no loading i.e., $\boldsymbol{j} = \boldsymbol{0}$, $\boldsymbol{H} \times \boldsymbol{n} = \boldsymbol{0}$ on Γ_h and there is no Γ_e. Hence, $\bar{E}$ should be conserved as discussed in section 4.2.3. The adopted time-stepping strategy perfectly mimic this conservation characteristics in Fig. 4.7.

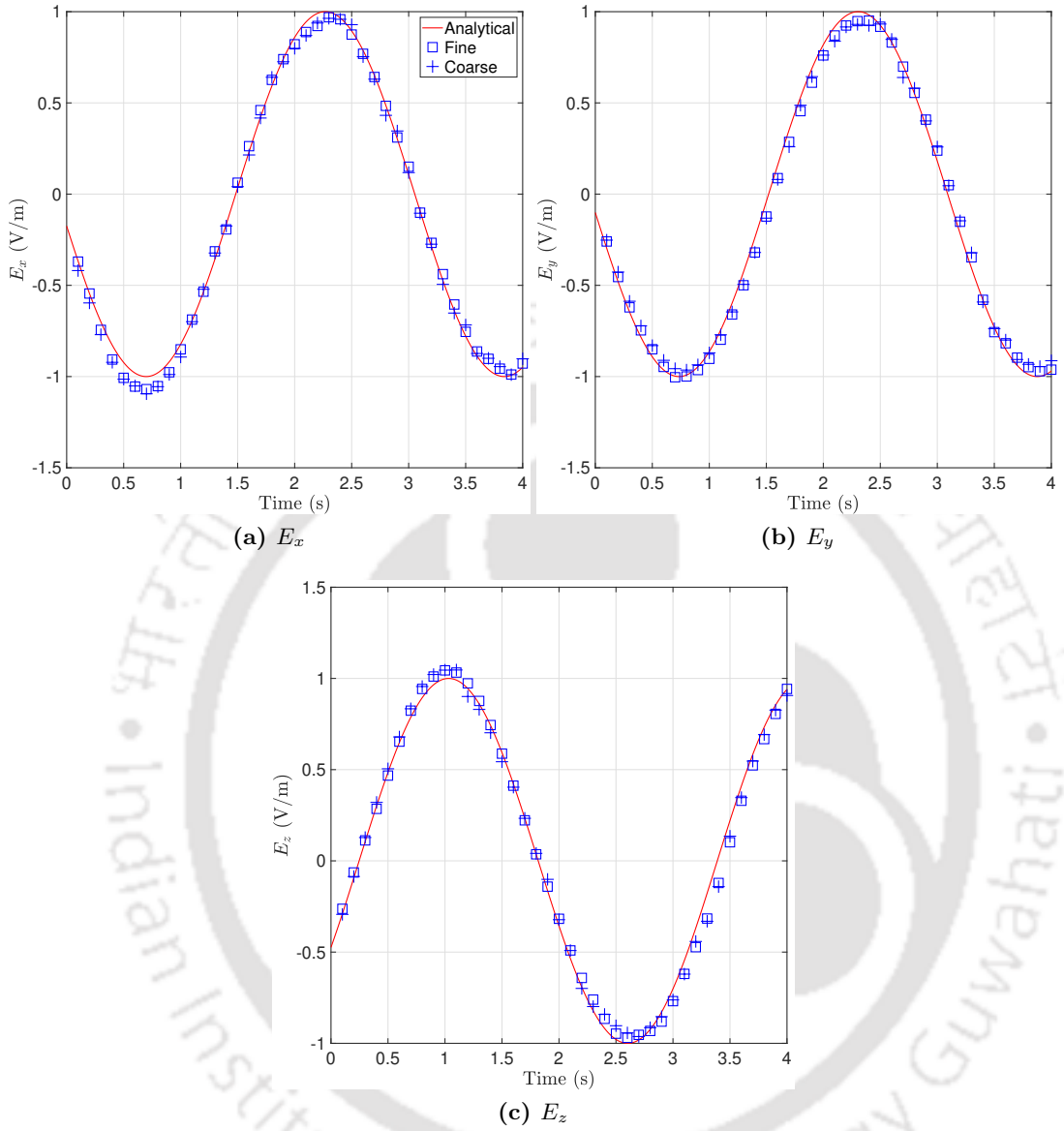
4.3.3 Sphere with Transverse Magnetic (TM) loading

In previous two examples, we have solved electromagnetic radiation problems in interior domain. The present example from [27] is one radiation analysis in exterior domain. The computational domain is a hollow sphere with inner and outer radii a and R_∞, respectively. Radiation absorbing boundary condition is imposed on the outer surface, whereas on the inner surface at a, transverse magnetic loading $\boldsymbol{H}$ is applied as per Eq. (4.36) Solution for this problem with transverse magnetic loading from [27], is given as

$$\begin{aligned}
E_r &= \frac{\omega n(n+1)}{kr} \sin(m\phi) P_n^{(m)}(\cos\theta) \left[j_n(kr) \sin(\omega t) - y_n(kr) \cos(\omega t) \right], \\
E_\theta &= t_1 t_2 \left[t_4 \cos(\omega t) - t_5 \sin(\omega t) \right], \\
E_\phi &= -t_3 \left[t_4 \cos(\omega t) - t_5 \sin(\omega t) \right], \\
H_r &= 0, \\
H_\theta &= \frac{\omega r t_3}{c\mu} \left[j_n(kr) \cos(\omega t) + y_n(kr) \sin(\omega t) \right], \\
H_\phi &= \frac{\omega r t_1 t_2}{c\mu} \left[j_n(kr) \cos(\omega t) + y_n(kr) \sin(\omega t) \right],
\end{aligned} \tag{4.36}$$

where m and n are any integers, and

$$\begin{aligned}
t_1 &= \frac{\sin(m\phi)}{r \sin\theta}, \\
t_2 &= (n+1) \cos\theta P_n^{(m)}(\cos\theta) + (m-n-1) P_{n+1}^{(m)}(\cos\theta), \\
t_3 &= \frac{m \cos(m\phi) P_n^{(m)}(\cos\theta)}{r \sin\theta}, \\
t_4 &= c(n+1) y_n(kr) - \omega r y_{n+1}(kr), \text{ and}
\end{aligned}$$



(a) E_x **(b)** E_y

(c) E_z

Figure 4.6: Comparison of transient electric field components at $(x, y, z) = (1.01, 0.16, 0.98)$ for the problem of cube with prescribed magnetic loading on walls.

$$t_5 = c(n+1)j_n(kr) - \omega r j_{n+1}(kr).$$

For $m = n = 1$, we have the following limiting values:

$$\begin{aligned}
E_\theta|_{\theta\to 0} &= [t_6\cos(\omega t) - t_7\sin(\omega t)]\,\omega\sin\phi,\\
E_\phi|_{\theta\to 0} &= [t_6\cos(\omega t) - t_7\sin(\omega t)]\,\omega\cos\phi,\\
H_\theta|_{\theta\to 0} &= [t_8\cos(\omega t) + t_9\sin(\omega t)]\,t_{10}\cos\phi,\\
H_\phi|_{\theta\to 0} &= -[t_8\cos(\omega t) + t_9\sin(\omega t)]\,t_{10}\sin\phi,\\
E_\theta|_{\theta\to\pi} &= -E_\theta|_{\theta\to 0}, \qquad E_\phi|_{\theta\to\pi} = E_\phi|_{\theta\to 0},
\end{aligned}$$

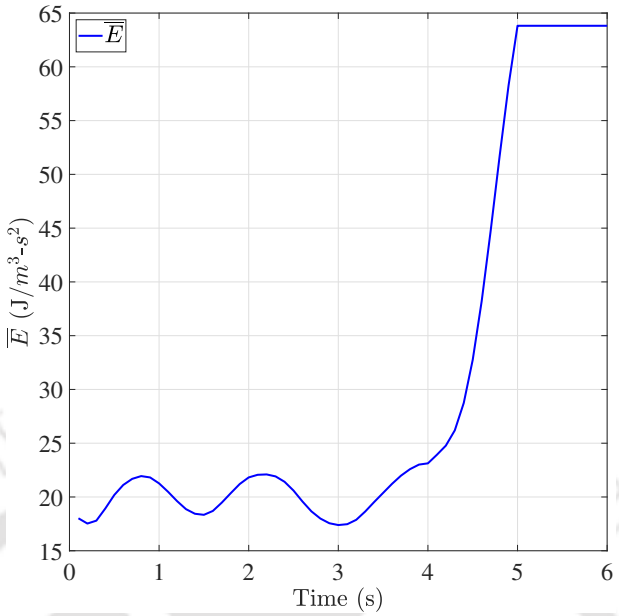


Figure 4.7: Conservation of $\bar{E}$ when there is no external loading i.e., $\boldsymbol{j} = \mathbf{0}$, and $\boldsymbol{H} \times \boldsymbol{n} = \mathbf{0}$ on boundary of the domain.

$$H_\theta|_{\theta\to\pi} = H_\theta|_{\theta\to 0}, \qquad H_\phi|_{\theta\to\pi} = -H_\phi|_{\theta\to 0},$$

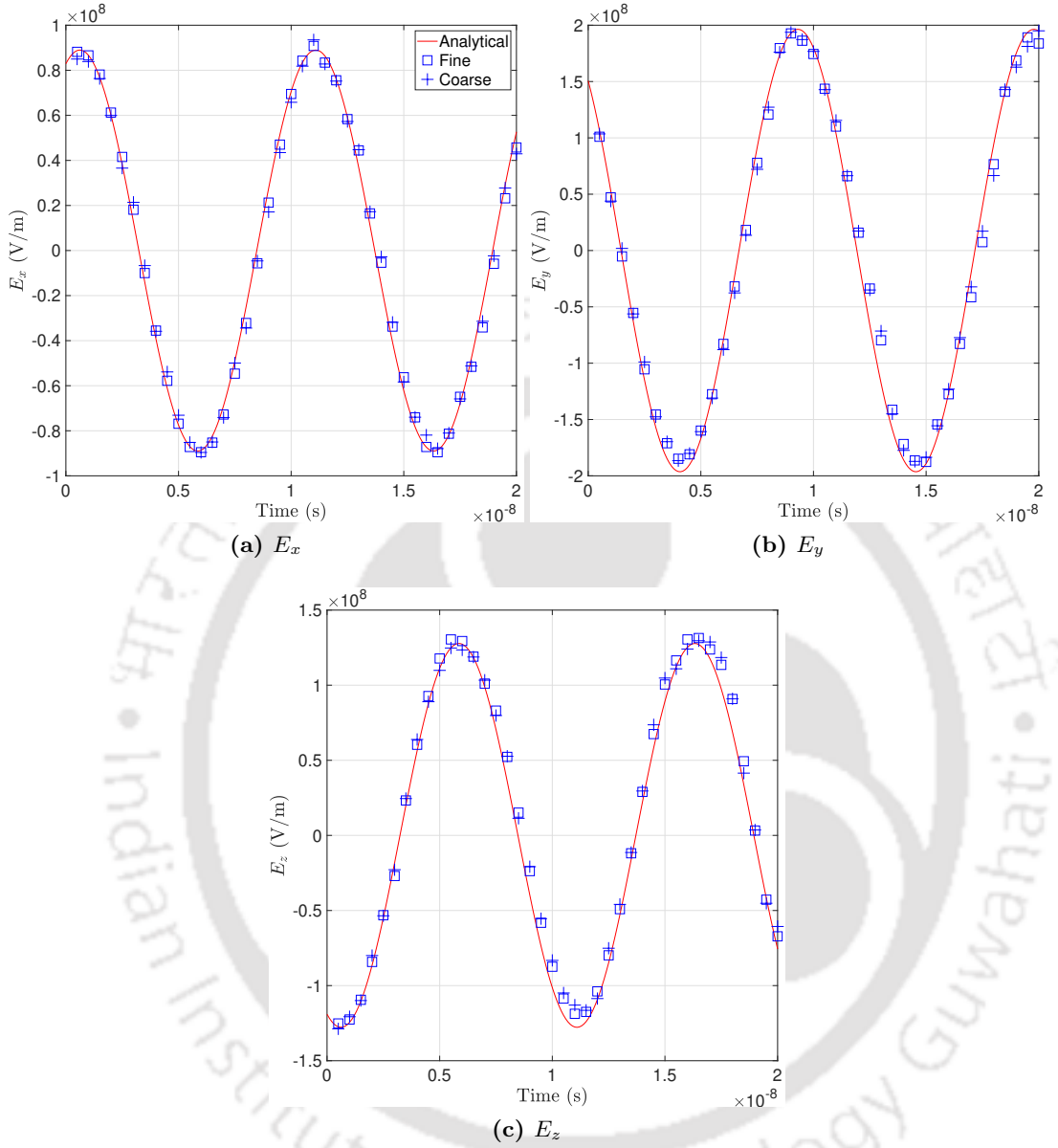
where

$$t_6 = \frac{2}{kr} y_1(kr) - y_2(kr), \qquad t_7 = \frac{2}{kr} j_1(kr) - j_2(kr),$$
$$t_8 = kr\cos(kr) - \sin(kr), \qquad t_9 = kr\sin(kr) + \cos(kr),$$
$$\text{and } t_{10} = \frac{\omega}{\mu c k^2 r^2}.$$

Table 4.4: Mesh details for the problem of sphere with transverse magnetic loading on boundary.

	No. of elements	No. of Equations
Coarse mesh	17397	22070
Fine mesh	28913	36152

In our simulation, we set parameters $ka = 2$, $kR_\infty = 6$, $\mu_r = 1$ and $\epsilon_r = 1$. For two different levels of mesh refinement, simulation of the electric field is carried out for a duration of 2×10^{-8} seconds, with a time-step of 1 nanosecond. Table 4.4 provides the information about the mesh details and resulting equations in our analysis. Fig. 4.8 and Fig. 4.9 illustrates the time response of different electric field components at two different locations within the computational domain. It is evident from these plots that FEM results are in good agreement with the analytical benchmark solutions.



(a) E_x **(b)** E_y

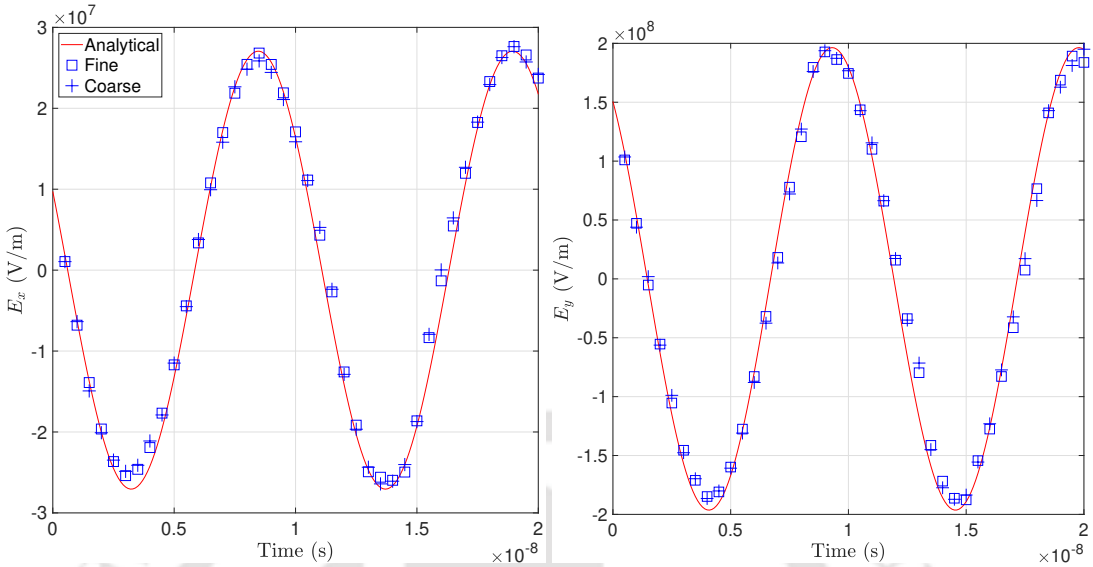
(c) E_z

Figure 4.8: Transient electric field at near field for $kr = 2.24, \theta = 47.5°, \phi = 21.5°$ for sphere with transverse magnetic loading.

4.3.4 Scattering from a conducting sphere

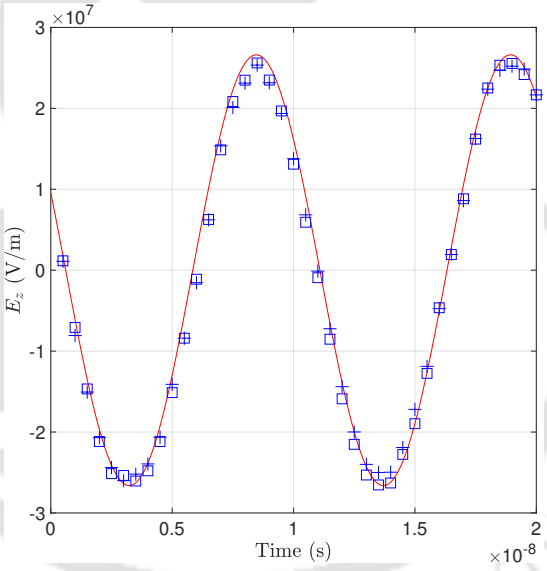
After solving three electromagnetic radiation problems in previous sections, here electromagnetic scattering analysis from a conducting sphere [94] is presented. A conducting sphere having radius of 0.8 m is subjected to an incident wave, expressed as

$$\boldsymbol{E}_{\text{inc}} = 2\{t - t_0 - \hat{\boldsymbol{k}} \cdot (\boldsymbol{r} - \boldsymbol{r}_0)/c\} \exp\left[-\frac{\{t - t_0 - \hat{\boldsymbol{k}} \cdot (\boldsymbol{r} - \boldsymbol{r}_0)/c\}^2}{\tau^2}\right] \hat{\boldsymbol{E}}, \tag{4.37}$$



(a) E_x

(b) E_y



(c) E_z

Figure 4.9: Transient electric field at far field for $kr = 4.4, \theta = 137°, \phi = -147°$ for sphere with transverse magnetic loading.

where the parameters $t_0, \boldsymbol{r}_0$ and τ denote the temporal reference point, spatial reference point and pulse shape, respectively.

Table 4.5: Mesh details in scattering from a conducting sphere.

	No. of elements	No. of Equations
Coarse mesh	27096	33455
Fine mesh	54236	67011

The computational domain is truncated at $R_\infty = 3$ m. Considering the parameters from [94] as $\hat{\boldsymbol{k}} = \hat{\boldsymbol{z}}$, $\hat{\boldsymbol{E}} = \hat{\boldsymbol{x}}$, $\boldsymbol{r}_0$=-1.2$\hat{z}$ m, $\epsilon_r = 1$, $\mu_r = 1$, $t_0 = 25.99$ ns and $\tau = 5.25$ ns, scattering analysis is conducted using two different meshes. Details of the numerical analysis is presented in Table 4.5. Time variation of $\frac{\partial E_x}{\partial t}$ values from coarse and fine meshes, obtained at $(0.33, -1.03, -0.45)$, is plotted in Fig. 4.10 along with the benchmark results [94]. It is clearly evident from Fig. 4.10 that FEM results for both the meshes are perfectly matching with the benchmark result.

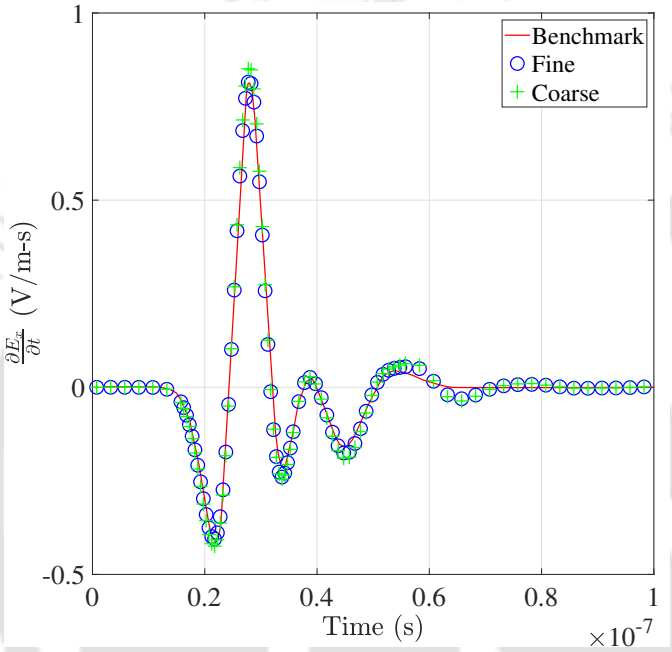


Figure 4.10: Comparison of variation of $\frac{\partial E_x}{\partial t}$ with time at $(0.33, -1.03, -0.45)$ for scattering from a conducting sphere.

