2004-05-06

MoM modeling of metal-dielectric structures using volume integral equations

Shashank Dilip Kulkarni
Worcester Polytechnic Institute

MoM Modeling of Metal-Dielectric Structures via Volume Integral Equation

by

Shashank Kulkarni

Thesis submitted to the Faculty of the

Worcester Polytechnic Institute

in partial fulfillment of the requirements for the

Degree of Master of Science

in Electrical and Computer Engineering

6th May 2004

Approved by MS Thesis Committee Members

Dr. Sergey Makarov Dr. Reinhold Ludwig

Dr. David Cyganski Dr. Brian King

Abstract

Modeling of patch antennas and resonators on arbitrary dielectric substrates using surface RWG and volume edge based basis functions and the Method of Moments (MoM) is implemented in Matlab and C/C++. The performance of the solver is studied for different mesh configurations. The results obtained are tested by comparison with analytical solutions (Mie theory) and Ansoft HFSS v9 simulator. The latter uses a large number of finite elements (up to 200,000) and adaptive mesh refinement, thus providing the reliable data for comparison.

The error in the resonant frequency is estimated for canonical resonator structures at different values of the relative dielectric constant ε_r, which ranges from 1 to 200. The reported results show a near perfect agreement in the estimation of resonant frequency for all the metal-dielectric resonators.

Behavior of the antenna input impedance is tested, close to the first resonant frequency for the patch antenna. The error in the resonant frequency is estimated for different structures at different values of the relative dielectric constant ε_r , which ranges from 1 to 10. A larger error is observed in the calculation of the resonant frequency of the patch antenna. Moreover, this error increases with increase in the dielectric constant of the substrate. Further scope for improvement lies in the investigation of this effect.

Acknowledgements:

I would like to thank my adviser Prof. S. Makarov and the ECE Department of WPI for support of this work. I would also like to thank Andy Li for building the prototype antenna structure and Rostislav Lemdiasov for providing the code for the Mie series solution. I would like to thank all my family members for the good things they did for me.

Table of Contents

List of Tables

1. Introduction

The MoM solution for patch antennas can be obtained using the method of the volume integral equation [1]. The method of the volume integral equation has a number of advantages including applicability to arbitrary inhomogeneous dielectric materials [2] as well as a potentially better accuracy at resonances (compared to the surface integral formulation [3]). At the same time, it suffers from a rapid growth of computational complexity with an increase in the size of the grid. Therefore, possible reduction of the number of basis functions (unknowns) and simplification of the basis functions themselves may improve the performance of the method.

The simplest choice for the volume integral equation is the pulse basis functions [4]. However, they tend to be unstable when relative permittivity becomes higher [1]. The face-based tetrahedral basis functions proposed by Schaubert, Wilton, and Glisson [2] are more robust and are increasingly used today [3, 5].

A convenient choice is to select the electric flux density $\vec{D}$ as the unknown for the volume integral equation [2]. Then, the continuity of the normal component of $\vec{D}$ must be enforced on the faces within the same basis function. This is in contrast to the finite element method with the edge-based vector basis functions, where the continuity of the tangential $\vec{E}$-field is required on the faces [6-8].

The face-based vector functions for the volume integral equation [2] are built from two tetrahedra sharing the same face; with a linear $\vec{D}$ variation essentially in the direction of the face normal. These basis functions exactly replicate any constant $\vec{D}$-flux, but not its linear variation. The number of unknowns is equal to the number of the faces of the tetrahedral mesh. Note that, for a non-trivial tetrahedral mesh, the number of faces is larger than the number of the edges. This is a disadvantage compared to the edge-based functions, where the number of *independent* unknowns for the system matrix may be even smaller than the number of the edges [9, 10]. It is therefore inviting to employ edge based basis functions that still acquire the condition of the continuous normal $\vec{D}$ - component according to [2], but include all tetrahedra sharing the same edge – similar to the edge-based divergence-free basis functions [7].

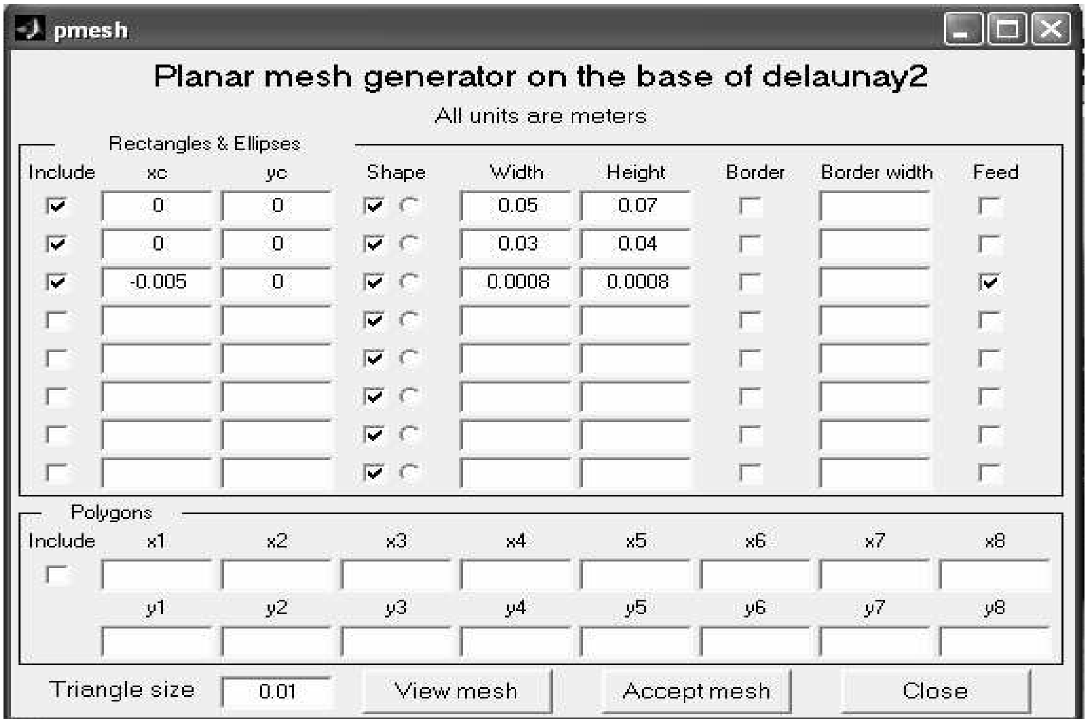
This thesis aims at simulation of combined metal-dielectric structures using the Method of Moments based on surface Rao-Wilton-Glisson (RWG) and volume edge based basis functions. The next section gives a general overview of the practical implementation of the MoM solver.

1.1 Solver overview

The entire solver is split up into three directories:

a. 1_mesh

The objectives of this directory are to specify the geometry of the structure and mesh generation. Below the functional layout of the mesh generator is represented. The GUI interface and some blocks in this directory were programmed by Mr. Andrew Marut, a student in the ECE department at WPI. The directory consists of two files `pmesh.m` and vmesh.m. On execution, the file pmesh.m prompts the user to specify the two dimensional geometry of the structure as shown in Fig. 1.2.1.



Planar mesh generator on the base of delaunay2

All units are meters

Rectangles & Ellipses

Include	xc	yc	Shape	Width	Height	Border	Border width	Feed
☑	0	0	☑ ○	0.05	0.07	☐		☐
☑	0	0	☑ ○	0.03	0.04	☐		☐
☑	-0.005	0	☑ ○	0.0008	0.0008	☐		☑
☐			☑ ○			☐		☐
☐			☑ ○			☐		☐
☐			☑ ○			☐		☐
☐			☑ ○			☐		☐
☐			☑ ○			☐		☐

Polygons

Include	x1	x2	x3	x4	x5	x6	x7	x8
☐								
	y1	y2	y3	y4	y5	y6	y7	y8

Triangle size 0.01

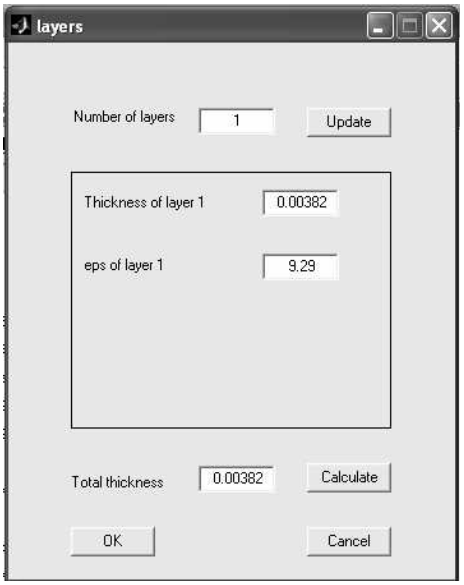
Fig 1.1.1 Specifying the 2D geometry of the structure.