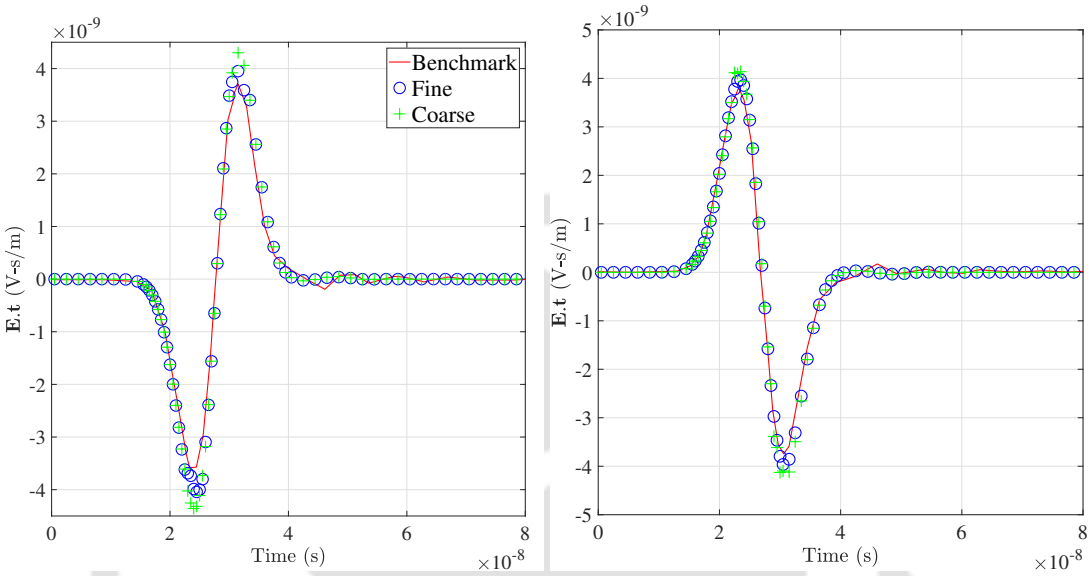
4.3.5 Scattering from a dielectric sphere

Table 4.6: Mesh details in scattering from a dielectric sphere.

	No. of elements	No. of Equations
Coarse mesh	17210	21757
Fine mesh	32251	40249

In this example, scattering from a dielectric sphere is analyzed where a sphere having radius of 0.5 m, having material properties $\mu_r = 1$ and $\epsilon_r = 6.0$, is exposed to Neumann pulse wave as described in Eq. (5.21). The pulse parameters of the wave are specified as $\hat{\boldsymbol{k}} = \hat{\boldsymbol{z}}$, $\hat{\boldsymbol{E}} = \hat{\boldsymbol{x}}$, $t_0 = 25.9$ ns, $\boldsymbol{r}_0 = -1.3\hat{\boldsymbol{z}}$ m and $\tau = 5.25$ ns. Here, the computational domain is truncated at $R_\infty = 2$ m, and the domain is modelled with two different level of meshes and time-step as 0.5 ns. Table 4.6 presents the mesh details and corresponding equations for both coarse and fine mesh. In Fig. 4.11, time variation of $\boldsymbol{E} \cdot \boldsymbol{t}$ for two

different $\boldsymbol{t}$ i.e., $\boldsymbol{t} = 0.96\hat{\boldsymbol{x}} + 0.26\hat{\boldsymbol{y}} + 0.13\hat{\boldsymbol{z}}$ at $(-0.04, -0.07, -0.72)$, and $\boldsymbol{t} = -0.89\hat{\boldsymbol{x}} - 0.08\hat{\boldsymbol{y}} + 0.44\hat{\boldsymbol{z}}$ at $(0.05, 0.05, -0.96)$ is plotted along with the benchmark results [147]. Once again, it is evident from the plot that FEM results for both coarse and fine mesh are in excellent match with the benchmark results from [147].



(a) $\boldsymbol{E}.\boldsymbol{t}$ at $(x, y, z) = (-0.04, -0.07, -0.72)$ with $\boldsymbol{t} = 0.96\hat{\boldsymbol{x}} + 0.26\hat{\boldsymbol{y}} + 0.13\hat{\boldsymbol{z}}$

(b) $\boldsymbol{E}.\boldsymbol{t}$ at $(x, y, z) = (0.05, 0.05, -0.96)$ with $\boldsymbol{t} = -0.89\hat{\boldsymbol{x}} - 0.08\hat{\boldsymbol{y}} + 0.44\hat{\boldsymbol{z}}$

Figure 4.11: Comparison of transient variation of $\boldsymbol{E} \cdot \boldsymbol{t}$ for scattering from a dielectric sphere.

4.4 Summary

In the present study, Maxwell's electromagnetic equation has been solved in time domain in edge finite element framework. One mathematical quantity in continuum framework is derived, which conserves in the absence of electromagnetic loading and under specific boundary conditions. Furthermore, a time-stepping strategy is proposed within edge element framework which mimic the conservation characteristic of the quantity under specific boundary and loading conditions. The proposed strategy is formulated to simulate electromagnetic radiation and scattering problems of both conducting and dielectric materials. To validate the proposed formulation, a series of numerical examples have been demonstrated. Very close agreement of the results obtained from the proposed strategy with the analytical and other benchmark results exhibit the efficiency and robustness of the proposed formulation.

Chapter 5

A novel implementation of Symmetric Boundary Condition in Harmonic and Transient analysis of electromagnetic wave propagation[‡]

While doing electromagnetic analysis using FEM (Finite element method), if we can implement the underlying symmetric nature of the problem, there will be significant reduction in the computational cost. Symmetric nature of the problem can be identified from the given physical loading and boundary conditions of the problem. But for electromagnetic analysis in potential formulation, it is not very straight forward to implement the symmetric boundary condition. In the present chapter, a novel implementation of symmetric boundary condition in potential formulation within nodal framework, has been demonstrated in form of thin-patch method. The implementation has been carried out in both electromagnetic harmonic and transient analysis for a wide range of radiation and scattering problems. The thin patch method also has been successfully extended to amplitude formulation.

The remaining chapter is arranged as follows: in section 5.1, we have briefly presented the mathematical and FEM formulation for both harmonic and transient electromagnetic analysis. We have presented the implementation of symmetric boundary condition in potential formulation in section 5.1.5. Thereafter, we have selected various electromagnetic radiation and scattering examples in both harmonic (section 5.2) and transient (section 5.3) analysis in such a way that there exist some plane of symmetry in those

[‡]D. Kamireddy, S. Karanam, and A. Nandy, "A Novel Implementation of Symmetric Boundary Condition in Harmonic and Transient Analysis of Electromagnetic Wave Propagation" (Under review). Available at: https://arxiv.org/abs/2212.04957

problems. We have implemented our novel method in those problems to demonstrate the computational efficacy of the proposed method as compared to the existing formulations. Finally, a detailed numerical analysis is conducted in section 5.4 to represent the requirement of having thin patch as compared to other dimensions of the computational domain in accurately predicting the electric fields.

5.1 Mathematical Formulation

In this section, after brief presentation of the mathematical and FEM formulation for harmonic and transient electromagnetic analysis using potential formulation, we elaborately present implementation of the symmetric boundary condition.

Maxwell's electromagnetic wave equation in terms of electric field $\boldsymbol{E}$ [108] can be written as

$$\frac{\epsilon_r}{c^2}\frac{\partial^2 \boldsymbol{E}}{\partial t^2} + \mu_0 \frac{\partial \boldsymbol{j}}{\partial t} + \boldsymbol{\nabla} \times \left(\frac{1}{\mu_r}\boldsymbol{\nabla} \times \boldsymbol{E}\right) = \boldsymbol{0}, \tag{5.1}$$

where $c = 1/\sqrt{\epsilon_0 \mu_0}$ is the speed of light and $\boldsymbol{j}$ is the current density. Relative permittivity, ϵ_r and relative permeability, μ_r are given as $\epsilon_r = \frac{\epsilon}{\epsilon_0}$ and $\mu_r = \frac{\mu}{\mu_0}$, where ϵ_0 and μ_0 are the electric permittivity and magnetic permeability for vacuum, and ϵ and μ are the electric permittivity and magnetic permeability of the medium, respectively.

5.1.1 Harmonic electromagnetic analysis

For harmonic excitation, Eq. 5.14 can be expressed as

$$\boldsymbol{\nabla} \times \left(\frac{1}{\mu}\boldsymbol{\nabla} \times \boldsymbol{E}\right) - \frac{k^2}{\mu}\boldsymbol{E} = -i\omega \boldsymbol{j}, \tag{5.2}$$

where $i = \sqrt{-1}$, ω is the excitation frequency, $k = k_0\sqrt{\mu_r \epsilon_r}$ is the wave number of the medium, and $k_0 = \omega/c$ is the wave number of the vacuum. For no charge condition, the electric field is subjected to the following constraint

$$\boldsymbol{\nabla} \cdot (\epsilon \boldsymbol{E}) = 0. \tag{5.3}$$

For perfectly conducting boundary (Γ_e), $\boldsymbol{E} \times \boldsymbol{n} = 0$, whereas for other part (Γ_h), $\boldsymbol{H} \times \boldsymbol{n}$ is prescribed. Assuming no surface current at material discontinuity, both $\boldsymbol{E} \times \boldsymbol{n}$ and $\boldsymbol{H} \times \boldsymbol{n}$ must be continuous across the material interface [1]. In Potential formulation, electric field $\boldsymbol{E}$ is replaced as $\boldsymbol{A} + \boldsymbol{\nabla}\psi$. Then, the governing differential equation given by Eq. (5.2), and the constraint equation given by Eq. (5.3), can be written as

$$\boldsymbol{\nabla} \times \left(\frac{1}{\mu}\boldsymbol{\nabla} \times \boldsymbol{A}\right) - \frac{k^2}{\mu}(\boldsymbol{A} + \boldsymbol{\nabla}\psi) = -i\omega \boldsymbol{j}, \tag{5.4a}$$

$$\boldsymbol{\nabla} \cdot (\epsilon \boldsymbol{A}) + \boldsymbol{\nabla} \cdot (\epsilon \boldsymbol{\nabla}\psi) = 0. \tag{5.4b}$$

In the variational statements of the Eqs. (5.4a) and (5.4b), we have to include a regularization term corresponding to the Coulomb gauge condition $\boldsymbol{\nabla} \cdot \boldsymbol{A} =$ constant, and the first-order absorbing boundary condition on truncated boundary (Γ_∞) [22]. Finally, the finite element equation can be expressed as

$$\begin{bmatrix} \boldsymbol{K}_{AA} & \boldsymbol{K}_{A\psi} \\ \boldsymbol{K}_{\psi A} & \boldsymbol{K}_{\psi\psi} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{A}} \\ \hat{\psi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_A \\ \boldsymbol{F}_\psi \end{bmatrix}. \tag{5.5}$$

Detailed derivation of the above FE equation are presented in [22].

From the expansion theorem of Wilcox [88, 148], for exterior domain electromagnetic problems, field $\boldsymbol{E}$ can be presented as

$$\boldsymbol{E} = \frac{e^{-ik|\boldsymbol{x}|}}{|\boldsymbol{x}|} \sum_{n=0}^{\infty} \frac{\boldsymbol{a}_n(\theta, \phi)}{|\boldsymbol{x}|^n}, \tag{5.6}$$

where $|\boldsymbol{x}|$ is the distance of a point from the origin, or, alternatively, the radial coordinate in the spherical coordinate system (r, θ, ϕ). Being inspired from this expansion theorem, Amplitude formulation for harmonic electromagnetic analysis has been proposed in [22]. Here, rapidly varying part $\frac{e^{-ik|\boldsymbol{x}|}}{|\boldsymbol{x}|}$ is separated out apriori from the potentials as

$$\boldsymbol{A} = \frac{1}{|\boldsymbol{x}|} \bar{\boldsymbol{A}}(\boldsymbol{x}) e^{-ik|\boldsymbol{x}|}, \tag{5.7a}$$

$$\psi = \frac{1}{|\boldsymbol{x}|} \bar{\psi}(\boldsymbol{x}) e^{-ik|\boldsymbol{x}|}. \tag{5.7b}$$

FEM has to capture a relatively gentle variation of $\bar{\boldsymbol{A}}$ and $\bar{\psi}$. Finally the resulting finite element equations are given as

$$\begin{bmatrix} \boldsymbol{K}_{AA} & \boldsymbol{K}_{A\psi} \\ \boldsymbol{K}_{\psi A} & \boldsymbol{K}_{\psi\psi} \end{bmatrix} \begin{bmatrix} \hat{\bar{\boldsymbol{A}}} \\ \hat{\bar{\psi}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_A \\ \boldsymbol{F}_\psi \end{bmatrix}. \tag{5.8}$$

Different terms in the above equations have been derived elaborately in [22].

5.1.2 Transient electromagnetic analysis

For transient electromagnetic analysis using potential formulation [24], electric and magnetic fields are expressed in terms of $\boldsymbol{v} := \partial \boldsymbol{A}/\partial t$ and $\boldsymbol{w} := \partial(\boldsymbol{\nabla}\psi)/\partial t$ as

$$\boldsymbol{E} = -\boldsymbol{w} - \boldsymbol{v}, \tag{5.9}$$

$$\boldsymbol{H} = \frac{1}{\mu} \boldsymbol{\nabla} \times \boldsymbol{A}. \tag{5.10}$$

Then, Eq. 5.14 and Eq. 5.3 are expressed as

$$\epsilon \left[\frac{\partial^2 \boldsymbol{A}}{\partial t^2} + \boldsymbol{\nabla} \left(\frac{\partial^2 \psi}{\partial t^2} \right) \right] + \frac{\partial \boldsymbol{j}}{\partial t} + \boldsymbol{\nabla} \times \left(\frac{1}{\mu} \boldsymbol{\nabla} \times \boldsymbol{A} \right) = \boldsymbol{0}, \tag{5.11}$$

$$\boldsymbol{\nabla} \cdot \left[\epsilon \left(\boldsymbol{\nabla} \frac{\partial \psi}{\partial t} + \frac{\partial \boldsymbol{A}}{\partial t} \right) \right] = 0. \tag{5.12}$$

In transient analysis choice of time stepping strategy is very crucial to conserve certain physical quantities with a view to make our numerical formulation unconditionally stable. After following the detailed derivations as depicted in [24], the finite element equation can be written as

$$\begin{bmatrix} \boldsymbol{K}_{AA} & \boldsymbol{K}_{A\psi} \\ \boldsymbol{K}_{\psi A} & \boldsymbol{K}_{\psi\psi} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{A}}_{n+1} \\ \hat{\boldsymbol{\Psi}}_{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_A \\ \boldsymbol{f}_\psi \end{bmatrix}. \tag{5.13}$$

5.1.3 Application of the symmetric boundary condition in electromagnetic analysis

Let us consider that for an electromagnetic radiation or scattering problem, the geometry of physical domain, say Ω, and physical loading exhibit a symmetry about a plane Γ. Then, the domain Ω can be separated into two sub-domains i.e., Ω^+ or Ω^- which are symmetric with respect to the symmetric plane Γ. As in the electromagnetic analysis, normal electric field to the symmetric plane Γ should be zero, we have $\boldsymbol{E} \cdot \boldsymbol{n} = 0$ on Γ. Here, $\boldsymbol{n}$ represents the outward normal vector to Γ considering either Ω^+ or Ω^- as computational domain. Therefore, to solve the electric field for entire domain Ω, it is sufficient to consider the electromagnetic problem within Ω^+ (or Ω^-), and ensure that the boundary condition $\boldsymbol{E} \cdot \boldsymbol{n} = 0$ is satisfied on Γ.

In nodal finite element analysis, potential formulation is used in solving harmonic and time domain problems. Here, the symmetric boundary condition is implemented as $(\boldsymbol{A} \cdot \boldsymbol{n} + \boldsymbol{\nabla}\psi \cdot \boldsymbol{n}) = 0$ on Γ. We implemented this condition in two different novel methods: (1) Adding a Lagrange multiplier λ to the condition $\boldsymbol{E} \cdot \boldsymbol{n} = 0$ on Γ, (2) Adding a thin patch to the symmetric surface Γ. The condition $(\boldsymbol{A} \cdot \boldsymbol{n} + \boldsymbol{\nabla}\psi \cdot \boldsymbol{n}) = 0$ is implemented explicitly on the nodes of the finite elements created in the patch.

Based on our numerical experimentations we found that for the same mesh resolution, the former method could not accurately predict the desired electric fields whereas the thin patch method successfully simulate various harmonic and transient problems. In the following sections, we elaborate on the implementation of both the methods: Lagrange multiplier method and the thin patch method using a specific example.

5.1.4 Implementation of the symmetric boundary condition using Lagrange multiplier method

Consider the condition

$$\int \lambda(\boldsymbol{E} \cdot \boldsymbol{n})\, d\Gamma = \boldsymbol{0}, \tag{5.14}$$

where λ is Lagrange multiplier. From potential formulation, electric field ($\boldsymbol{E}$) can be expressed as $\boldsymbol{E} = \boldsymbol{A} + \boldsymbol{\nabla}\psi$. Therefore, Eq. 5.14 can be written as

$$\int \lambda \left[(\boldsymbol{A} + \boldsymbol{\nabla}\psi) \cdot \boldsymbol{n} \right] d\Gamma = \boldsymbol{0}. \tag{5.15}$$

Applying variational operator to the above Eq. 5.15, we get as

$$\int \left(\lambda_\delta \left(\boldsymbol{A} \cdot \boldsymbol{n} + \boldsymbol{\nabla}\psi \cdot \boldsymbol{n} \right) + \lambda \left(\boldsymbol{A}_\delta \cdot \boldsymbol{n} + \boldsymbol{\nabla}\psi_\delta \cdot \boldsymbol{n} \right) \right) d\Gamma = \boldsymbol{0}. \tag{5.16}$$

Let the fields and their variations in the Eq. 5.16 be interpolated as

$$\begin{aligned}
&\boldsymbol{A} = \boldsymbol{N}\hat{\boldsymbol{A}}, && \boldsymbol{A}_\delta = \boldsymbol{N}\hat{\boldsymbol{A}}_\delta, && \boldsymbol{A} \cdot \boldsymbol{n} = \boldsymbol{n}^T \boldsymbol{N}\hat{\boldsymbol{A}}, && \boldsymbol{A}_\delta \cdot \boldsymbol{n} = \boldsymbol{n}^T \boldsymbol{N}\hat{\boldsymbol{A}}_\delta, \\
&\psi = \boldsymbol{N}_\psi \hat{\boldsymbol{\psi}}, && \psi_\delta = \boldsymbol{N}_\psi \hat{\boldsymbol{\psi}}_\delta, && \boldsymbol{\nabla}\psi = \boldsymbol{B}_\psi \hat{\boldsymbol{\psi}}, && \boldsymbol{\nabla}\psi_\delta = \boldsymbol{B}_\psi \hat{\boldsymbol{\psi}}_\delta, \\
&\lambda = \boldsymbol{N}_\lambda \hat{\boldsymbol{\lambda}}, && \lambda_\delta = \boldsymbol{N}_\lambda \hat{\boldsymbol{\lambda}}_\delta,
\end{aligned}$$

where

$$\boldsymbol{N} = \begin{bmatrix} N_1 & 0 & 0 & N_1 & 0 & 0 & \dots \\ 0 & N_2 & 0 & 0 & N_2 & 0 & \dots \\ 0 & 0 & N_3 & 0 & 0 & N_3 & \dots \end{bmatrix}, \qquad \boldsymbol{N}_\psi = \begin{bmatrix} N_1 & N_2 & N_3 & \dots \end{bmatrix},$$

$$\boldsymbol{N}_\lambda = \begin{bmatrix} N_1 & N_2 & N_3 & \dots \end{bmatrix}, \qquad \boldsymbol{B}_\psi = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \dots \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \dots \\ \frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial z} & \dots \end{bmatrix}.$$

Here, $\boldsymbol{n}$ is the unit normal to the surface parameterized by natural coordinates (ξ, η) which has three components n_1, n_2 and n_3 and can be expressed as

$$\boldsymbol{n} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix}.$$

The individual terms in Eq. 5.16 are interpolated and can be expressed as

$$\begin{aligned}
\lambda_\delta(\boldsymbol{A} \cdot \boldsymbol{n}) &= \hat{\boldsymbol{\lambda}_\delta}^T \boldsymbol{N}_\lambda{}^T \boldsymbol{n}^T \boldsymbol{N}\hat{\boldsymbol{A}}, & \lambda(\boldsymbol{A}_\delta \cdot \boldsymbol{n}) &= \hat{\boldsymbol{A}_\delta}^T \boldsymbol{N}^T \boldsymbol{n} \boldsymbol{N}_\lambda \hat{\boldsymbol{\lambda}}, \\
\lambda_\delta(\boldsymbol{\nabla}\psi \cdot \boldsymbol{n}) &= \hat{\boldsymbol{\lambda}_\delta}^T \boldsymbol{N}_\lambda{}^T \boldsymbol{n}^T \boldsymbol{B}_\psi \hat{\psi}, & \lambda(\boldsymbol{\nabla}\psi_\delta \cdot \boldsymbol{n}) &= \hat{\boldsymbol{\psi}_\delta}^T \boldsymbol{B}_\psi{}^T \boldsymbol{n} \boldsymbol{N}_\lambda \hat{\boldsymbol{\lambda}}.
\end{aligned}$$

In harmonic analysis the Eq. 5.16 is added to the variational statements of the Eqs. (5.4a) and (5.4b). Then, the finite element equation for harmonic analysis can be expressed as

$$\begin{bmatrix} \boldsymbol{K}_{AA} & \boldsymbol{K}_{A\psi} & \boldsymbol{K}_{A\lambda} \\ \boldsymbol{K}_{\psi A} & \boldsymbol{K}_{\psi\psi} & \boldsymbol{K}_{\psi\lambda} \\ \boldsymbol{K}_{\lambda A} & \boldsymbol{K}_{\lambda\psi} & \boldsymbol{K}_{\lambda\lambda} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{A}} \\ \hat{\psi} \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_A \\ \boldsymbol{F}_\psi \\ 0 \end{bmatrix}, \tag{5.17}$$

where

$$\boldsymbol{K}_{AA} = \int_\Omega \frac{1}{\mu}\left[\boldsymbol{B}^T\boldsymbol{B} + \boldsymbol{B}_p^T\boldsymbol{B}_p - k^2\boldsymbol{N}^T\boldsymbol{N}\right] d\Omega + i\int_{\Gamma_\infty} \frac{k}{\mu}\boldsymbol{N}^T\boldsymbol{n}_{\text{mat}}\boldsymbol{n}_{\text{mat}}^T\boldsymbol{N}\, d\Gamma,$$

$$\boldsymbol{K}_{A\psi} = -\int_\Omega \frac{k^2}{\mu}\boldsymbol{N}^T\boldsymbol{B}_\psi\, d\Omega,$$

$$\boldsymbol{K}_{\psi A} = \int_\Omega \epsilon\boldsymbol{B}_\psi^T\boldsymbol{N}\, d\Omega - \int_{\Gamma_\infty} \epsilon\boldsymbol{N}_\psi^T\boldsymbol{n}^T\boldsymbol{N}\, d\Gamma - \int_{\Gamma_h} \epsilon\boldsymbol{N}_\psi^T\boldsymbol{n}^T\boldsymbol{N}\, d\Gamma,$$

$$\boldsymbol{K}_{\psi\psi} = \int_\Omega \epsilon\boldsymbol{B}_\psi^T\boldsymbol{B}_\psi\, d\Omega + \int_{\Gamma_\infty} \frac{\epsilon}{R}\boldsymbol{N}_\psi^T\boldsymbol{N}_\psi\, d\Gamma - \int_{\Gamma_h} \epsilon\boldsymbol{N}_\psi^T\boldsymbol{n}^T\boldsymbol{B}_\psi\, d\Gamma,$$

$$\boldsymbol{K}_{A\lambda} = \int \boldsymbol{N}^T\boldsymbol{n}\boldsymbol{N}_\lambda\, d\Gamma, \qquad \boldsymbol{K}_{\psi\lambda} = \int {\boldsymbol{B}_\psi}^T\boldsymbol{n}\boldsymbol{N}_\lambda\, d\Gamma, \qquad \boldsymbol{K}_{\lambda A} = \int \boldsymbol{N}_\lambda^T\boldsymbol{n}^T\boldsymbol{N}\, d\Gamma,$$

$$\boldsymbol{K}_{\lambda\psi} = \int {\boldsymbol{N}_\lambda}^T\boldsymbol{n}^T\boldsymbol{B}_\psi\, d\Gamma, \qquad \boldsymbol{K}_{\lambda\lambda} = \boldsymbol{0},$$

$$\boldsymbol{F}_A = -i\omega\left(\int_{\Gamma_h} \boldsymbol{N}^T\bar{\boldsymbol{H}}\, d\Gamma + \int_\Omega \boldsymbol{N}^T\boldsymbol{j}\, d\Omega\right), \qquad \boldsymbol{F}_\psi = \boldsymbol{0}.$$

On imposing the symmetry boundary condition the Eq. 5.5 in harmonic electromagnetic analysis is modified into Eq. 5.17. In harmonic analysis the terms in Eq. 5.17 $\boldsymbol{K}_{AA}$, $\boldsymbol{K}_{A\psi}$, $\boldsymbol{K}_{\psi A}$, and $\boldsymbol{K}_{\psi\psi}$ are same as of terms in [22].

5.1.5 Implementation of the symmetric boundary condition using a thin patch

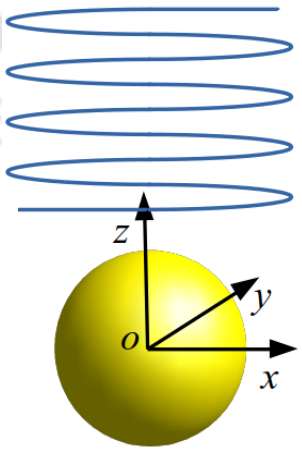


Figure 5.1: Plane electromagnetic wave incident on a conducting sphere.

Let us first understand the existence of symmetry in electromagnetic analysis with one example. Consider one conducting sphere whose center is located at the origin as

shown in Fig. 5.1. One electromagnetic plane wave, $\boldsymbol{E} = E_x e^{-ikz}\hat{\boldsymbol{e}}_x$, impinges on the sphere. The geometry of the sphere is symmetric about the origin. The incident wave has field component in x-direction and the wave is travelling from z-direction. Thus, the incident wave is symmetric about the xz-plane. There will be electromagnetic scattering by the conducting sphere. As the geometry and the loading condition (incident wave) is symmetric about the xz-plane, the scattered electromagnetic field also will be symmetric about the xz-plane.

When there exists some plane of symmetry, we can consider half of the domain on any one side of the symmetry plane for our numerical analysis. In that case, one additional boundary surface gets created in the new half domain in form of the symmetric plane. Proper boundary condition should be applied on that plane.

Because of symmetry, always normal electric field to the symmetric plane should be zero. This is the required boundary condition on the symmetric plane for electromagnetic analysis. Thus, for the above example, on the symmetric plane (xz-plane), we have to satisfy $E_y = 0$. In potential formulation, it is not very straight forward to satisfy $\boldsymbol{E} \cdot \boldsymbol{n} = 0$ as we have to eventually satisfy $(\boldsymbol{A} \cdot \boldsymbol{n} + \boldsymbol{\nabla}\psi \cdot \boldsymbol{n}) = 0$.

In this work, we have proposed a novel method of implementation of the symmetric boundary condition, $(\boldsymbol{A}\cdot\boldsymbol{n}+\boldsymbol{\nabla}\psi\cdot\boldsymbol{n}) = 0$. Here, a thin patch of finite elements is attached to the symmetric plane, which extend our numerical domain. The patch thickness t should be very small as compared to the other dimensions of the domain. For this extended domain, on the boundary surface which is parallel to the symmetric plane, $\boldsymbol{A} \cdot \boldsymbol{n} = 0$ is implemented explicitly on all the surface nodes. In addition to that, for all the volume nodes inside this thin patch, $\psi = 0$ is applied explicitly. As the patch is very thin, it will result in $\boldsymbol{\nabla}\psi \cdot \boldsymbol{n} = 0$ on the symmetric plane.

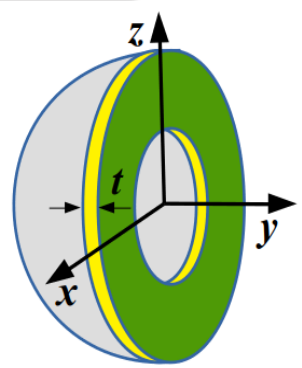


Figure 5.2: Sphere with a thin cylindrical patch of thickness, t.

In the context of our example, the full FEM analysis domain is one hollow sphere. After considering the existing symmetry of the problem, we can take hollow hemisphere as our computational domain. The flat face of the hollow hemisphere is the symmetric xz-plane. As shown in Fig. 5.2, one thin yellow patch of finite elements are attached to the symmetric flat face. On the green surface which is parallel to the original symmetric

plane, $A_y = 0$ is imposed for all the surface nodes. For all the nodes inside the yellow volume of the cylindrical patch, $\psi = 0$ is applied so that we have $\boldsymbol{\nabla}\psi \cdot \boldsymbol{n} = 0$ on the symmetric face.

In different examples, we have shown that the original converged solution field of the full domain is accurately predicted by the half domain with this novel implementation. Detailed numerical analysis has been done, and electric fields are compared for both full and half domains along with the benchmark values. In each and every case, results of both the domains are perfectly matching with the analytical/ benchmark values. Moreover, it is also observed that in all the cases, with adoption of the present method, electric fields are simulated efficiently with less number of unknowns as compared to the full domain analysis. Thus, the proposed method can be computationally efficient in electromagnetic analysis. We have implemented the proposed method in both harmonic and transient electromagnetic analysis. The proposed method has also been extended in amplitude formulation [22] for harmonic electromagnetic analysis.

5.2 Numerical Examples with Electromagnetic Harmonic Analysis

In this section, we have presented the efficacy of the proposed method of implementing symmetric boundary condition, while solving various examples using potential formulation in harmonic analysis. 27 node hexahedral brick elements (B27) and 18 node wedge elements (W18) are used. The results obtained from the implementation of the symmetric boundary condition in the half domain, are compared with the full domain results, and the analytical benchmark results. To assess the accuracy of the proposed method in predicting the electric field throughout the entire domain, L_2-norm is calculated between the simulation results ($\boldsymbol{E}_{\text{FEM}}$) obtained from the half domain with thin patch and analytical benchmark values ($\boldsymbol{E}_{\text{Analytical}}$) as

$$L_2\text{-norm error} = \frac{1}{N}\sqrt{\sum_{i=1}^{N}\left[\sum_{j=x,y,z}\left(\frac{E_{j_{(\text{FEM})_i}} - E_{j_{(\text{Analytical})_i}}}{E_{j_{(\text{Analytical})_i}}}\right)^2\right]} \tag{5.18}$$

where N is the total number of nodes of the discretized domain.

5.2.1 Scattering from a conducting sphere

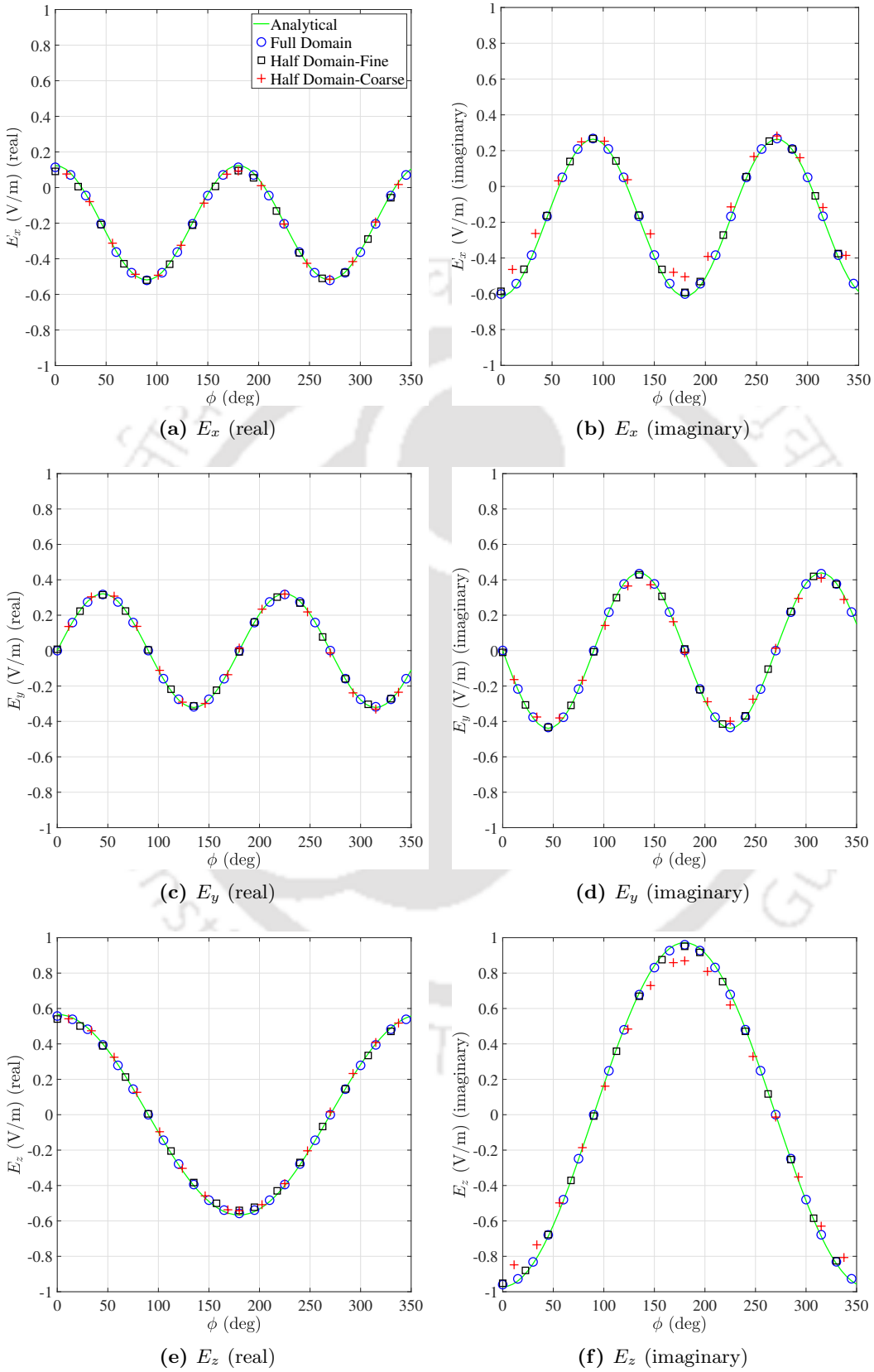
As the first example, we have solved scattering problem from one conducting sphere [22]. Here, one electromagnetic plane wave, $\boldsymbol{E}_{\text{inc}} = E_0 e^{-ikz}\hat{\boldsymbol{e}}_x$, impinges on a conducting sphere of radius a. To implement FEM, our computational domain (air surrounding the conducting sphere) is one hollow sphere which is truncated at radius R_∞. On this truncated surface, we apply Sommerfeld absorbing boundary condition. Conducting boundary condition is imposed on the inner spherical surface at radius a. Our interest

is to find the scattered electric field in the hollow spherical air domain having inner and outer radii of a and R_∞ respectively. From the discussion in section 5.1.5, we can understand that the problem is symmetric about the xz-plane. Therefore, we consider half of the hollow sphere on one side of the xz plane as our computational domain, and we apply symmetric boundary condition on the symmetry face (xz) using two approaches as discussed in sections 5.1.4 and 5.1.5. Considering the results in [22], we discretize the hemispherical domain with two different level of mesh sizes. The coarse mesh consists of $8 \times 6 \times 8$ B27 and W18 elements, whereas the fine mesh has $16 \times 12 \times 12$ B27 and W18 elements. As discussed in section 5.1.5, the symmetric boundary condition is implemented with a thin cylindrical patch having $16 \times 12 \times 1$ ($r \times \theta \times t$) B27 elements in case of fine mesh, whereas for coarse mesh we have used $8 \times 12 \times 1$ elements. Similarly, as discussed in section 5.1.4, the symmetric boundary condition is implemeneted implicitly for the hemispherical domain and solved for electric fields using both coarse and fine meshes. In Table 5.1, mesh details along with the total number of equations, are presented for both full domain (existing formulation) and half domain with the thin patch, as well as half domain with Lagrange multiplier (current formulation). We have considered $k_0 a = 1$, $k_0 R_\infty = 5$ with $E_0 = 1$.

Table 5.1: Mesh and equation details for scattering from a conducting sphere.

	Half domain with thin patch method		Half domain with Lagrange multiplier method		Full domain [22]
	Coarse mesh	Fine mesh	Coarse mesh	Fine mesh	
Mesh size ($r \times \theta \times \phi$) + ($r \times \theta \times t$)	$8 \times 6 \times 8$ + $8 \times 12 \times 1$	$16 \times 12 \times 12$ + $16 \times 24 \times 1$	$8 \times 6 \times 8$	$16 \times 12 \times 12$	$16 \times 12 \times 24$
Number of equations	13689	80125	9585	75133	106858

Fig. 5.3 and Fig. 5.4 represent variations along ϕ for all three components of the scattered electric field obtained using thin patch method at $\theta = \pi/4$, at near field ($k_0 r = 1.25$) and at far field ($k_0 r = 4.75$) respectively. Likewise, for the Lagrange multiplier method, Figs. 5.5 and 5.6 depict the electric field variations in the near and far fields, respectively. During numerical analysis for the half domain, we directly solve for $0 \leq \phi \leq \pi$. For the remaining half domain, we can use $\boldsymbol{E}(\pi - \phi) = \boldsymbol{E}(\phi)$ due to symmetry.

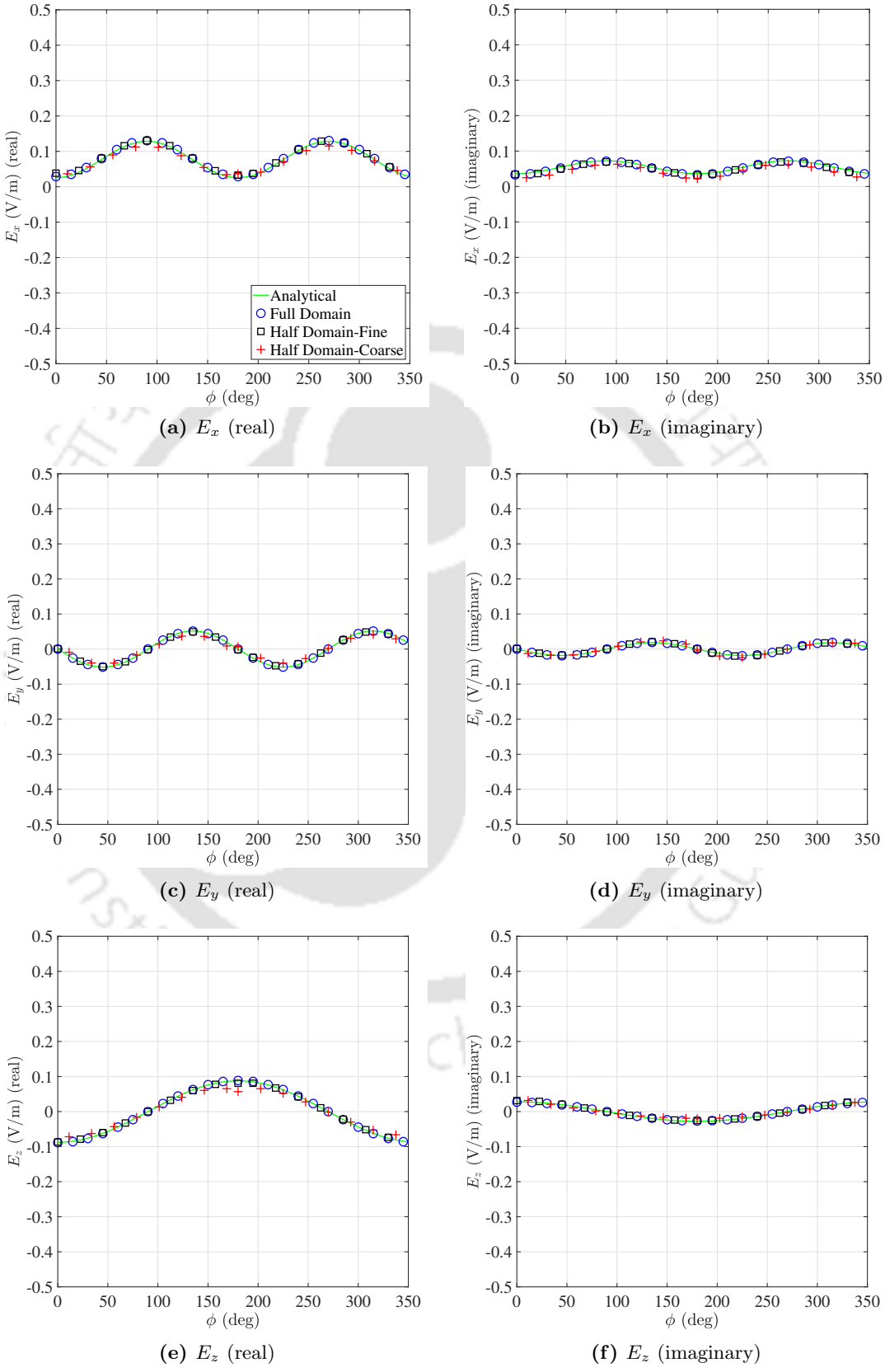


(a) E_x (real) **(b)** E_x (imaginary)

(c) E_y (real) **(d)** E_y (imaginary)

(e) E_z (real) **(f)** E_z (imaginary)

Figure 5.3: Near field field variation along ϕ in the electric field for parameters $k_0 r = 1.25$, $\theta = \pi/4$, $k_0 a = 1$, $k_0 R_\infty = 5$ for the scattering from a conducting sphere using thin patch method.

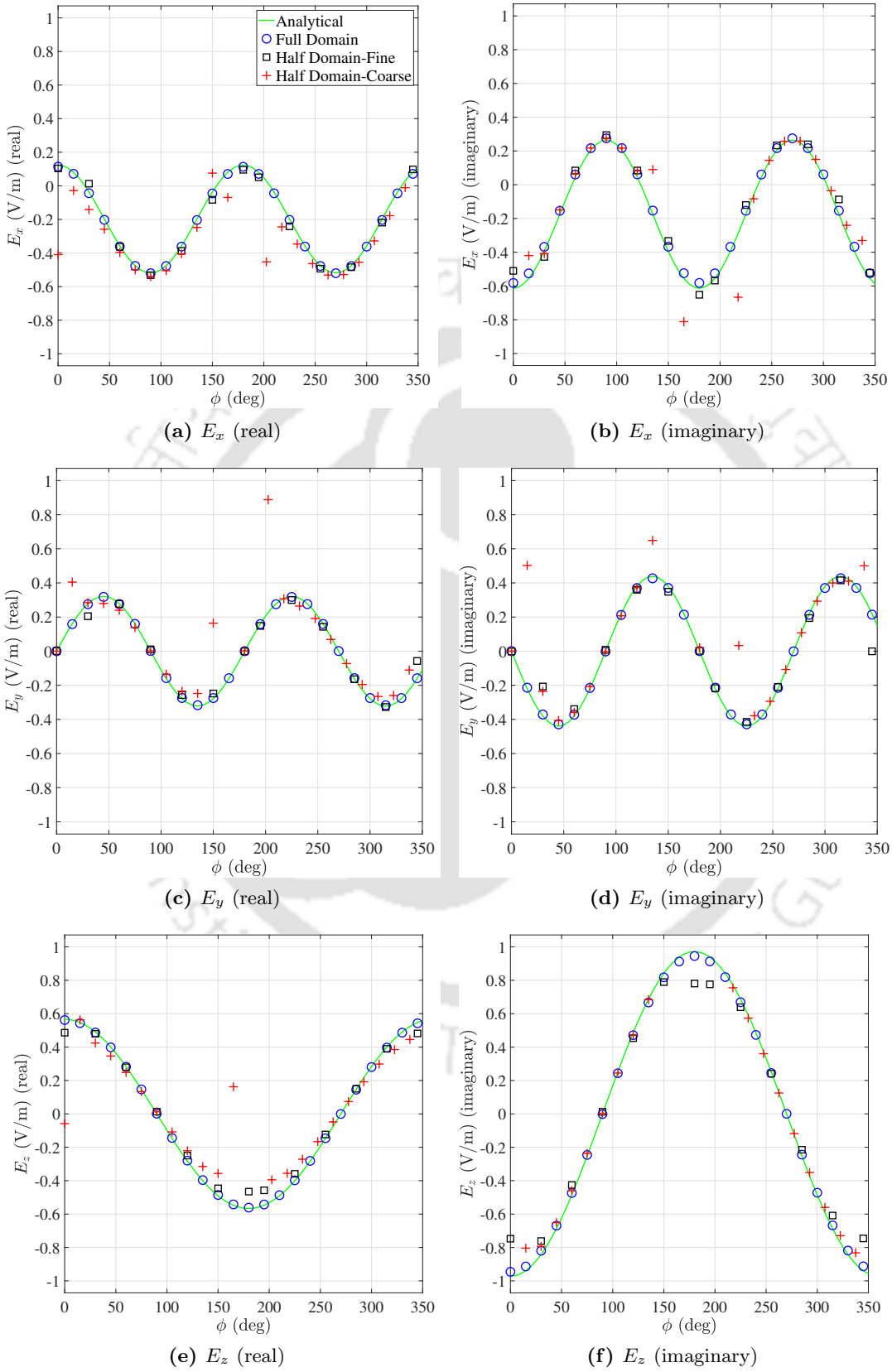


(a) E_x (real) **(b)** E_x (imaginary)

(c) E_y (real) **(d)** E_y (imaginary)

(e) E_z (real) **(f)** E_z (imaginary)

Figure 5.4: Far field variation along ϕ in the electric field for parameters $k_0 r = 4.75$, $\theta = \pi/4$, $k_0 a = 1$, $k_0 R_\infty = 5$ for the scattering from a conducting sphere using thin patch method.