Once the geometry is specified the file generates a two dimensional triangular mesh by using the Matlab built-in mesh generator ‘delaunay’. The 2D geometry is saved in pmesh.mat in terms of the t and P matrices which correspond to the vertices of the triangles and their x and y coordinates respectively for a given structure.

P (2, p) – array of Cartesian node coordinates x and y; the number of nodes is p.

t (4, N) – array of node numbers for each triangle; number of triangles is N. The fourth row in the triangle array is the domain number.

The `P` and `t` matrices are the input to the file `vmesh.m`. On execution, the file `vmesh.m` prompts the user to determine the thickness of the structure and the dielectric constant of the material as shown in Fig. 1.1.2.



Number of layers 1

Thickness of layer 1 0.00382

eps of layer 1 9.29

Total thickness 0.00382

Fig 1.1.2 Specifying the 3 dimensional geometry of the structure

With this data, the file generates a three dimensional tetrahedral mesh by using the Matlab mesh generator ‘delaunay3’. The 3D geometry is saved in `vmesh.mat` in the form of `T` and `P` matrices which correspond to the vertices of the tetrahedra and their x, y and z coordinates respectively for the given structure. The `t` matrix for the triangles is also saved. In addition to the mesh geometry the unique edges for the triangles and the tetrahedra are also saved. These are then used to define the basis functions. All the functions in this directory are .m files so that the GUI of Matlab can be used. On successful execution of `vmesh.m` a new window is generated which provides the option to view the surface, volume or the feed mesh as shown in Fig. 1.1.3.

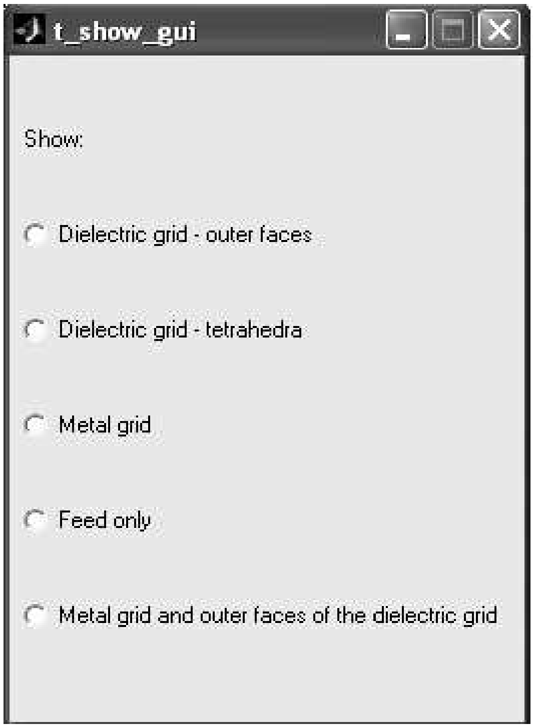

t_show_gui
Show:
○ Dielectric grid - outer faces
○ Dielectric grid - tetrahedra
○ Metal grid
○ Feed only
○ Metal grid and outer faces of the dielectric grid

Fig 1.1.3 GUI for viewing the meshes

b. 2_basis

The objective of this directory is to define the surface RWG and volume edge based basis functions for the metal surface and dielectric volume respectively. The directory consists of two functions `rwgm` for the surface RWG basis functions and `rwgd` for the volume edge based basis functions. The function `rwgm` returns a structure `geom` which defines all the parameters of the metal triangles including potential integrals. The potential integrals are calculated using the function `analytma` and `analytmb`. The function `analytma` returns the analytically calculated potential integrals for the metal triangle with itself. The function `analytmb` returns the numerically calculated potential integrals for the metal triangle with other metal triangles. The numerical integration is obtained by using the Gaussian integration points. The function `rwgd` returns a structure `GEOM` which defines all the parameters of the dielectric tetrahedra including potential integrals. The potential integrals for dielectric structure are calculated numerically by using the Gaussian integration points for both the face and the tetrahedra. This is implemented in the function `analytd` for the volume integrals and `analtys` for the surface integrals. All the functions in this directory are C files compiled as dynamically linked libraries (.dll) for optimizing the speed and the memory usage. Both the structures `geom` and `GEOM` are saved in `rwg.mat` which is then used to solve the MoM equations.