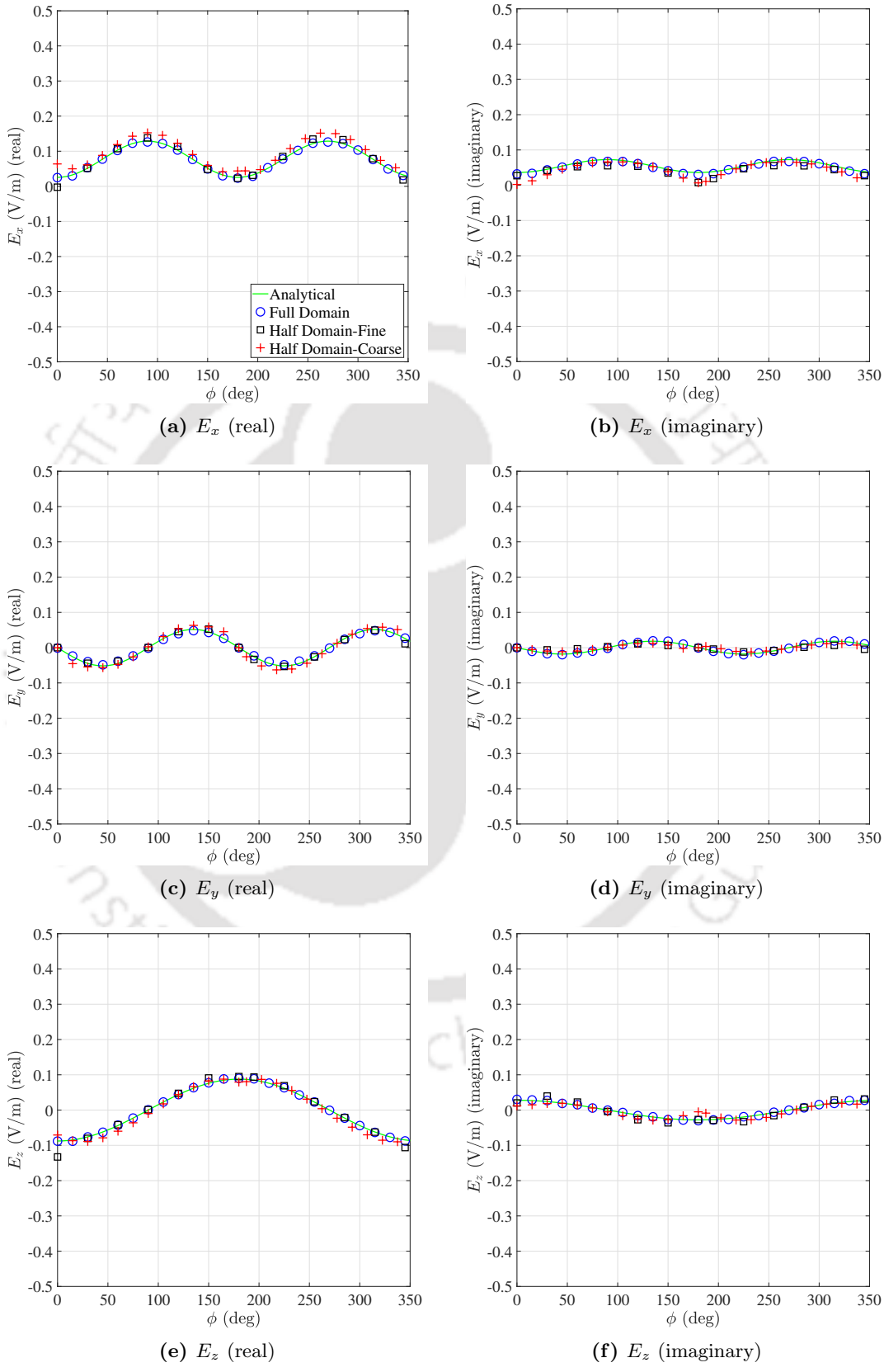


(a) E_x (real) **(b)** E_x (imaginary)

(c) E_y (real) **(d)** E_y (imaginary)

(e) E_z (real) **(f)** E_z (imaginary)

Figure 5.5: Near field field variation along ϕ in the electric field for parameters $k_0 r = 1.25$, $\theta = \pi/4$, $k_0 a = 1$, $k_0 R_\infty = 5$ for the scattering from a conducting sphere using Lagrange mulitplier method.



(a) E_x (real) **(b)** E_x (imaginary)

(c) E_y (real) **(d)** E_y (imaginary)

(e) E_z (real) **(f)** E_z (imaginary)

Figure 5.6: Far field variation along ϕ in the electric field for parameters $k_0 r = 4.75$, $\theta = \pi/4$, $k_0 a = 1$, $k_0 R_\infty = 5$ for the scattering from a conducting sphere using Lagrange mulitplier method.

From our analysis, we observe 3.65 % error in the half domain results with fine mesh (from thin patch method), while solving 25 % less equations than the equivalent full domain. Whereas, the coarse mesh requires 87.1 % less equations to predict the numerical results with a 15.95 % error. Fig. 5.3 and Fig. 5.4 indicate that with both coarse and fine meshes, numerical results are converging with analytical values. From Figs. 5.5 and 5.6, it can be observed that the thin patch method produces more accurate result than the Lagrange multiplier method.

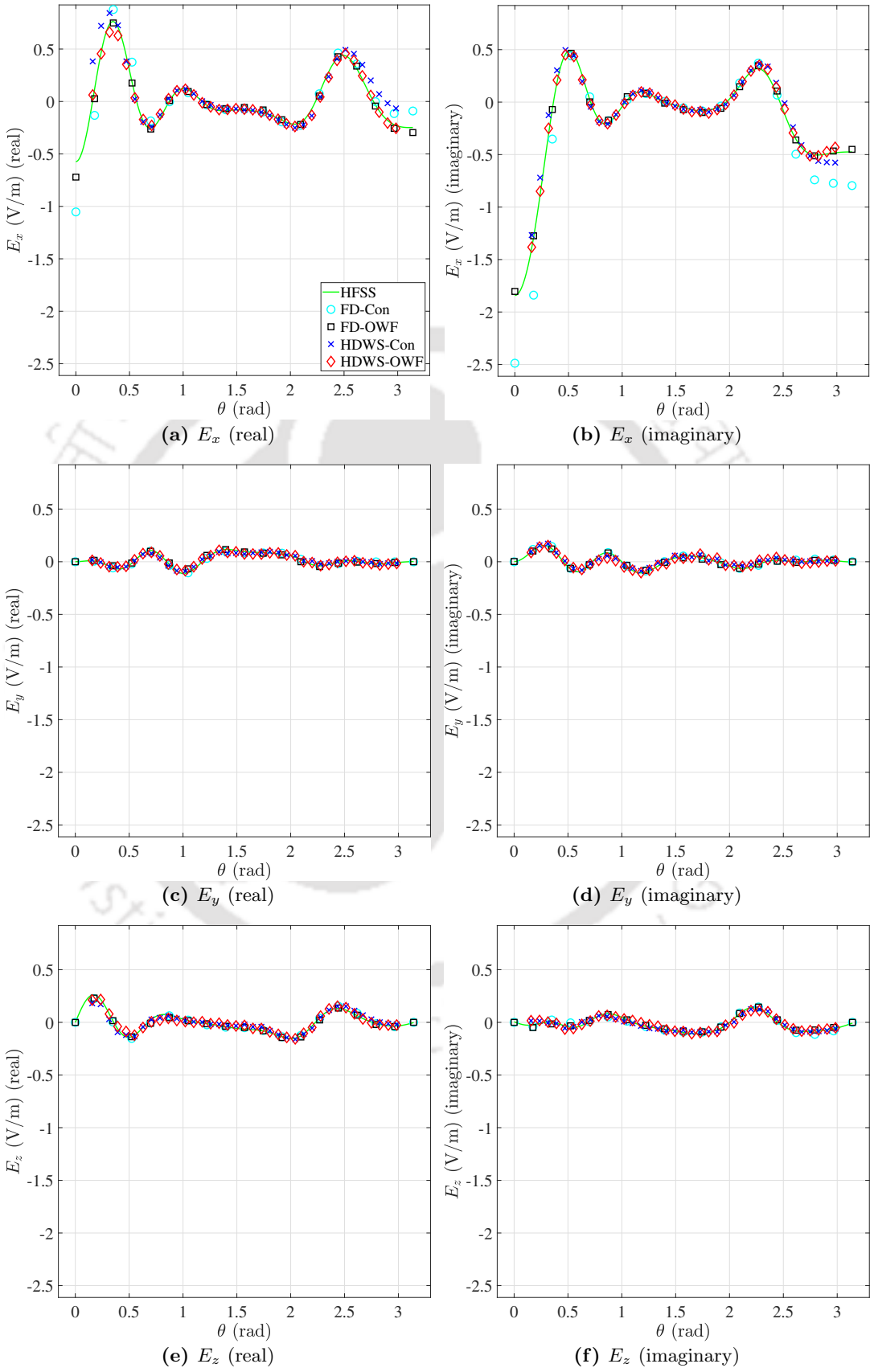
5.2.2 Scattering from a conducting ellipsoid

This example is chosen to show that the proposed implementation of symmetric boundary condition also works for non spherical geometry. A wave of the form $e^{-ik_0 z}\boldsymbol{e}_x$ impinges on one conducting ellipsoid having diameters $2a$, $2a$, and $2c$ along X, Y and Z axes, respectively. The domain is truncated at a spherical surface given by $k_0 R_\infty = 24$. We present our results for $k_0 a = 12$ and $c = a/4$. In this problem, the symmetry boundary condition $\boldsymbol{E} \cdot \boldsymbol{n} = \boldsymbol{0}$ is implemented as described in section 5.1.5. The half annular ellipsoidal domain is meshed with $12 \times 20 \times 8$ $(r \times \theta \times \phi)$ B27 and W18 elements and an additional cylindrical patch has $12 \times 40 \times 1$ $(r \times \theta \times t)$ B27 elements. Then we have to solve 73,949 equations. Table 5.2 summarizes the mesh details of half domain along with the full domain.

Table 5.2: Mesh and equation details for scattering from a conducting ellipsoid.

	Half domain	Full domain [22]
Mesh size $(r \times \theta \times \phi)+$ $(r \times \theta \times t)$	$12 \times 20 \times 8 +$ $12 \times 40 \times 1$	$12 \times 20 \times 16$
Number of equations	73949	123018

We have taken $E_0 = 1, k_0 a = 12, k_0 R_\infty = 24$. The scattered electric field components are compared in Fig. 5.7 along θ for $k_0 r = 24$ and $\phi = \frac{\pi}{4}$. We have compared the results of the half domain with symmetry (HDWS) in both conventional (Con) and amplitude formulation (OWF) with the converged HFSS [149] result obtained using a very fine mesh (1,49,530 unknowns). In Fig. 5.7, we have also presented the respective full domain result for both Con and OWF formulations. In the current half domain analysis, we directly solve for $0 \leq \phi \leq \pi$, and because of symmetry we have $\boldsymbol{E}(2\pi - \phi) = \boldsymbol{E}(\phi)$ for the remaining half domain. We obtain equivalent accuracy with half domain (thin patch) solving a system of equations having 40% less unknowns than that of full domain.



(a) E_x (real) **(b)** E_x (imaginary)

(c) E_y (real) **(d)** E_y (imaginary)

(e) E_z (real) **(f)** E_z (imaginary)

Figure 5.7: Electric field variation along θ for $k_0 r = 24$, $\phi = \pi/4$, $k_0 a = 12$, $k_0 R_\infty = 24$ for the scattering from a conducting ellipsoid.

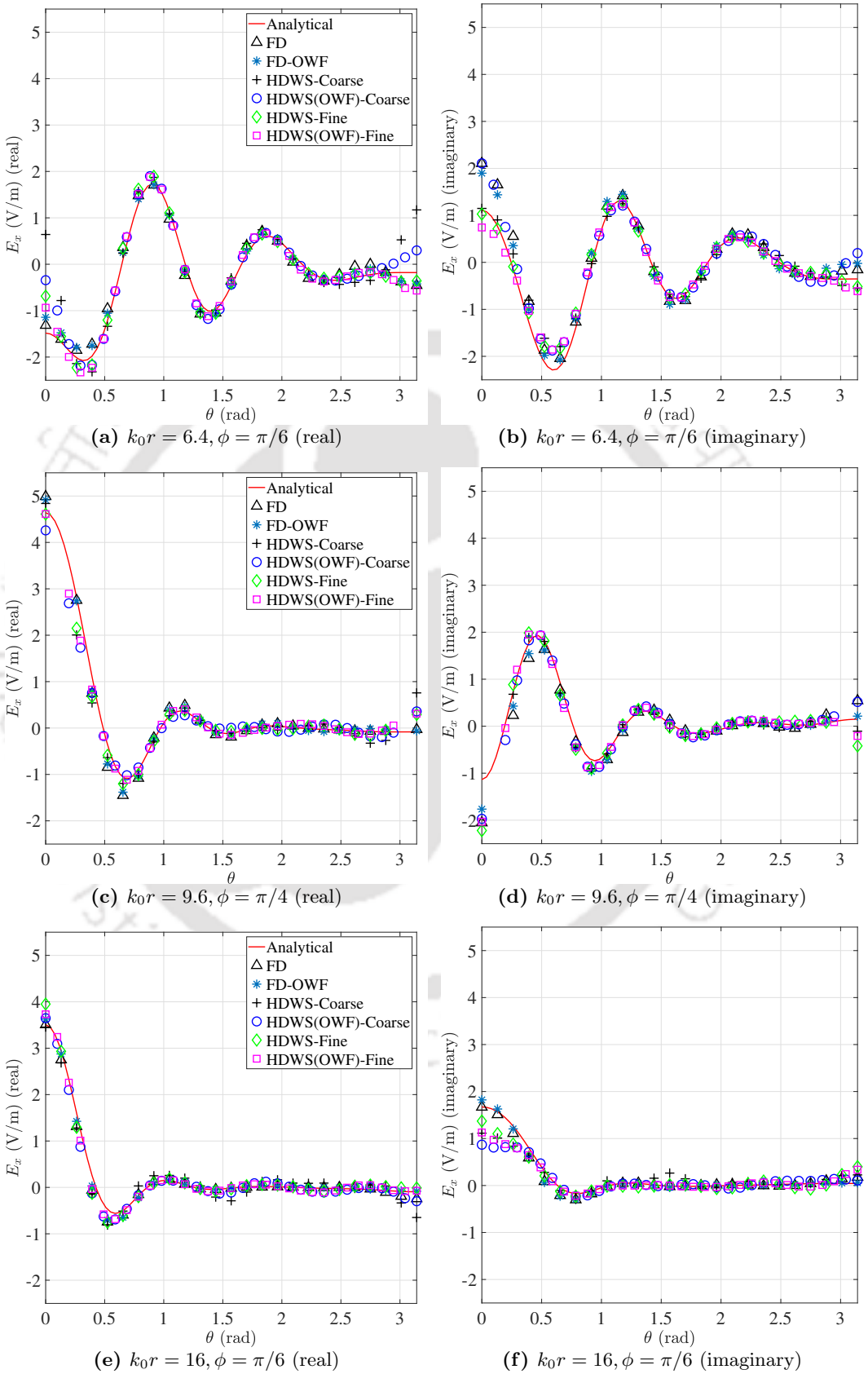
5.2.3 Scattering from a dielectric sphere

An incident wave of the form $\boldsymbol{E}_{\text{inc}} = E_0 e^{-ikz}\hat{\boldsymbol{e}}_x$, impinges on a dielectric sphere. The relative permeability and relative permittivity of the dielectric sphere and the surrounding medium are denoted by (μ_1, ϵ_1) and (μ_2, ϵ_2), respectively. The analytical solution has been presented by Stratton [107]. In our numerical analysis, we assume the dielectric sphere to be in vacuum i.e., $\mu_2 = \epsilon_2 = 1$ and it has material properties $\mu_1 = 1$ and $\epsilon_1 = 1.5$. We choose $ka = 8$, $kR_\infty = 24$, and $E_0 = 1$. For the proposed implementation of the half domain with symmetric boundary condition, we have used two different levels of meshing to discretize the domain. For coarse mesh 750 B27 and W18 elements are used to model the hemisphere, whereas the thin cylindrical patch is modelled with 250 B27 elements. In case of finer mesh, hemisphere is discretized with 1050 B27 and W18 elements and 350 B27 elements discretize the attached patch. In Table 5.3, information regarding the mesh details along with the resultant equations can be found.

Table 5.3: Mesh and equation details for scattering by dielectric sphere.

	Half domain		Full domain [22]
	Coarse	Fine	
Mesh details (No. of elements)	750(hemispherical domain) + 250(cylindrical patch)	1050(hemispherical domain) + 350(cylindrical patch)	1500
No. of equations	28496	38427	44692

Fig. 5.8 shows the results for the scattered electric field E_x inside the dielectric sphere at $k_0 r = 6.4$, at a point just outside the dielectric sphere ($k_0 r = 9.6$), and at a point in the far-field ($k_0 r = 16$) respectively. We have compared the results of both full domain (FD) and half domain with symmetry (HDWS) with analytical benchmark results as given by equation 34 in [22]. In Fig. 5.8, results from both conventional and OWF formulations have been presented. It is evident from these plots that the simulation results of both coarse and fine mesh are converging with analytical results. We have observed that with fine mesh using the conventional formulation, HDWS results exhibit a 4 % error, using 14 % less equations than corresponding full domain. Whereas, for the same mesh, using OWF formulation, 2.17 % error is achieved. However, with the chosen coarse mesh (36 % less equations than full domain), electric fields are predicted with 8.9 % error and 4.8 % error for conventional and OWF formulations, respectively.



(a) $k_0r = 6.4, \phi = \pi/6$ (real) **(b)** $k_0r = 6.4, \phi = \pi/6$ (imaginary)

(c) $k_0r = 9.6, \phi = \pi/4$ (real) **(d)** $k_0r = 9.6, \phi = \pi/4$ (imaginary)

(e) $k_0r = 16, \phi = \pi/6$ (real) **(f)** $k_0r = 16, \phi = \pi/6$ (imaginary)

Figure 5.8: For two different meshes, variation of electric field E_x along θ at different locations for the scattering from dielectric sphere problem.

5.3 Numerical Examples with Electromagnetic Transient Analysis

In this section we apply the novel implementation of the symmetric boundary condition for various electromagnetic transient problems from [24] to analyze the efficacy of the proposed method in time domain. These problems includes radiation inside a cube with conducting boundaries, scattering from a conducting and dielectric sphere. Again, we choose B27 and W18 elements to discretize the computational domains and we take $c = 3 \times 10^8/\sqrt{\epsilon_r \mu_r}$ m/s.

5.3.1 Electromagnetic radiation inside a cube with conducting walls

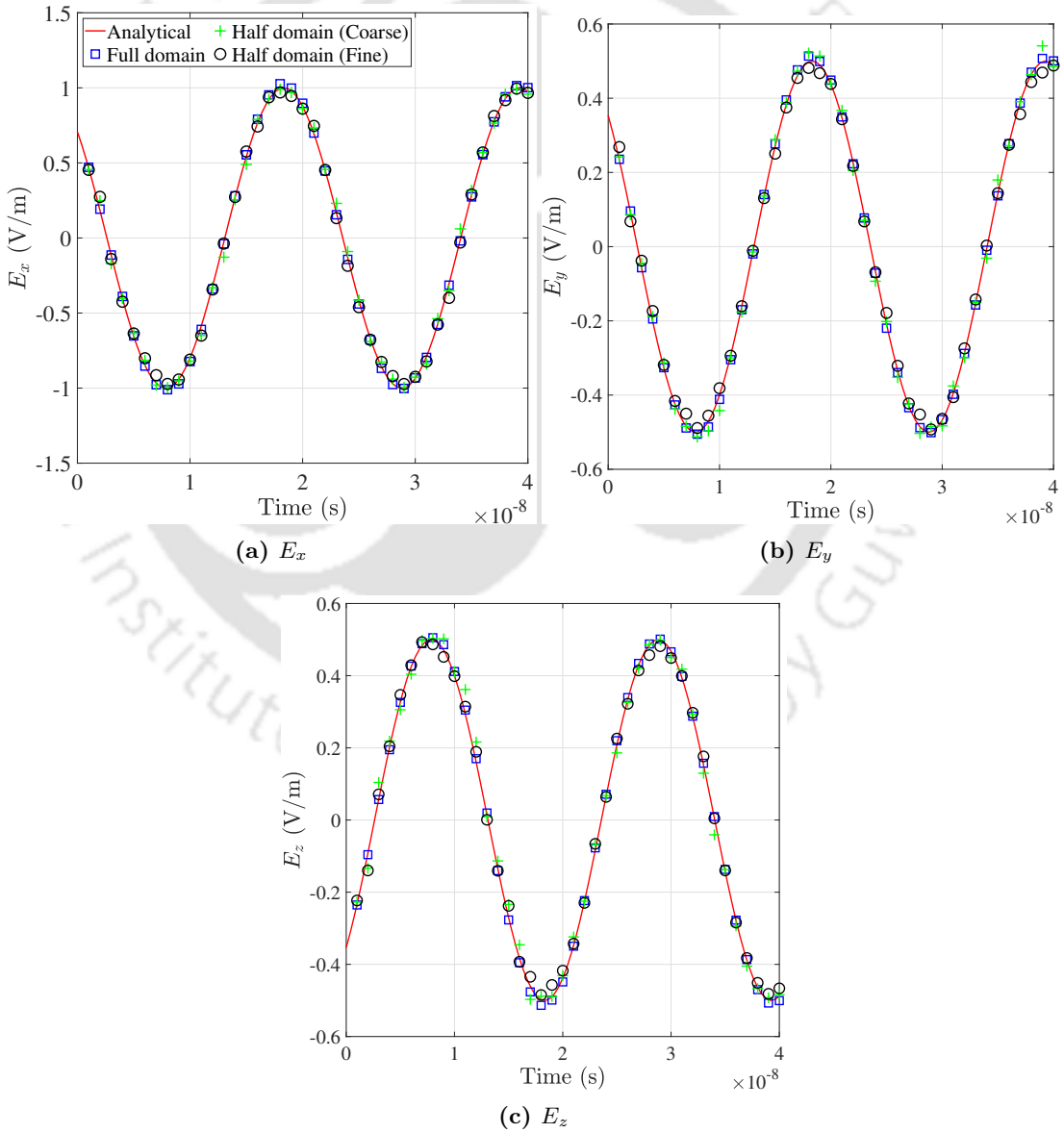
A cube of dimension $\pi \times \pi \times \pi$ with conducting walls is excited with $\boldsymbol{j}$ and with initial $\boldsymbol{A}$ values given in [24] as

$$
\begin{aligned}
E_x &= 2\cos(x)\sin(y)\sin(z)\left[\cos(\omega t) - \sin(\omega t)\right], \\
E_y &= \sin(x)\cos(y)\sin(z)\left[\sin(\omega t) - \cos(\omega t)\right], \\
E_z &= \sin(x)\sin(y)\cos(z)\left[\sin(\omega t) - \cos(\omega t)\right], \\
H_x &= 0, \\
H_y &= \frac{-3}{\mu\omega}\cos(x)\sin(y)\cos(z)\left[\cos(\omega t) + \sin(\omega t)\right], \\
H_z &= \frac{3}{\mu\omega}\cos(x)\cos(y)\sin(z)\left[\cos(\omega t) + \sin(\omega t)\right], \\
A_x &= \frac{-2}{\omega}\cos(x)\sin(y)\sin(z)\left[\sin(\omega t) + \cos(\omega t)\right], \\
A_y &= \frac{1}{\omega}\sin(x)\cos(y)\sin(z)\left[\cos(\omega t) + \sin(\omega t)\right], \\
A_z &= \frac{1}{\omega}\sin(x)\sin(y)\cos(z)\left[\cos(\omega t) + \sin(\omega t)\right], \\
\psi &= 0, \\
j_x &= \left(\frac{2\epsilon\mu\omega^2 - 6}{\mu\omega}\right)\cos(x)\sin(y)\sin(z)\left[\cos(\omega t) + \sin(\omega t)\right], \\
j_y &= \left(\frac{3 - \epsilon\mu\omega^2}{\mu\omega}\right)\sin(x)\cos(y)\sin(z)\left[\cos(\omega t) + \sin(\omega t)\right], \\
j_z &= \left(\frac{3 - \epsilon\mu\omega^2}{\mu\omega}\right)\sin(x)\sin(y)\cos(z)\left[\cos(\omega t) + \sin(\omega t)\right].
\end{aligned}
\tag{5.19}
$$

We have taken $\omega = 3 \times 10^8$m/s, and simulated upto 4×10^{-8}sec with a time step of $t_\Delta = 1 \times 10^{-9}$sec.

Table 5.4: Mesh and equation details for electromagnetic radiation inside a cube with conducting walls.

	Half domain		Full domain [24]
	Coarse mesh	Fine mesh	
Mesh size $(x \times y \times z)$	$2 \times 4 \times 2$ (cuboid) + $(1 \times 4 \times 2) + (2 \times 4 \times 1)$ + $(1 \times 4 \times 1)$ (patches)	$3 \times 3 \times 3$ (cuboid) + $(1 \times 3 \times 3) + (3 \times 3 \times 1)$ + $(1 \times 3 \times 1)$ (patches)	$8 \times 8 \times 8$
No. of elements	36	48	64
No. of equations	890	1208	1666



(a) E_x **(b)** E_y

(c) E_z

Figure 5.9: Variation of the electric fields at $(x, y, z) = (1.1780, 0.3926, 0.7853)$ with respect to time for the cube with conducting walls.

To implement the symmetric boundary conditions, we have taken one quarter of the original cube with $\frac{\pi}{2} \times \pi \times \frac{\pi}{2}$. On the symmetry faces i.e., on xy and yz-planes, we have applied $\boldsymbol{E} \cdot \boldsymbol{n} = \boldsymbol{0}$ as explained in section 5.1.5, and on the remaining faces of the cuboid, conducting boundary conditions $\boldsymbol{E} \times \boldsymbol{n} = \boldsymbol{0}$ are applied. For transient analysis, two distinct meshes are used to mesh the quarter domain and the thin patches with B27 elements, whereas full domain is meshed with 64 B27 elements. For the quarter domain, the coarse mesh comprises of 36 elements, and the fine mesh employs 48 elements. Table 5.4 summarizes the mesh details for both the domains. In Fig. 5.9, a comparison has been carried out for the electric field components at ($x = 1.1780$, $y = 0.3926$, $z = 0.7853$) among obtained results in the proposed formulation for both meshes of the quarter domain, the corresponding results from the full domain, and the analytical solution [24]. Again, it is evident, by adopting the symmetric boundary conditions, 1208 equations (for finer mesh) are sufficient for the perfect match with analytical benchmark as compared to 1666 unknowns for the full domain. The coarse mesh solution (with only 890 equations) is deviating by little amount from the benchmark solution.

5.3.2 Electromagnetic scattering from a conducting sphere

Table 5.5: Mesh and equation details for scattering from a conducting sphere.

	Half domain with thin patch		Full domain [24]
	Coarse mesh	Fine mesh	
Mesh size $(r \times \theta \times \phi)+$ $(r \times \theta \times t)$	$7 \times 8 \times 5 +$ $7 \times 16 \times 1$	$8 \times 8 \times 5 +$ $8 \times 16 \times 1$	$8 \times 8 \times 10$
Number of elements	392	448	640
Number of equations	11169	11411	14946

To know the time domain performance of the proposed method on curved objects, we consider the present example where we simulate the transient scattering electric fields from a conducting sphere. An incident wave as stated in [24] given as

$$\boldsymbol{E}_{\text{inc}} = 2\{t - t_0 - \hat{\boldsymbol{k}} \cdot (\boldsymbol{r} - \boldsymbol{r}_0)/c\} \exp\left[-\frac{\{t - t_0 - \hat{\boldsymbol{k}} \cdot (\boldsymbol{r} - \boldsymbol{r}_0)/c\}^2}{\tau^2}\right] \hat{\boldsymbol{E}}, \tag{5.20}$$

impinges on a conducting sphere. Here, t_0 is the temporal reference point, $\boldsymbol{r}_0$ is the spatial reference point, and τ is the pulse shape. Following pulse parameters such as $\hat{\boldsymbol{k}} = \hat{\boldsymbol{z}}$, $\hat{\boldsymbol{E}} = \hat{\boldsymbol{x}}$, $t_0 = 25.99$ ns, $\boldsymbol{r}_0 = -1.2\hat{\boldsymbol{z}}$ m and $\tau = 5.25$ ns are used for simulation. Computational full domain which is truncated at a radius $R_\infty = 3.8$ m, is meshed with 640 B27 and W18 elements which results in 14,946 equations in [24]. In the current symmetric boundary condition implementation, the half domain with cylindrical patch

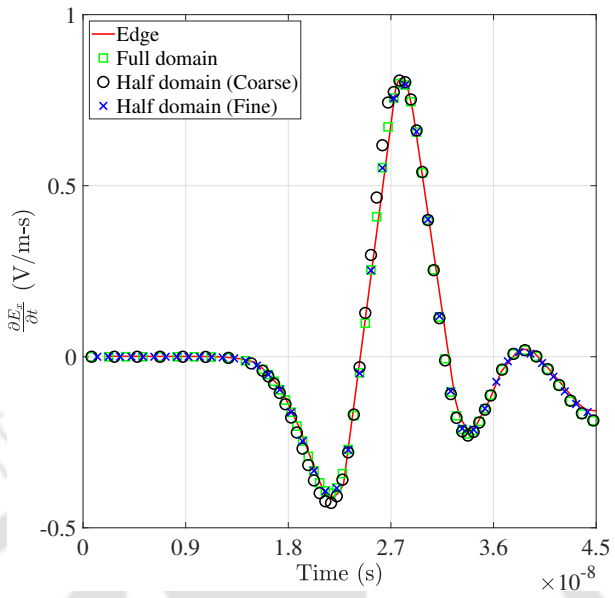


Figure 5.10: Time variation of the $\frac{\partial E_x}{\partial t}$ at $(x, y, z) = (0.33, -1.03, -0.45)$ for scattering from a conducting sphere.

is meshed with 392 B27 and W18 elements resulting in 11,169 equations for the coarse mesh, and 448 elements (11,411 unknowns) for the fine mesh, as specified in Table 5.5. Boundary conditions such as $\boldsymbol{E}\cdot\boldsymbol{n} = \boldsymbol{0}$ is implemented on the surface of the thin patch as described in section 5.1.5, and $\boldsymbol{E}\times\boldsymbol{n} = \boldsymbol{0}$ is applied on the conducting surface. We have taken a time step of 5×10^{-10}sec in our simulation. From Fig. 5.10, it is evident that $\frac{\partial E_x}{\partial t}$ values at $(0.33, -1.03, -0.45)$ in both full domain and half domain with symmetry, are perfectly matching with the standard benchmark results in [94]. While solving for electric fields, with proposed symmetric boundary condition around 25% (for coarse mesh) and approximately 23% (for finer mesh) less equations are required to solve as compared to the full domain. This clearly illustrates the computational efficacy of the proposed method.

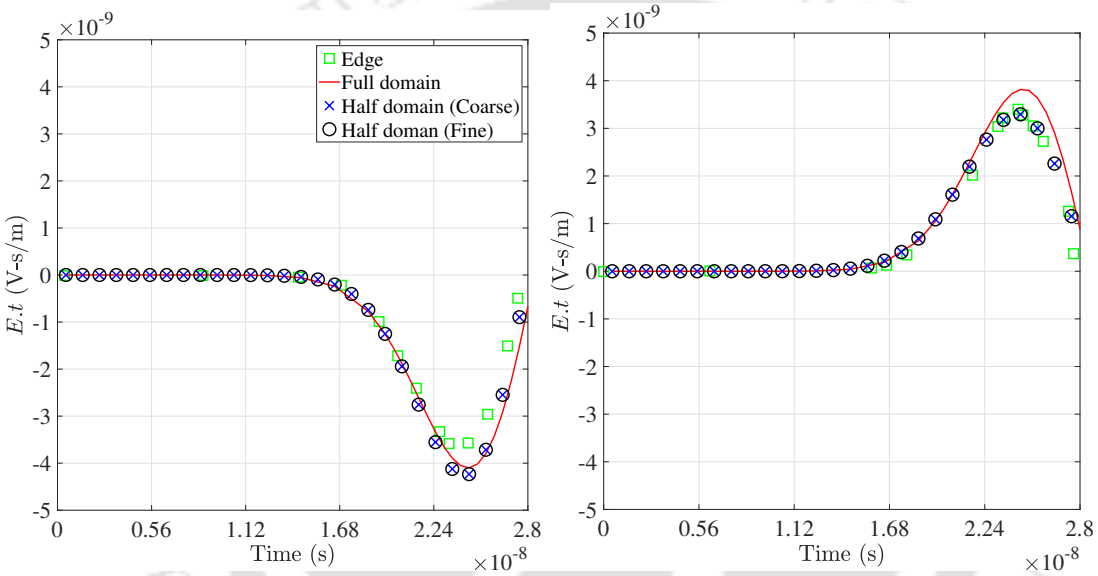
5.3.3 Electromagnetic scattering from a dielectric sphere

Table 5.6: Mesh and equation details for scattering from a dielectric sphere.

	Half domain		Full domain [24]
	Coarse	Fine	
Mesh size	1536 (hemispherical + domain) + 512 (patch)	2196 (hemispherical + domain) + 732 (patch)	
No. of elements	2048	3702	4239
No. of equations	49076	73755	106139

This example is chosen to test the effectiveness of the proposed method in predicting scattering electric fields from a dielectric body in time domain. On a dielectric sphere of radius 0.5 m, having $\mu_r = 1$ and $\epsilon_r = 6.0$, Neumann pulse wave as given by

$$\boldsymbol{E}_{\text{inc}} = 2\{t - t_0 - \hat{\boldsymbol{k}} \cdot (\boldsymbol{r} - \boldsymbol{r}_0)/c\} \exp\left[-\frac{\{t - t_0 - \hat{\boldsymbol{k}} \cdot (\boldsymbol{r} - \boldsymbol{r}_0)/c\}^2}{\tau^2}\right] \hat{\boldsymbol{E}}, \tag{5.21}$$



(a) $\boldsymbol{E}.\boldsymbol{t}$ at $(x, y, z) = (-0.04, -0.07, -0.72)$ with $\boldsymbol{t} = 0.96\hat{\boldsymbol{x}} + 0.26\hat{\boldsymbol{y}} + 0.13\hat{\boldsymbol{z}}$

(b) $\boldsymbol{E}.\boldsymbol{t}$ at $(x, y, z) = (0.05, 0.05, -0.96)$ with $\boldsymbol{t} = -0.89\hat{\boldsymbol{x}} - 0.08\hat{\boldsymbol{y}} + 0.44\hat{\boldsymbol{z}}$

Figure 5.11: Transient variation of the $\boldsymbol{E}.\boldsymbol{t}$ for scattering from a dielectric sphere.

in [24] is impinged. The computational domain which is truncated at radius $R_\infty = 2$ m, is modeled with B27 and W18 elements. Table 5.6 gives the numerical analysis details for both full and half domains. Proper symmetric boundary condition is applied with thin cylindrical patch as described in section 5.1.5. With a time step of 0.5 ns, simulations are performed. We have plotted $\boldsymbol{E} \cdot \boldsymbol{t}$ along time in Fig. 5.11 for full domain, half domain with symmetry and compared it with the edge element results in [147] at two locations: (a) at $(-0.04, -0.07, -0.72)$ with $\boldsymbol{t} = 0.96\hat{\boldsymbol{x}} + 0.26\hat{\boldsymbol{y}} + 0.13\hat{\boldsymbol{z}}$ and (b) at $(0.05, 0.05, -0.96)$ with $\boldsymbol{t} = -0.89\hat{\boldsymbol{x}}i - 0.08\hat{\boldsymbol{y}} + 0.44\hat{\boldsymbol{z}}$. Fig. 5.11 shows the efficacy of our proposed method for transient scattering problem with dielectric body. Our proposed method predicts the benchmark electric fields with 54% (for coarse mesh) and 30% (for fine mesh) less equations as compared to the full domain analysis.

5.4 Study on the effect of variation of thickness of a patch

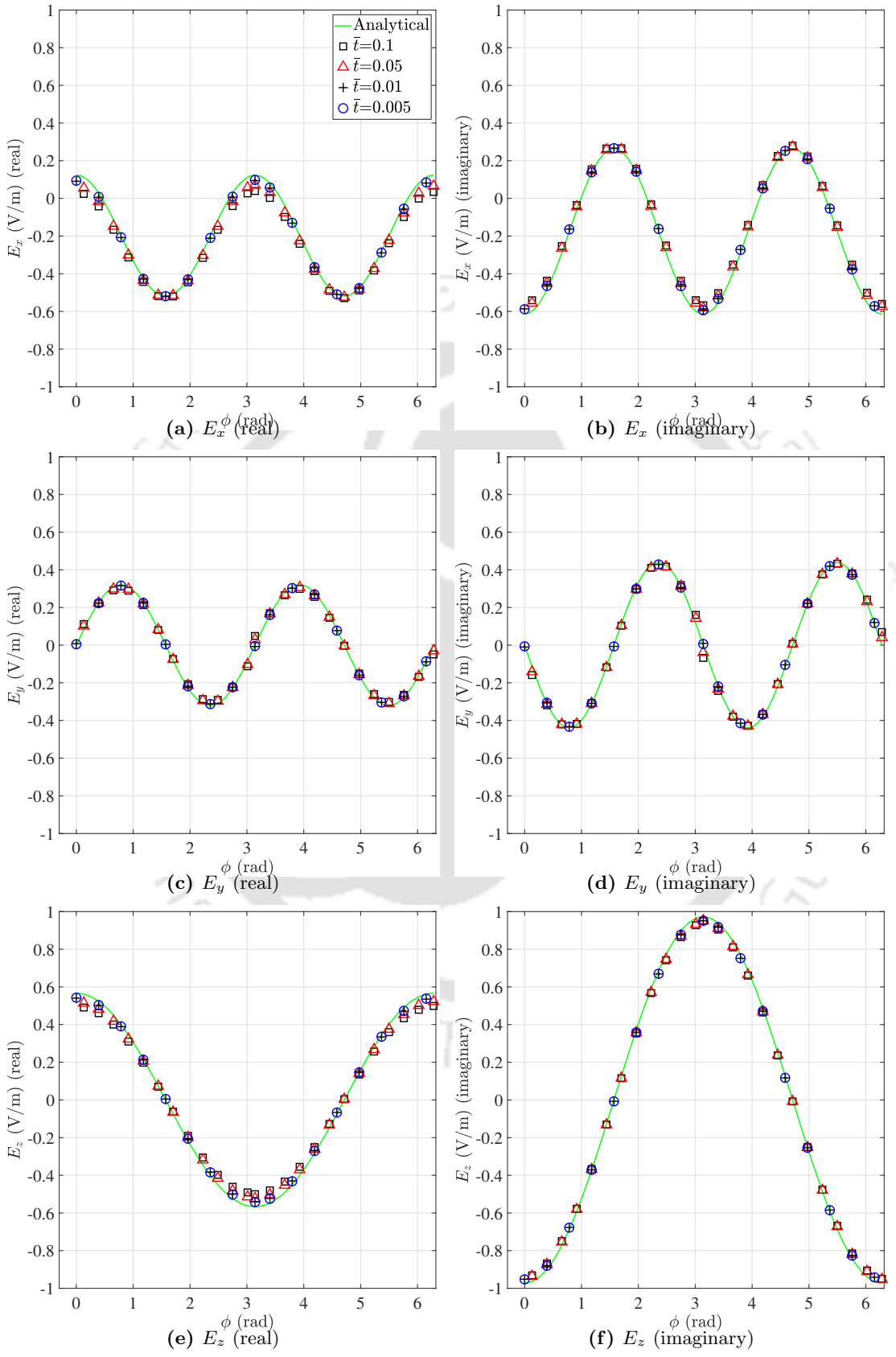
It is important to understand the performance of the proposed method with respect to the variation of thickness of the cylindrical patch attached to the symmetry face. For this, one non-dimensional parameter $\bar{t}$, the ratio of thickness of the patch to the characteristic length of the full domain, is defined. We perform numerical simulation for four different values of $\bar{t}$ with $t = 0.1$, 0.05, 0.01, and 0.005 for three representative electromagnetic harmonic and transient problems.

5.4.1 Scattering from a conducting sphere (Harmonic analysis)

Table 5.7: Patch thickness variations for scattering from a conducting sphere.

Mesh size $(r \times \theta \times \phi)+$ $(r \times \theta \times t)$	Number of Equations	Thickness of patch (t)	$R = 1$ Parameter, $\bar{t} = \frac{t}{R}$	L_2-norm values
$16 \times 12 \times 12$ + $16 \times 24 \times 12$	80125	0.1	0.1	4.3290×10^{-2}
		0.05	0.05	3.7952×10^{-2}
		0.01	0.01	3.6947×10^{-2}
		0.005	0.005	3.6671×10^{-2}

To demonstrate the thickness variation analysis, let us consider the numerical example presented in section 5.2.1. For four different cases, cylindrical patches are generated with four different thickness (t) values. Here, we have chosen a mesh size of $16 \times 12 \times 12$ to discretize the hemisphere using B27 and W18 elements. On the other hand, the thin patch is meshed with a mesh size of $16 \times 24 \times 12$ B27 elements. With a total of 80125 equations, simulations are executed to generate the electric fields. Considering the radius of sphere $R = 1$, $\bar{t} = \frac{t}{R}$, is calculated for all the t values as displayed in Table 5.7. The resulting L_2-norm values, obtained from Eq. (5.18) are also presented in Table 5.7. At $\phi = \pi/4$ and at two different locations, namely the near field ($k_0 r = 1.25$) and far field ($k_0 R_\infty = 4.75$), variation of electric fields along the azimuthal angle (ϕ) are plotted for different $\bar{t}$ values along with the analytical values. Fig. 5.12 and Fig. 5.13 illustrate these plots for near field and far field, respectively. The observation from these plots indicates a divergence between the simulation results and the corresponding analytical values as the thickness (t) increases. Our analysis suggests that, for accurate solutions, the patch should be thinner ($\bar{t} < 0.01$) in comparison to the other dimensions of the computational domain.

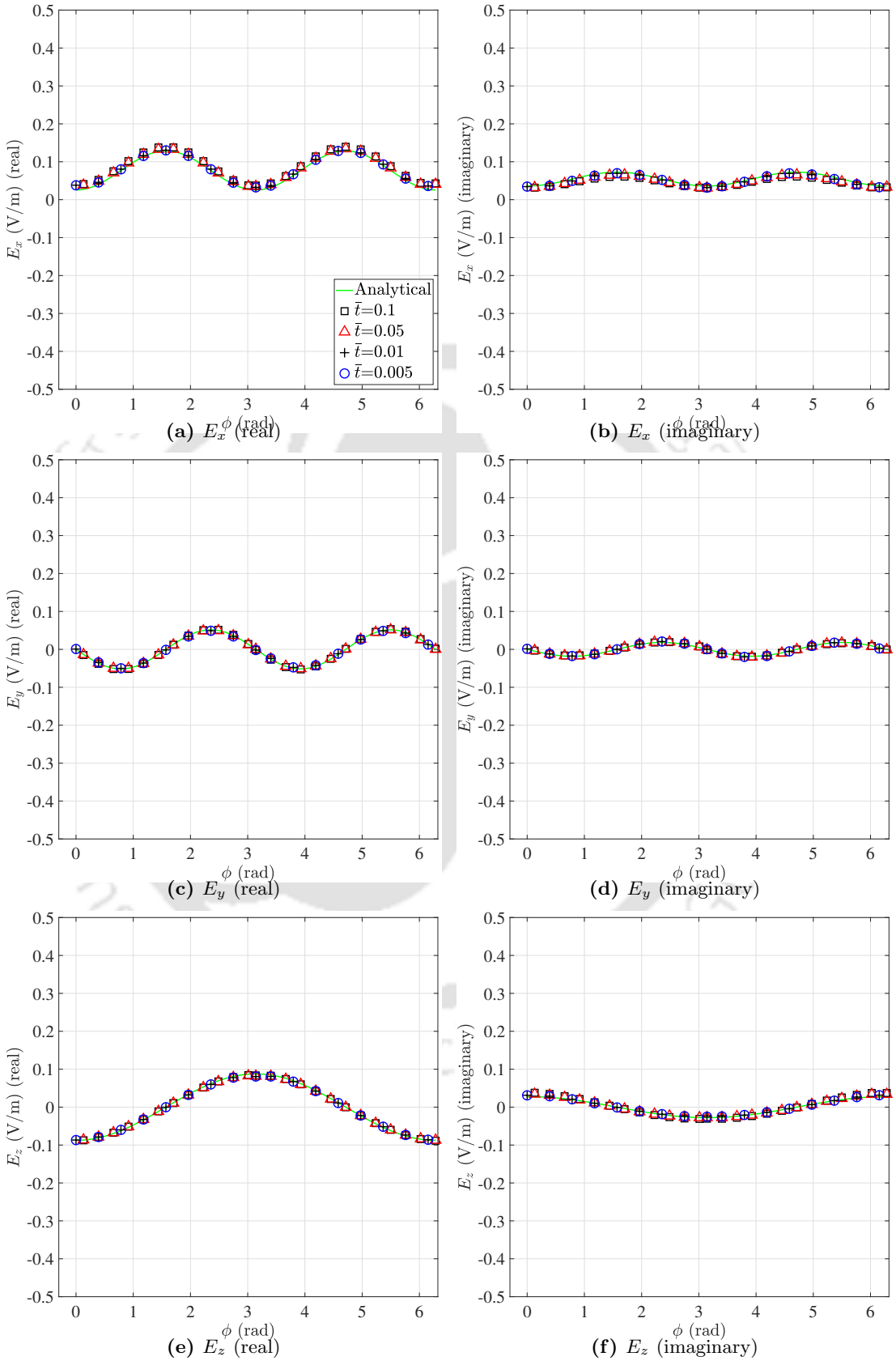


(a) E_x (real) **(b)** E_x (imaginary)

(c) E_y (real) **(d)** E_y (imaginary)

(e) E_z (real) **(f)** E_z (imaginary)

Figure 5.12: For different values of $\bar{t}$, near field variation along ϕ in the electric field for parameters $k_0 r = 1.25$, $\theta = \pi/4$, $k_0 a = 1$, $k_0 R_\infty = 5$ for the scattering from a conducting sphere.



(a) E_x (real) **(b)** E_x (imaginary)

(c) E_y (real) **(d)** E_y (imaginary)

(e) E_z (real) **(f)** E_z (imaginary)

Figure 5.13: For different values of $\bar{t}$, far field variation along ϕ in the electric field for parameters $k_0 r = 4.75$, $\theta = \pi/4$, $k_0 a = 1$, $k_0 R_\infty = 5$ for the scattering from a conducting sphere.

5.4.2 Scattering from a dielectric sphere (Harmonic analysis)

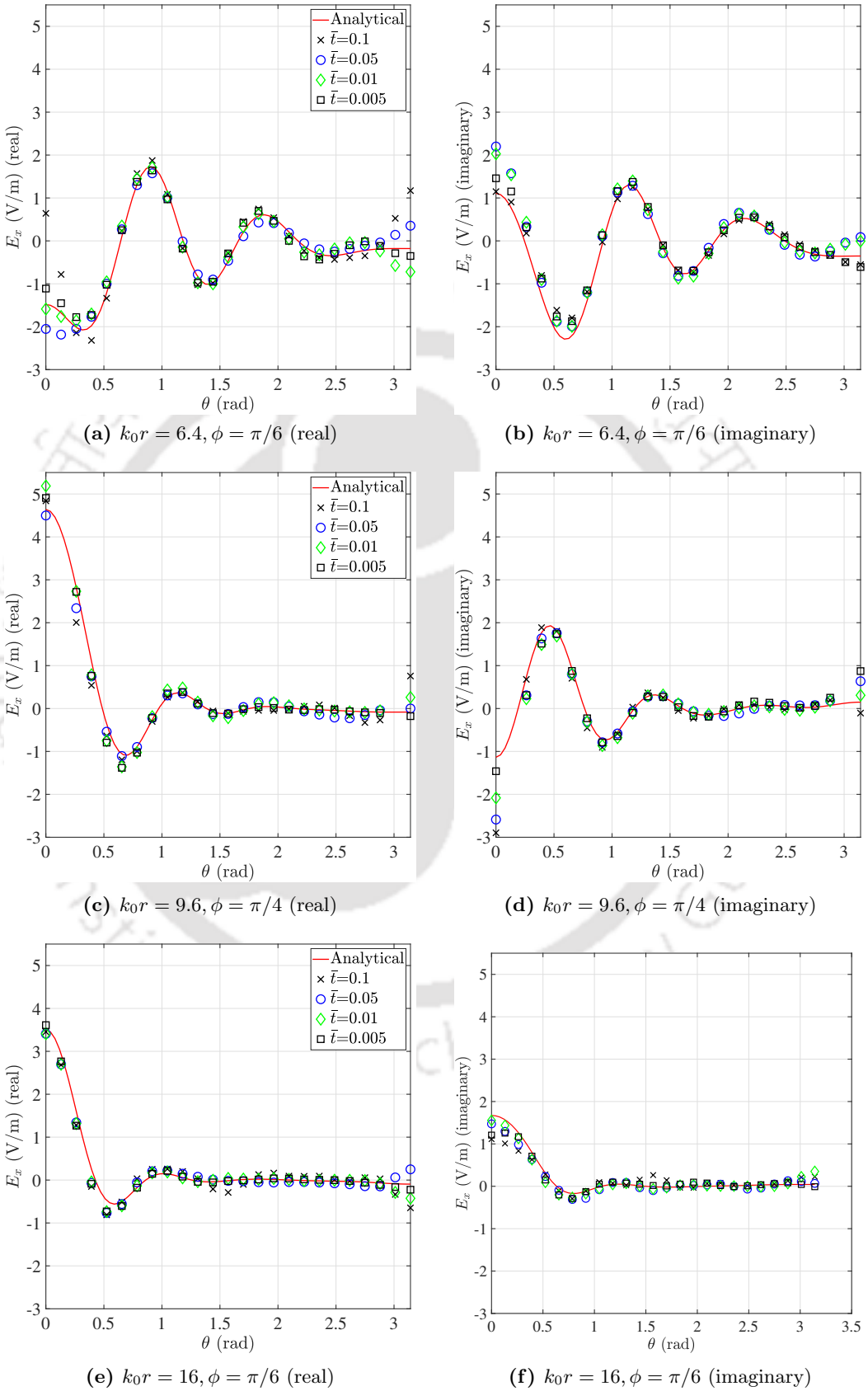
In this section, effect of thickness variation is analysed for harmonic scattering problem from a dielectric sphere, presented in section 5.2.3. Here, the hemisphere and thin patch are modeled with 750 B27 and W18 elements, and 250 B27 elements respectively. By solving a set of 28496 equations, electric fields are obtained for four different thickness (t) cases using both conventional and OWF formulations. In Table 5.8, for four $\bar{t}$ values, computed L_2-norm error values are presented and compared for both the formulations. At three different locations, namely at $k_0r = 6.4$ with $\phi = \pi/6$, $k_0r = 9.6$ with $\phi = \pi/4$, and at $k_0r = 16$ with $\phi = \pi/6$, E_x values are plotted along θ for four thickness parameters in Fig. 5.14 and Fig. 5.15 for conventional and OWF formulations, respectively. Numerical results obtained with $t = 0.1$ exhibit a more pronounced deviation from the actual values. This finding again emphasizes the need to conduct the analysis with an extremely thin patch to achieve accurate solutions.

Table 5.8: Patch thickness variations for scattering from a dielectric sphere.

Number of elements	Number of Equations	Thickness of patch (t)	$R = 1$ Parameter, $\bar{t} = \frac{t}{R}$	L_2-norm values	
				Conventional	OWF
750(hemispherical domain)+ 250(cylindrical patch)	28496	0.1	0.1	11.34×10^{-2}	8.51×10^{-2}
		0.05	0.05	10.93×10^{-2}	7.86×10^{-2}
		0.01	0.01	9.54×10^{-2}	6.61×10^{-2}
		0.005	0.005	8.91×10^{-2}	4.85×10^{-2}

5.4.3 Radiation inside a cube with conducting walls (Transient analysis)

Now, to analyse the effect of thickness variation in transient electromagnetic problems, we consider the problem of cube with conducting walls having a side length $a = \pi$, as outlined in section 5.9. In the same manner, one quarter of the domain is considered to apply the proposed method. Various thickness values are considered for the mesh with 36 B27 elements and 890 equations (see table 5.4). For different t values, thickness parameters ($\bar{t}$) and are listed in Table 5.9 along with the mesh details. In Fig. 5.16, we can observe the electric field time response for various components for three different t values at a specific point $(1.1780, 0.3926, 0.7853)$ along with the analytical results. It is worth noting that the field components match more effectively with a decrease in $\bar{t}$. Hence, it again becomes evident that, for precise finite element solutions using the proposed method, the patch thickness should be very small in comparison to other dimensions of the domain.



(a) $k_0 r = 6.4, \phi = \pi/6$ (real)

(b) $k_0 r = 6.4, \phi = \pi/6$ (imaginary)

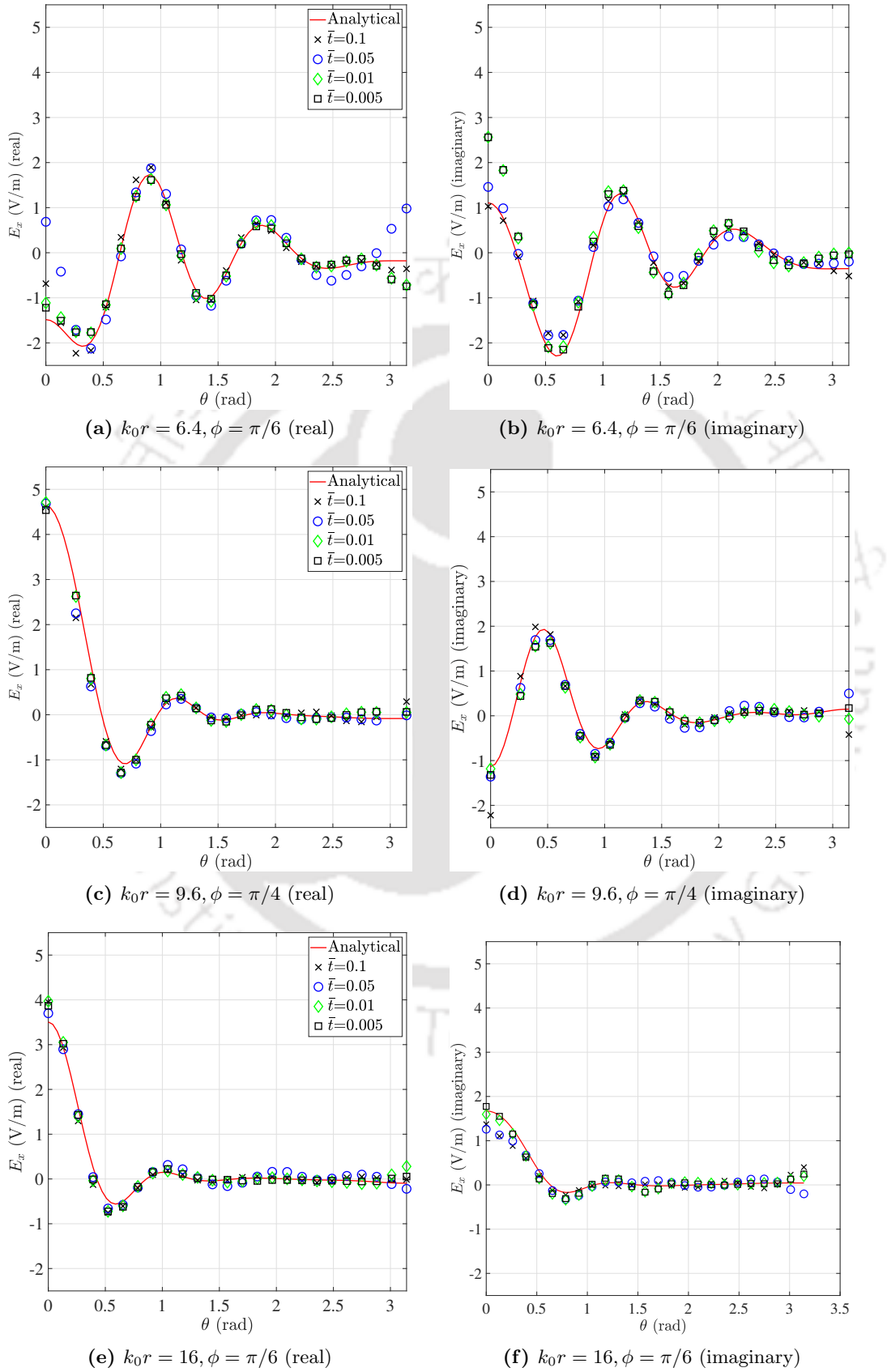
(c) $k_0 r = 9.6, \phi = \pi/4$ (real)

(d) $k_0 r = 9.6, \phi = \pi/4$ (imaginary)

(e) $k_0 r = 16, \phi = \pi/6$ (real)

(f) $k_0 r = 16, \phi = \pi/6$ (imaginary)

Figure 5.14: For distinct values of $\bar{t}$ i.e., $\bar{t} = \frac{t}{R}$, E_x variation (conventional formulation) along θ at different locations for the scattering from dielectric sphere problem.



(a) $k_0 r = 6.4, \phi = \pi/6$ (real)

(b) $k_0 r = 6.4, \phi = \pi/6$ (imaginary)

(c) $k_0 r = 9.6, \phi = \pi/4$ (real)

(d) $k_0 r = 9.6, \phi = \pi/4$ (imaginary)

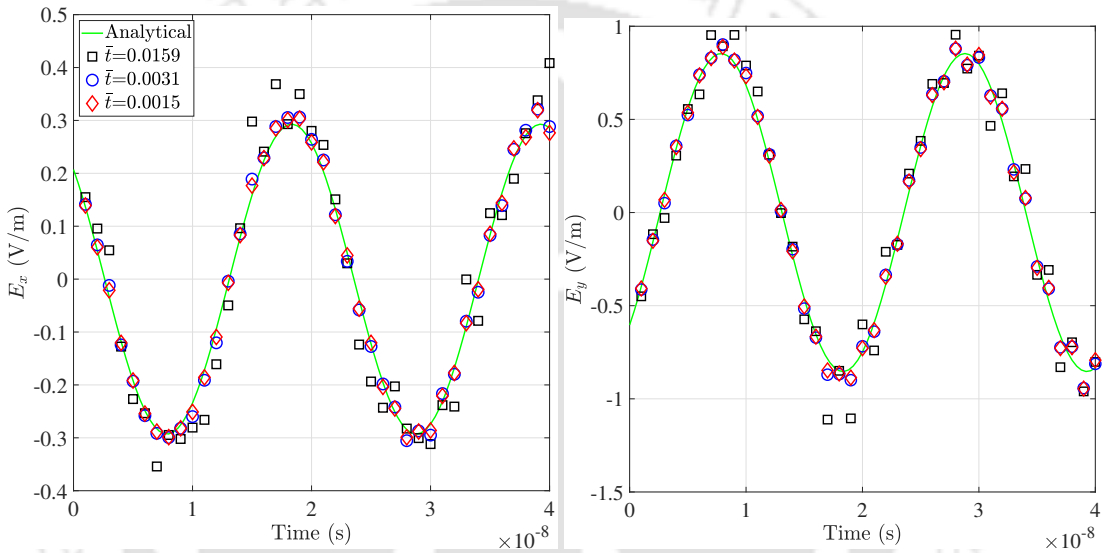
(e) $k_0 r = 16, \phi = \pi/6$ (real)

(f) $k_0 r = 16, \phi = \pi/6$ (imaginary)

Figure 5.15: For various values of $\bar{t}$ i.e., $\bar{t} = \frac{t}{R}$, variation of E_x (OWF formulation) along θ at different locations for the scattering from dielectric sphere problem.

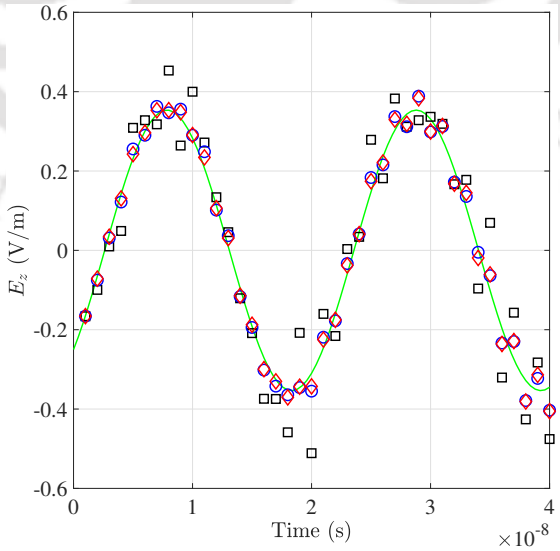
Table 5.9: Variation of patch thickness for electromagnetic radiation inside a cube with conducting walls.

Mesh size $(x \times y \times z)$	Number of elements	Number of equations	Thickness of patch (t)	side, $a = \pi$ Parameter $(\bar{t})$ $\bar{t} = \frac{t}{a}$
$2 \times 4 \times 2$ (cuboid)+ $(1 \times 4 \times 2)$+$(2 \times 4 \times 1)$ +$(1 \times 4 \times 1)$(patches)	36	890	0.05 0.01 0.005	0.0159 0.0031 0.0015



(a) E_x

(b) E_y



(c) E_z

Figure 5.16: For different values of $\bar{t}$ $(\bar{t} = t/a)$, variation of the electric fields at $(x, y, z) = (1.1780, 0.3926, 0.7853)$ with respect to time for the cube with conducting walls.

5.5 Summary

One novel method is proposed to implement symmetric boundary conditions to electromagnetic problems for both harmonic and transient analysis. In nodal framework, potential formulations are followed and it is not very straight forward to implement the symmetric boundary condition in that formulation. In the proposed method, a very thin patch is attached to the symmetry faces of the domain. In that thin patch, two different conditions are required for scalar and vector potentials for proper implementation of the symmetric boundary condition. We have validated the proposed method solving a wide range of problems from radiation to scattering, from spherical to non-spherical domains, from harmonic to transient analysis. For all the problems, we have validated our results against either analytical benchmark results or other benchmark from existing literatures. For harmonic analysis, we have implemented the proposed novel method in amplitude formulation [22] also. Furthermore, we have presented the computational efficacy of our proposed method as compared to the existing full domain analysis for all the examples. Lastly, we have studied the effect of thickness variation of the thin patch on the proposed method in both harmonic and transient problems. In order to obtain more accurate results with the proposed method, it is recommended that the patch should be very thin as compared to the other dimension of the domain.

Chapter 6

Conclusions and future scope of the research work

6.1 Conclusions

In this chapter, we want to conclude the work done in our research work briefly, along with the future scope of the work.

In the present work, different finite element strategies for solving electromagnetic problems have been enriched with some additional numerical strategies.

In Chapter 1, a comprehensive literature review is provided on various numerical methods in computational electromagnetics. The methods discussed include Finite Element Method (FEM), Finite Difference Time Domain (FDTD), Time Domain Finite Element Method (TDFEM), Perfectly Matched Layer (PML), Boundary Integral Element Method (BIEM), Finite Element Boundary Integral Method(FEBIM), etc. Additionally, a brief overview of the development and application of edge elements in FEM is presented. The chapter also covers different strategies and algorithms available for implementing edge elements in computational electromagnetics. Furthermore, numerical methods in the time domain, such as FEM, FDTD, TDFEM, etc., are discussed. Finally, the chapter concludes with a brief discussion of the different numerical methods available for applying symmetric boundary condition in electromagnetics.

In Chapter 2, a conversion algorithm which generates edge finite element data from nodal finite element data, is proposed. This algorithm transforms 2D edge elements, including lower-order elements such as 3-edge triangle, 4-edge quadrilateral, as well as higher-order elements such as 8-edge triangle, and 12-edge quadrilateral. The algorithm is then extended to generate the 3D edge elements, including 12-edge hexahedral and 6-edge tetrahedra. The basic intrinsic details of the algorithm are provided with suitable examples to enable anyone interested in developing their own code for simulating electromagnetic problems. Then, the transformed edge elements are validated through eigenvalue analysis.

A detailed performance comparison between conventional nodal and edge elements has been presented in Chapter 3. Various solved examples include all possible complexities like curved boundaries, non-convex domains, sharp corners domains. 2D domains include square, homogeneous and inhomogeneous L shape, curved L shape, circle with and without crack, whereas 3D domains include cube, Fichera, and 3D L shape. In every case, edge elements have shown better coarse mesh accuracy than nodal elements. We have observed in many cases that, in order to achieve the same level of accuracy, the required no. of equations with edge element is less than half of that with nodal elements. For the non-convex domains with sharp corners, nodal elements cannot predict the singular eigen value which is well predicted by all the edge elements. In addition, for such domains, nodal elements predict one additional spurious eigen value which is not present with edge elements. A computational cost analysis is also performed between standard nodal and transformed edge elements in 2D to evaluate the effectiveness of the edge elements.

Further, in Chapter 4, these edge elements are extended to perform electromagnetic transient analysis. One quantity ($\bar{E}$) is proposed in continuum framework which conserve under certain loading and boundary conditions. A time-stepping strategy is developed in edge-element framework which exactly mimic the conservation of the continuum quantity. Then, electromagnetic radiation and scattering analysis for conducting and dielectric domains have been performed with the developed strategy.

Finally, in Chapter 5, one novel method is proposed to implement symmetric boundary condition to electromagnetic problems for both harmonic and transient analysis. Then we have validated the proposed method solving a wide range of problems from radiation to scattering, from spherical to non-spherical domains, from harmonic to transient analysis. For all the problems, we have validated our results against either analytical benchmark results or other benchmark from existing literatures. For harmonic analysis, we have implemented the proposed method in amplitude formulation [22] also. An extensive study is also carried out for proper selection of the thickness in the thin patch method.

The following are the objectives which can be carried out as future scope of the present work:

In our present work, conversion algorithm is implemented in 2D lower and higher order node-to-edge elements, and in 3D lower order tetrahedral and hexahedral finite elements. As the future scope of work, conversion technique can be extended to higher order tetrahedral, hexahedral and wedge finite elements. To know the efficacy of these elements, comparative performance analysis can be performed with the corresponding nodal finite elements.

In the present study, electromagnetic radiation and scattering analysis in time domain problems involving both conducting and dielectric bodies are conducted using 3D transformed edge elements. As a future scope of work, these elements can be imple-

mented in electromagnetic harmonic analysis. Additionally, an extension of the work could involve implementing amplitude formulation within the context of an edge-based framework. Further, we can perform the comparative study of nodal and edge elements for time domain problems.

In this work, we have implemented the proposed time integration scheme using the Newmark$-\beta$ method, with the parameters β and γ as 0.25 and 0.5, respectively. To extend the research work, different time marching schemes can be developed by varying the values of β and γ. Then, performance comparison of these newly developed methods could then be conducted to evaluate their effectiveness.

List of publications

The following is the list of publications based on the work reported in this thesis:

Publications in Peer Reviewed Journals

1. D. Kamireddy and A. Nandy, "A novel conversion technique from nodal to edge finite element data structure for electromagnetic analysis", *Computer Assisted Methods in Engineering and Science* 28(4), 291−319 (2022). http://dx.doi.org/10.24423/cames.384
2. D. Kamireddy, S. Madhukar Chavan, and A. Nandy, "Comparative performance of novel nodal-to-edge finite elements over conventional nodal element for electromagnetic analysis", *Journal of Electromagnetic Waves and Applications* 37(1), 131−161 (2023). https://doi.org/10.1080/09205071.2022.2117102
3. D. Kamireddy and A. Nandy, "Exact conservation in transient electromagnetics using edge elements", *International Journal for Computational Methods in Engineering Science and Mechanics*,1−15 (2004). https://doi.org/10.1080/15502287.2024.2372813

Publications Submitted

1. D. Kamireddy, S. Karanam, and A. Nandy, "A novel implementation of symmetric boundary condition in harmonic and transient analysis of a electromagnetic wave propagation".(under review) [Available at: https://arxiv.org/abs/2212.04957]

Book chapter

1. D. Kamireddy, and A. Nandy, "Conversion algorithm from nodal to edge finite element data structure for higher order elements", *in Recent Advances in Mechanical Engineering, Lecture Notes in Mechanical Engineering*, pp. 612-629, Springer Nature, Singapore, 2022. https://doi.org/10.1007/978-981-19-2188-9_56
2. S. Karanam, D. Kamireddy, and A. Nandy, "Implementing symmetric boundary condition in electromagnetic harmonic analysis: two different ways", *in Advances*

in Signal Processing, Embedded Systems and IoT, Lecture Notes in Electrical Engineering, pp. 115125, Springer Nature, Singapore, 2023. https://doi.org/10.1007/978-981-19-8865-3_10

Conference

1. D. Kamireddy, and A. Nandy, "Creating edge element from four node quadrilateral element", *IOP Conference Series: Materials Science and Engineering*, 1080(1), pp. 12-15, 2021. https://dx.doi.org/10.1088/1757-899X/1080/1/012015
2. D. Kamireddy, S. Madhukar Chavan and A. Nandy, "Electromagnetic eigen analysis: performance comparison of four node and four edge quadrilateral elements with the effect of distortion", *Materials Today: Proceedings*, 66(4), pp. 1968-1972, 2022. https://doi.org/10.1016/j.matpr.2022.05.434

References

[1] J.-M. Jin, The Finite Element Method in Electromagnetics, 3rd Edition, John Wiley & Sons, Inc., New Jersey, 2014.

[2] S. Yang, Z. Chen, Y. Yu, S. Ponomarenko, A divergence-free meshless method based on the vector basis function for transient electromagnetic analysis, IEEE Transactions on Microwave Theory and Techniques 62 (7) (2014) 1409–1416. doi:10.1109/tmtt.2014.2322334.

[3] N. Z. Lima, R. C. Mesquita, Edge meshless method applied to vector electromagnetic problems, IEEE Transactions on Magnetics 53 (6) (2017) 1–4. doi:10.1109/TMAG.2017.2653769.

[4] J. F. Wang, Z. D. Chen, P. Cheng, J. Li, A time-domain meshless method without discretization in time, IEEE Microwave and Wireless Components Letters 30 (6) (2020) 545–548. doi:10.1109/lmwc.2020.2987733.

[5] L. A. G. Ortiz, R. C. Mesquita, N. Z. Lima, Nodal meshless method with vectorial shape functions based on h(curl), IEEE Transactions on Magnetics 57 (6) (2021) 1–4. doi:10.1109/TMAG.2021.3061526.

[6] A. Buffa, G. Sangalli, R. Vzquez, Isogeometric methods for computational electromagnetics: B-spline and t-spline discretizations, Journal of Computational Physics 257 (2014) 1291–1320, physics-compatible numerical methods. doi:https://doi.org/10.1016/j.jcp.2013.08.015.

[7] A. Buffa, R. Vzquez, Isogeometric analysis for electromagnetic scattering problems, 2014, cited by: 8. doi:10.1109/NEMO.2014.6995712.

[8] R. Simpson, Z. Liu, R. Vzquez, J. Evans, An isogeometric boundary element method for electromagnetic scattering with compatible b-spline discretizations, Journal of Computational Physics 362 (2018) 264 289, cited by: 50; All Open Access, Bronze Open Access. doi:10.1016/j.jcp.2018.01.025.

[9] A. Simona, L. Bonaventura, C. de Falco, S. Schps, Isogeometric approximations for electromagnetic problems in axisymmetric domains, Computer Methods in Applied Mechanics and Engineering 369, cited by: 7; All Open Access, Green Open Access (2020). doi:10.1016/j.cma.2020.113211.

[10] S. Kurz, S. Schps, G. Unger, F. Wolf, Solving maxwell's eigenvalue problem via isogeometric boundary elements and a contour integral method, Mathematical Methods in the Applied Sciences 44 (13) (2021) 10790 10803, cited by: 4; All Open Access, Green Open Access. doi:10.1002/mma.7447.

[11] J.-H. Lee, T. Xiao, Q. Liu, A 3-d spectral-element method using mixed-order curl conforming vector basis functions for electromagnetic fields, IEEE Transactions on Microwave Theory and Techniques 54 (1) (2006) 437–444. doi:10.1109/TMTT.2005.860502.

[12] N. Liu, L. E. Tobn, Y. Zhao, Y. Tang, Q. H. Liu, Mixed spectral-element method for 3-d maxwell's eigenvalue problem, IEEE Transactions on Microwave Theory and Techniques 63 (2) (2015) 317–325. doi:10.1109/TMTT.2014.2387839.

[13] Y. Zhou, L. Shi, N. Liu, C. Zhu, H. Liu, Q. H. Liu, Spectral element method and domain decomposition for low-frequency subsurface em simulation, IEEE Geoscience and Remote Sensing Letters 13 (4) (2016) 550–554. doi:10.1109/LGRS.2016.2524558.

[14] W. Wang, Z. Zhang, Spectral element methods for eigenvalue problems based on domain decomposition, SIAM Journal on Scientific Computing 44 (2) (2022) A689–A719. doi:10.1137/20M1345980.

[15] T. Wan, B. Tang, M. Li, An iteration-free domain decomposition method for the fast finite element analysis of electromagnetic problems, IEEE Transactions on Antennas and Propagation 68 (1) (2020) 400–410. doi:10.1109/TAP.2019.2943352.

[16] H.-X. Zhang, Q. Zhan, L. Huang, Y.-D. Wang, W.-J. Wang, Z. Qin, Z.-G. Zhao, D.-W. Wang, H.-J. Zhou, K. Kang, L. Zhou, W.-Y. Yin, A scalable hpc-based domain decomposition method for multiphysics modeling of rf devices, IEEE Transactions on Components, Packaging and Manufacturing Technology 11 (12) (2021) 2158–2170. doi:10.1109/TCPMT.2021.3121540.

[17] A. Amor-Martin, L. E. Garcia-Castillo, L. L. Tth, O. Floch, R. Dyczij-Edlinger, A rigorous code verification process of the domain decomposition method in a finite element method for electromagnetics, IEEE Transactions on Antennas and Propagation 72 (1) (2024) 100–109. doi:10.1109/TAP.2023.3294001.

[18] J. L. Volakis, J. L. Volakis, A. Chatterjee, L. C. Kempel, Finite element method for electromagnetics, Universities Press, 1998.

[19] K. Paulsen, D. Lynch, Elimination of vector parasites in finite element Maxwell solutions, IEEE Transactions on Microwave Theory and Techniques 39 (3) (1991) 395–404. doi:10.1109/22.75280.

[20] M. Koshiba, K. Hayata, M. Suzuki, Finite-Element Formulation in Terms of the Electric-Field Vector for Electromagnetic Waveguide Problems, IEEE Transactions on Microwave Theory and Techniques 33 (10) (1985) 900–905. doi:10.1109/TMTT.1985.1133148.

[21] R. Otin, Regularized Maxwell equations and nodal finite elements for electromagnetic field computations, Electromagnetics 30 (1-2) (2010) 190–204. doi:10.1080/02726340903485489.

[22] A. Nandy, C. S. Jog, An amplitude finite element formulation for electromagnetic radiation and scattering, Computers & Mathematics with Applications 71 (7) (2016) 1364–1391. doi:10.1016/j.camwa.2016.02.013.

[23] Arup Kumar, Robust finite element strategies for structures, acoustics, electromagnetics and magneto-hydrodynamics, Ph.D. thesis, Indian Institute of Science Banglore, Department of Mechanical Engineering, IISc, Banglore, India. (2016).

[24] A. Nandy, C. S. Jog, Conservation Properties of the Trapezoidal Rule for Linear Transient Electromagnetics, Journal of Advances in Mathematics and Computer Science 26 (4) (2018) 1–26. doi:10.9734/JAMCS/2018/39632.

[25] M. Agrawal, C. S. Jog, Monolithic formulation of electromechanical systems within the context of hybrid finite elements, Computational Mechanics 59 (3) (2017) 443–457. doi:10.1007/s00466-016-1356-1.

[26] A. Nandy, C. S. Jog, A monolithic finite-element formulation for magnetohydrodynamics, Sadhana - Academy Proceedings in Engineering Sciences 43 (9) (2018) 1–18. doi:10.1007/s12046-018-0905-z.

[27] C. S. Jog, A. Nandy, Mixed finite elements for electromagnetic analysis, Computers and Mathematics with Applications 68 (8) (2014) 887–902. doi:10.1016/j.camwa.2014.08.006.

[28] J. C. Nedelec, Mixed finite elements in $\mathbb{R}^3$, Numerische Mathematik 35 (3) (1980) 315–341. doi:10.1007/BF01396415.

[29] A. Bossavit, A rationale for 'edge-elements' in 3-D fields computations, IEEE Transactions on Magnetics 24 (1) (1988) 74–79. doi:10.1109/20.43860.

[30] Z. Cendes, Vector finite elements for electromagnetic field computation, IEEE Transactions on Magnetics 27 (5) (1991) 3958–3966. doi:10.1109/20.104970.

[31] D. Boffi, M. Farina, L. Gastaldi, On the approximation of Maxwell's eigenproblem in general 2D domains, Computers & Structures 79 (11) (2001) 1089–1096. doi:10.1016/S0045-7949(01)00003-7.

[32] D. Boffi, Finite element approximation of eigenvalue problems, Acta Numerica 19 (2010) 1–120. doi:10.1017/S0962492910000012.

[33] M. Ainsworth, J. Coyle, P. Ledger, K. Morgan, Computing maxwell eigenvalues by using higher order edge elements in three dimensions, IEEE Transactions on Magnetics 39 (5) (2003) 2149–2153. doi:10.1109/TMAG.2003.817097.

[34] L. E. Garcia-Castillo, M. Salazar-Palma, Second-order Nedelec tetrahedral element for computational electromagnetics, International Journal of Numerical Modelling: Electronic Networks, Devices and Fields 13 (2-3) (2000) 261–287. doi:10.1002/(SICI)1099-1204(200003/06)13:2/3<261::AID-JNM360>3.0.CO;2-L.

[35] J. H. Bramble, T. V. Kolev, J. E. Pasciak, The approximation of the Maxwell eigenvalue problem using a least-squares method, Mathematics of Computation 74 (252) (2005) 1575–1599. doi:10.1090/S0025-5718-05-01759-X.

[36] D. Kamireddy, A. Nandy, Creating edge element from four node quadrilateral element, IOP Conference Series: Materials Science and Engineering 1080 (1) (2021) 012015. doi:10.1088/1757-899x/1080/1/012015.

[37] A. Bossavit, J. C. Vérité, A mixed fem-biem method to solve 3-D eddy-current problems, IEEE Transactions on Magnetics 18 (2) (1982) 431–435. doi:10.1109/TMAG.1982.1061847.

[38] J. P. Webb, V. N. Kanellopoulos, Absorbing boundary conditions for the finite element solution of the vector wave equation, Microwave and Optical Technology Letters 2 (10) (1989) 370–372. doi:10.1002/mop.4650021010.

[39] X. Q. Sheng, J. M. Jin, C. C. Lu, W. C. Chew, On the formulation of hybrid finite-element and boundary-integral methods for 3-D scattering, IEEE Transactions on Antennas and Propagation 46 (3) (1998) 303–311. doi:10.1109/8.662648.

[40] Pekel, R. Mittra, An Application of the Perfectly Matched Layer (PML) Concept to the Finite Element Method Frequency Domain Analysis of Scattering Problems, IEEE Microwave and Guided Wave Letters 5 (8) (1995) 258–260. doi:10.1109/75.401074.

[41] E. S. Um, S.-S. Kim, H. Fu, A tetrahedral mesh generation approach for 3d marine controlled-source electromagnetic modeling, Computers & Geosciences 100 (2017) 1–9. doi:https://doi.org/10.1016/j.cageo.2016.11.007.

[42] O. Castillo-Reyes, J. de la Puente, J. M. Cela, Petgem: A parallel code for 3d csem forward modeling using edge finite elements, Computers & Geosciences 119 (2018) 123–136. doi:https://doi.org/10.1016/j.cageo.2018.07.005.

[43] H. Cai, B. Xiong, M. Han, M. Zhdanov, 3d controlled-source electromagnetic modeling in anisotropic medium using edge-based finite element method, Computers & Geosciences 73 (2014) 164–176. doi:https://doi.org/10.1016/j.cageo.2014.09.008.

[44] A. A. da Piedade, C. Rgis, C. M. B. Nunes, H. F. da Silva, Computational cost comparison between nodal and vector finite elements in the modeling of controlled source electromagnetic data using a direct solver, Computers & Geosciences 156 (2021) 104901. doi:https://doi.org/10.1016/j.cageo.2021.104901.

[45] L. J. Heagy, D. W. Oldenburg, Modeling electromagnetics on cylindrical meshes with applications to steel-cased wells, Computers & Geosciences 125 (2019) 115–130. doi:https://doi.org/10.1016/j.cageo.2018.11.010.

[46] C. M. B. Nunes, C. Rgis, Gemm3d: An edge finite element program for 3d modeling of electromagnetic fields and sensitivities for geophysical applications, Computers & Geosciences 139 (2020) 104477. doi:https://doi.org/10.1016/j.cageo.2020.104477.

[47] J. P. Webb, Edge Elements and What They Can Do for You, IEEE Transactions on Magnetics 29 (2) (1993) 1460–1465. doi:10.1109/CEFC.1992.720787.

[48] D. Boffi, P. Fernandes, L. Gastaldi, I. Perugia, Computational Models of Electromagnetic Resonators: Analysis of Edge Element Approximation, SIAM Journal on Numerical Analysis 36 (4) (1999) 1264–1290. doi:10.1137/S003614299731853X.

[49] C. J. Reddy, M. D. Deshpande, C. R. Cockrell, F. B. Beck, Finite Element Method for Eigenvalue Problems in Electromagnetics, NASA Technical Paper 3485 (December) (1994).

[50] M. L. Barton, Z. J. Cendes, New vector finite elements for three-dimensional magnetic field computation, Journal of Applied Physics 61 (8) (1987) 3919–3921. doi:10.1063/1.338584.

[51] T. V. Yioultsis, Multiparametric vector finite elements: a systematic approach to the construction of three-dimensional, higher order, tangential vector shape functions, IEEE Transactions on Magnetics 32 (3) (1996) 1389–1392. doi:10.1109/20.497506.

[52] T. Yioultsis, T. Tsiboukis, Development and implementation of second and third order vector finite elements in various 3-D electromagnetic field problems, IEEE Transactions on Magnetics 33 (2) (1997) 1812–1815. doi:10.1109/20.582630.

[53] F. Rapetti, A. Bossavit, Whitney Forms of Higher Degree, SIAM Journal on Numerical Analysis 47 (3) (2009) 2369–2386. doi:10.1137/070705489.

[54] J. F. Lee, D. K. Sun, Z. J. Cendes, Tangential vector finite elements for electromagnetic field computation, IEEE Transactions on Magnetics 27 (5) (1991) 4032–4035. doi:10.1109/20.104986.

[55] R. D. Graglia, D. R. Wilton, A. F. Peterson, Higher order interpolatory vector bases for computational electromagnetics, IEEE Transactions on Antennas and Propagation 45 (3) (1997) 329–342. doi:10.1109/8.558649.

[56] J. P. Webb, Hierarchal vector basis functions of arbitrary order for triangular and tetrahedral finite elements, IEEE Transactions on Antennas and Propagation 47 (8) (1999) 1244–1253. doi:10.1109/8.791939.

[57] Seung-Cheol Lee, Jin-Fa Lee, R. Lee, Hierarchical vector finite elements for analyzing waveguiding structures, IEEE Transactions on Microwave Theory and Techniques 51 (8) (2003) 1897–1905. doi:10.1109/TMTT.2003.815263.

[58] A. Ahagon, A. Kameari, New types of second-order edge element by reducing edge variables for electromagnetic field analysis, IEEE Transactions on Magnetics 53 (6) (2017) 1–4. doi:10.1109/TMAG.2017.2658680.

[59] A. Ahagon, A. Kameari, H. Ebrahimi, K. Fujiwara, Y. Takahashi, New type of second-order tetrahedral edge elements by reducing edge variables for quasi-static field analysis, IEEE Transactions on Magnetics 54 (3) (2018) 1–4. doi:10.1109/TMAG.2017.2749444.

[60] L. Garca-Castillo, M. Salazar-Palma, Second-order nedelec tetrahedral element for computational electromagnetics, International Journal of Numerical Modelling: Electronic Networks, Devices and Fields 13 (2-3) (2000) 261 287. doi:10.1002/(sici)1099-1204(200003/06)13:2/3<261::aid-jnm360>3.0.co;2-l.

[61] F. Fuentes, B. Keith, L. Demkowicz, S. Nagaraj, Orientation embedded high order shape functions for the exact sequence elements of all shapes, Computers & Mathematics with Applications 70 (4) (2015) 353–458. doi:https://doi.org/10.1016/j.camwa.2015.04.027.

[62] I. Anjam, J. Valdman, Fast matlab assembly of fem matrices in 2d and 3d: Edge elements, Applied Mathematics and Computation 267 (2015) 252–263, the Fourth European Seminar on Computing (ESCO 2014). doi:https://doi.org/10.1016/j.amc.2015.03.105.

[63] M. Olm, S. Badia, A. F. Martn, On a general implementation of h- and p-adaptive curl-conforming finite elements, Advances in Engineering Software 132 (2019) 74–91. doi:https://doi.org/10.1016/j.advengsoft.2019.03.006.

[64] J. Schberl, Netgen an advancing front 2d/3d-mesh generator based on abstract rules, Computing and Visualization in Science 1 (1997) 41–52, fourth International Conference on Finite Element Methods in Engineering and Sciences (FEMTEC 2013). doi:https://doi.org/10.1007/s007910050004.

[65] F. Hecht, New development in freefem+, Journal of Numerical Mathematics 20 (3-4) (2012) 251 265. doi:10.1515/jnum-2012-0013.

[66] P. Solin, L. Korous, P. Kus, Hermes2d, a c++ library for rapid development of adaptive hp-fem and hp-dg solvers, Journal of Computational and Applied Mathematics 270 (2014) 152–165, fourth International Conference on Finite Element Methods in Engineering and Sciences (FEMTEC 2013). doi:https://doi.org/10.1016/j.cam.2014.02.007.

[67] A. Chatterjee, J. Jin, J. L. Volakis, Edge-based finite elements and vector abcs applied to 3-d scattering, IEEE transactions on antennas and propagation 41 (2) (1993) 221–226.

[68] V. Kanellopoulos, J. Webb, A numerical study of vector absorbing boundary conditions for the finite-element solution of maxwell's equations, IEEE microwave and guided wave letters 1 (11) (1991) 325–327.

[69] B. Stupfel, Numerical implementation of second-and third-order conformal absorbing boundary conditions for the vector-wave equation, IEEE Transactions on Antennas and Propagation 45 (3) (1997) 487–492.

[70] J. Webb, V. Kanellopoulos, Absorbing boundary conditions for the finite element solution of the vector wave equation, Microwave and Optical Technology Letters 2 (10) (1989) 370–372.

[71] J.-P. Brenger, On the huygens absorbing boundary conditions for electromagnetics, J. Comput. Phys. 226 (2007) 354–378.

[72] S. Gupta, D. K. Chandrappa, A. Razik Sebak, 2-d inhomogeneous frequency dependent metasurface-cylinder based absorbing boundary conditions, in: 2019 13th European Conference on Antennas and Propagation (EuCAP), 2019, pp. 1–5.

[73] M. Y. S. Harwani, T. L. Yoon, Analyses of absorbing boundary conditions in 2d fdtd simulations for electromagnetic wave propagation in anisotropic ionosphere., Progress in Electromagnetics Research C 131 (2023) 103–117. doi:http://dx.doi.org/10.2528/PIERC22112302.

[74] M. M. Rana, A. S. Mohan, Convolutional perfectly matched layer abc for 3-d lod-fdtd using fundamental scheme, IEEE Microwave and Wireless Components Letters 23 (8) (2013) 388–390. doi:10.1109/LMWC.2013.2271296.

[75] J.-F. Wang, G. Li, Z. Chen, Convolutional implementation and analysis of the cfs-pml abc for the fdtd method based on wave equation, IEEE Microwave and Wireless Components Letters 32 (7) (2022) 811–814. doi:10.1109/LMWC.2022.3151497.

[76] F. N. Xiuzheng Fang, Di Wu, G. Yao, A new implementation of convolutional pml for second-order elastic wave equation, Exploration Geophysics 53 (5) (2022) 501–516. doi:10.1080/08123985.2021.1993441.

[77] J.-P. Brenger, A historical review of the absorbing boundary conditions for electromagnetics, in: Forum for Electromagnetic Research Methods and Application Technologies, Vol. 9, 2015, pp. 1–28.

[78] J.-P. Brenger, Theoretical investigation of the reflection from impedance absorbing boundary conditions, IEEE Microwave and Wireless Components Letters 28 (7) (2018) 543–545. doi:10.1109/LMWC.2018.2835775.

[79] J. Guan, S. Yan, J.-M. Jin, An accurate and efficient finite element-boundary integral method with gpu acceleration for 3-d electromagnetic analysis, IEEE Transactions on Antennas and Propagation 62 (12) (2014) 6325–6336. doi:10.1109/TAP.2014.2361896.

[80] T. Wan, M. Z. Li, L. F. Li, Direct solution of finite element-boundary integral system for electromagnetic analysis in half-space, IEEE Transactions on Antennas and Propagation 68 (8) (2020) 6461–6466. doi:10.1109/TAP.2020.2970024.

[81] E. M. Nelson, High accuracy electromagnetic field solvers for cylindrical waveguides and axisymmetric structures using the finite element method, Stanford University, 1994.

[82] U. van Rienen, T. Weiland, Triangular discretization method for the evaluation of rf-fields in cylindrically symmetric cavities, IEEE Transactions on Magnetics 21 (6) (1985) 2317–2320. doi:10.1109/TMAG.1985.1064183.

[83] D. Di Ruscio, P. Burghignoli, P. Baccarelli, D. Comite, A. Galli, Spectral method of moments for planar structures with azimuthal symmetry, IEEE Transactions on Antennas and Propagation 62 (4) (2014) 2317–2322. doi:10.1109/TAP.2014.2302831.

[84] V. Lombardi, M. Bozzi, L. Perregrini, Exploiting symmetries in the variational meshless method for 3-d inhomogeneous cavities, IEEE Transactions on Microwave Theory and Techniques 68 (2) (2020) 432–440. doi:10.1109/TMTT.2020.2964529.

[85] G. Garcia-Contreras, J. Crcoles, J. A. Ruiz-Cruz, Degeneracy-discriminating modal fem computation in higher order rotationally symmetric waveguides, IEEE Transactions on Antennas and Propagation 69 (11) (2021) 8003–8008. doi:10.1109/TAP.2021.3083790.

[86] G. Lin, T. Huang, X. Jin, Efficient waveguide port excitation for symmetric structures based on conformal non-uniform mesh, in: 2023 24th International Vacuum Electronics Conference (IVEC), 2023, pp. 1–2. doi:10.1109/IVEC56627.2023.10157805.

[87] C. S. Jog, A. Nandy, Mixed finite elements for electromagnetic analysis, Computers & Mathematics with Applications 68 (8) (2014) 887–902.

[88] R. Mittra, O. Ramahi, A. Khebir, R. Gordon, A. Kouki, A review of absorbing boundary conditions for two and three-dimensional electromagnetic scattering problems, IEEE Transactions on Magnetics 25 (4) (1989) 3034–3039.

[89] D. Pile, Numerical solution, Nature Photonics 9 (1) (2015) 5–6. doi:10.1038/nphoton.2014.305.

[90] R. S. Chen, D. Lei, Z. Ye, Zhenbao, Y. Yang, An efficient algorithm for implementing the Crank-Nicolson scheme in the mixed finite-element time-domain method, IEEE Transactions on Antennas and Propagation 57 (10 PART 2) (2009) 3216–3222. doi:10.1109/TAP.2009.2028675.

[91] J. F. Lee, Wetd-a finite element time-domain approach for solving maxwell's equations, IEEE Microwave and Guided Wave Letters 4 (1) (1994) 11–13. doi:10.1109/75.267679.

[92] M. F. Wong, O. Picon, V. F. Hanna, Finite Element Method Based On Whitney Forms, IEEE Transactions on Magnetics 31 (3) (1995) 4–7.

[93] M. Movahhedi, A. Abdipour, A. Nentchev, M. Dehghan, S. Selberherr, Alternating-direction implicit formulation of the finite-element time-domain method, IEEE Transactions on Microwave Theory and Techniques 55 (6) (2007) 1322–1331. doi:10.1109/TMTT.2007.897777.

[94] D. Jiao, A. A. Ergin, B. Shanker, E. Michielssen, J. M. Jins, A fast higher-order time-domain finite element-boundary integral method for 3-D electromagnetic scattering analysis, IEEE Transactions on Antennas and Propagation 50 (9) (2002) 1192–1202. doi:10.1109/TAP.2002.801375.

[95] J. Lee, R. Lee, A. Cangellaris, Time-domain finite-element methods, Antennas and Propagation, IEEE Transactions on 45 (3) (2002) 430–442.

[96] H. P. Tsai, Y. Wang, T. Itoh, An unconditionally stable extended (USE) finite-element time-domain solution of active nonlinear microwave circuits using perfectly matched layers, IEEE Transactions on Microwave Theory and Techniques 50 (10) (2002) 2226–2232. doi:10.1109/TMTT.2002.803442.

[97] A. Anees, L. Angermann, Time domain finite element method for maxwells equations, IEEE Access 7 (2019) 63852–63867. doi:10.1109/ACCESS.2019.2916394.

[98] D. S. Abraham, D. D. Giannacopoulos, A convolution-free finite-element time-domain method for the nonlinear dispersive vector wave equation, IEEE Transactions on Magnetics 55 (12) (2019) 1–4. doi:10.1109/TMAG.2019.2935681.

[99] Unconditionally stable time stepping method for mixed finite element maxwell solvers, Progress In Electromagnetics Research C 103 (2020) 17–30. doi:10.2528/PIERC20021001.

[100] F. Faghihi, H. Heydari, A combination of time domain finite element-boundary integral with time domain physical optics for calculation of electromagnetic scattering of 3-D structures, Progress in Electromagnetics Research 79 (2008) 463–474. doi:10.2528/PIER07110206.

[101] J. Liu, J. M. Jin, A novel hybridization of higher order finite element and boundary integral methods for electromagnetic scattering and radiation problems, IEEE Transactions on Antennas and Propagation 49 (12) (2001) 1794–1806. doi:10.1109/8.982462.

[102] Z. J. Qiu, J. D. Xu, G. Wei, X. Y. Hou, An improved time domain finite element-boundary integral scheme for electromagnetic scattering from 3-D objects, Progress In Electromagnetics Research 75 (2007) 119–135. doi:10.2528/PIER07053106.

[103] R. N. Rieben, G. H. Rodrigue, D. A. White, A high order mixed vector finite element method for solving the time dependent Maxwell equations on unstructured grids, Journal of Computational Physics 204 (2) (2005) 490–519. doi:10.1016/j.jcp.2004.10.030.

[104] G. Rodrigue, D. White, A vector finite element time-domain method for solving maxwell's equations on unstructured hexahedral grids, SIAM Journal on Scientific Computing 23 (3) (2001) 683–706. arXiv:https://doi.org/10.1137/S1064827598343826, doi:10.1137/S1064827598343826.

[105] D. Jiao, J. M. Jin, Three-dimensional orthogonal vector basis functions for time-domain finite element solution of vector wave equations, IEEE Transactions on Antennas and Propagation 51 (1) (2003) 59–66. doi:10.1109/TAP.2003.808524.

[106] A. Taflove, S. Hagness, Computational Electrodynamics: The Finite-difference Time-domain Method, Artech House antennas and propagation library, Artech House, 2005.

[107] J. Stratton, Electromagnetic Theory, IEEE Press Series on Electromagnetic Wave Theory, Wiley-IEEE Press, 2007.

[108] D. J. Griffiths, Introduction to Electrodynamics, 2nd Edition, Prentice Hall, New Jersey., 1999.

[109] A. Gupta, Recent advances in direct methods for solving unsymmetric sparse systems of linear equations, ACM Trans. Math. Softw. 28 (3) (2002) 301324. doi:10.1145/569147.569149.

[110] I. Bardi, O. Biro, K. Preis, Finite element scheme for 3D cavities without spurious modes, IEEE Transactions on Magnetics 27 (5) (1991) 4036–4039. doi:10.1109/20.104987.

[111] Wolfram Research, Inc., Mathematica 10.4.1, Champaign, IL (2020).

[112] A. Ahagon, T. Kashimoto, Three-dimensional electromagnetic wave analysis using high order edge elements, IEEE Transactions on Magnetics 31 (3) (1995) 1753–1756. doi:10.1109/20.376375.

[113] R. F. Harrington, Time-harmonic electromagnetic fields, McGraw-Hill, New York, NY, 1961.

[114] A. Elsherbeni, D. Kajfez, S. Zeng, Circular sectoral waveguides, IEEE Antennas and Propagation Magazine 33 (6) (1991) 20–27. doi:10.1109/74.107352.

[115] M. Dauge, Benchmark computations for Maxwell equations for the approximation of highly singular solutions, https://perso.univ-rennes1.fr/monique.dauge/benchmax.html (2020). URL https://perso.univ-rennes1.fr/monique.dauge/benchmax.html:

[116] K. Yee, Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media, IEEE Transactions on Antennas and Propagation 14 (3) (1966) 302–307. doi:10.1109/TAP.1966.1138693.

[117] A. C. Cangellaris, C. C. Lin, K. K. Mei, Point-Matched Time Domain Finite Element Methods for Electromagnetic Radiation and Scattering, IEEE Transactions on Antennas and Propagation 35 (10) (1987) 1160–1173. doi:10.1109/TAP.1987.1143981.

[118] D. R. Lynch, K. D. Paulsen, Time-Domain Integration of the Maxwell Equations on Finite Elements, IEEE Transactions on Antennas and Propagation 38 (12) (1990) 1933–1942. doi:10.1109/8.60982.

[119] J.-F. Lee, Z. Sacks, Whitney elements time domain (wetd) methods, IEEE Transactions on Magnetics 31 (3) (1995) 1325–1329. doi:10.1109/20.376272.

[120] D. Jiao, J. M. Jin, An effective algorithm for implementing perfectly matched layers in time-domain finite-element simulation of open-region EM problems, IEEE Transactions on Antennas and Propagation 50 (11) (2002) 1615–1623. doi:10.1109/TAP.2002.803987.

[121] T. Namiki, A new fdtd algorithm based on alternating-direction implicit method, IEEE Transactions on Microwave Theory and Techniques 47 (10) (1999) 2003–2007. doi:10.1109/22.795075.

[122] F. Zhen, Z. D. Chen, J. Zhang, Toward the development of a three-dimensional unconditionally stable finite-difference time-domain method, IEEE Transactions on Microwave Theory and Techniques 48 (9) (2000) 1550–1558. doi:10.1109/22.869007.

[123] F. Zheng, Z. Chen, J. Zhang, A finite-difference time-domain method without the courant stability conditions, IEEE Microwave and Guided Wave Letters 9 (11) (1999) 441–443. doi:10.1109/75.808026.

[124] T. Namiki, 3-d adi-fdtd method-unconditionally stable time-domain algorithm for solving full vector maxwell's equations, IEEE Transactions on Microwave Theory and Techniques 48 (10) (2000) 1743–1748. doi:10.1109/22.873904.

[125] E. L. Tan, Unconditionally stable lod-fdtd method for 3-d maxwell's equations, IEEE Microwave and Wireless Components Letters 17 (2) (2007) 85–87. doi:10.1109/LMWC.2006.890166.

[126] J. Shibayama, M. Muraki, J. Yamauchi, H. Nakano, Efficient implicit fdtd algorithm based on locally one-dimensional scheme, Electronics Letters 41 (19) (2005) 1. doi:10.1049/el:20052381.

[127] J. Lee, B. Fornberg, A split step approach for the 3-d maxwell's equations, Journal of Computational and Applied Mathematics 158 (2) (2003) 485–505. doi:https://doi.org/10.1016/S0377-0427(03)00484-9.

[128] G. Sun, C. Trueman, Unconditionally-stable fdtd method based on crank-nicolson scheme for solving three-dimensional maxwell equations, Electronics Letters 40 (10) (2004) 589–590. doi:10.1049/el:20040420.

[129] E. L. Tan, Efficient algorithms for cranknicolson-based finite-difference time-domain methods, IEEE Transactions on Microwave Theory and Techniques 56 (2) (2008) 408–413. doi:10.1109/TMTT.2007.914641.

[130] S. Cooke, M. Botton, T. Antonsen, B. Levush, A leapfrog formulation of the 3-d adi-fdtd algorithm, International Journal of Numerical Modelling: Electronic Networks, Devices and Fields 22 (2) (2009) 187–200. doi:10.1002/jnm.707.

[131] S.-C. Yang, Z. D. Chen, Y.-q. Yu, W.-Y. Yin, The unconditionally stable one-step leapfrog adi-fdtd method and its comparisons with other fdtd methods, IEEE Microwave and Wireless Components Letters 21 (12) (2011) 640–642. doi:10.1109/LMWC.2011.2173182.

[132] S.-C. Yang, Z. Chen, Y. Yu, W.-Y. Yin, An unconditionally stable one-step arbitrary-order leapfrog adi-fdtd method and its numerical properties, IEEE Transactions on Antennas and Propagation 60 (4) (2012) 1995–2003. doi:10.1109/TAP.2012.2186249.

[133] J. S. Kole, M. T. Figge, H. De Raedt, Unconditionally stable algorithms to solve the time-dependent Maxwell equations, Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics 64 (6) (2001) 14. arXiv:0107023, doi:10.1103/PhysRevE.64.066705.

[134] R. Luebbers, F. Hunsberger, Fdtd for nth-order dispersive media, IEEE Transactions on Antennas and Propagation 40 (11) (1992) 1297–1301. doi:10.1109/8.202707.

[135] D. Kelley, R. Luebbers, Piecewise linear recursive convolution for dispersive media using fdtd, IEEE Transactions on Antennas and Propagation 44 (6) (1996) 792–797. doi:10.1109/8.509882.

[136] P. M. Goorjian, A. Taflove, Direct time integration of maxwells equations in nonlinear dispersive media for propagation and scattering of femtosecond electromagnetic solitons, Optics letters 17 (3) (1992) 180–182. doi:10.1364/OL.17.000180.

[137] M. Okoniewski, M. Mrozowski, M. Stuchly, Simple treatment of multi-term dispersion in fdtd, IEEE Microwave and Guided Wave Letters 7 (5) (1997) 121–123. doi:10.1109/75.569723.

[138] T. Sekine, H. Asai, Mixed finite element time domain method based on iterative leapfrog scheme for fast simulations of electromagnetic problems, IEEE International Symposium on Electromagnetic Compatibility (2011) 596–601doi:10.1109/ISEMC.2011.6038381.

[139] C. Ma, Finite-element method for time-dependent Maxwell's equations based on an explicit-magnetic-field scheme, Journal of Computational and Applied Mathematics 194 (2) (2006) 409–424. doi:10.1016/j.cam.2005.08.008.

[140] M. Feliziani, F. Maradei, An explicit-implicit solution scheme to analyze fast transients by finite elements, IEEE Transactions on Magnetics 33 (2) (1997) 1452–1455. doi:10.1109/20.582533.

[141] W. A. Artuzi, Improving the Newmark time integration scheme in finite element time domain methods, IEEE Microwave and Wireless Components Letters 15 (12) (2005) 898–900. doi:10.1109/LMWC.2005.859968.

[142] S. Gedney, U. Navsariwala, An unconditionally stable finite element time-domain solution of the vector wave equation, IEEE Microwave and Guided Wave Letters 5 (10) (1995) 332–334. doi:10.1109/75.465046.

[143] D. Jiao, J. M. Jin, A general approach for the stability analysis of the time-domain finite-element method for electromagnetic simulations, IEEE Transactions on Antennas and Propagation 50 (11) (2002) 1624–1632. doi:10.1109/TAP.2002.803965.

[144] D. Jiao, J. M. Jin, Time-domain finite-element modeling of dispersive media, IEEE Microwave and Wireless Components Letters 11 (5) (2001) 220–222. doi:10.1109/7260.923034.

[145] W. Chen, X. Li, D. Liang, Energy-conserved splitting finite-difference time-domain methods for maxwell's equations in three dimensions, SIAM Journal on Numerical Analysis 48 (4) (2010) 1530–1554. arXiv:https://doi.org/10.1137/090765857, doi:10.1137/090765857.

[146] D. Kamireddy, A. Nandy, A novel conversion technique from nodal to edge finite element data structure for electromagnetic analysis, Computer Assisted Methods in Engineering and Science 28 (4) (2022) 291–319, available online https://cames.ippt.gov.pl/index.php/cames/article/view/384. doi:10.24423/cames.384.

[147] D. Jiao, J. Jin, E. Michielssen, D. Riley, Time-domain finite-element simulation of three-dimensional scattering and radiation problems using perfectly matched layers, in: IEEE Antennas and Propagation Society International Symposium (IEEE Cat. No.02CH37313), Vol. 2, 2002, pp. 158–161 vol.2. doi:10.1109/APS.2002.1016051.

[148] C. H. Wilcox, An expansion theorem for electromagnetic fields, Communications on Pure and Applied Mathematics 9 (2) (1956) 115–134. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.3160090202, doi:https://doi.org/10.1002/cpa.3160090202.

[149] Z. Cendes, The development of hfss, in: 2016 USNC-URSI Radio Science Meeting, 2016, pp. 39–40. doi:10.1109/USNC-URSI.2016.7588501.