c. 3_mom

The objective of this directory is to solve the liner system of MoM equations. This includes the calculation of the impedance matrix and the voltage vector to finally obtain the unknown currents or the fields in the structure. The calculation of the impedance matrix is split up into three parts. The function `zmm` calculates the metal–to-metal impedance matrix; the function `zdd` calculates the dielectric-to-dielectric impedance matrix while the function `zmd` calculates the effect of the metal-dielectric interaction. For the scattering problem the voltage vectors in the metal and dielectric are calculated using the functions `voltagem` and `voltaged` respectively. For the radiation problem the feed voltages are determined by using the function `infield`. The system of equations is solved by using the function `slv`. The function `slv` uses LAPACK function `zsysv` (`zsysv_` for LINUX) for solving equations of the form [Z] [I] = [V] where Z is a complex symmetric matrix. Once the unknown currents are obtained the output parameters such as input impedance, return loss, near fields and radiation patterns can be plotted. All the functions in the directory are C files compiled as `.dll` for memory and speed optimization.

Detailed information regarding these functions is given below in table 1.1.1. The derivation for calculation of the impedance matrix, voltage vectors and fields is explained in detail in sections 2 and 3.

Table 1.1.1 Description of the functions used in the MoM solver

File name	Function call	Description
rwgm.cpp	[geom] = rwgm (P, t, IndexF, Edgesm, Tp, Tm, Vp, Vm)	The function rwgm returns a structure `geom` which defines all the parameters of the metal triangles including potential integrals. The input parameters to the function are `IndexF` which defines the number of integration points and the `t` and `P` matrices which correspond to the vertices of the triangles and their x, y and z coordinates respectively for a given structure. The other inputs are `Edgesm` which correspond to the unique edges in the metal structure, `Tp` and `Tm` which correspond to the plus and minus triangles and `Vp` and `Vm` which correspond to the plus and minus vertices of the metal structure respectively. The last four parameters are calculated in the 1_mesh directory and saved in `vmesh.mat`. The self integrals are calculated using the function `analytma` and `analytmb`. The function `analytma` returns the analytically calculated potential integrals for the metal face with itself. The function `analytmb` returns the numerically calculated potential integrals for the metal face with other metal face. The numerical integration is obtained by considering the Gaussian integration points.
rwgd.cpp	[GEOM] = rwgd (P, T, Faces, IndexF, IndexT, const, geom, Edges, AT)	The function rwgd returns a structure `GEOM` which defines all the parameters of the dielectric tetrahedra including potential integrals. The self integrals for dielectric are calculated numerically by using the Gaussian integration points for both the face and the tetrahedra. This is implemented in the function `analytd` for the volume integrals and `analtys` for the surface integrals. The inputs to the function are two structures `geom`

		and const. The structure geom is obtained from rwgm and is used only for combined metal-dielectric structures. The structure const defines the electromagnetic constants such as μ, ε, c (speed of light) etc. In addition to these, the other input parameters include IndexF which defines the number of integration points for the faces, IndexT which defines the number of integration points for the volume tetrahedra, the coordinates of the faces and the T and P matrices which correspond to the vertices of the triangles and their x, y and z coordinates respectively for a given structure. The last two parameters are Edges which correspond to the unique edges of the tetrahedra and AT which corresponds to the associated faces i.e. the face shared by the two tetrahedra. The last two parameters are calculated in the 1_mesh directory and saved in vmesh.mat.
zmm.cpp	z = zmm (geom, const, frequency)	The function zmm returns matrix ZMM (geom. Edgestotal by geom. Edgestotal) which correspond to the impedance matrix for pure metal. The inputs to the function are two structures geom and const and the frequency of operation. The structure geom is obtained form rwgm.
zdd.cpp	ZDD = zdd (GEOM, const, omega_re, omega_im)	The function zdd returns matrix ZDD (GEOM. Edgestotal by GEOM. Edgestotal) which correspond to the impedance matrix for pure dielectric structure. The inputs to the function are two structures GEOM and const and the real and imaginary components of the frequency of operation. The structure GEOM is obtained from rwgd.
infield.cpp	[EfieldM EfieldD Index] = infield (geom, GEOM, const, omega_re, omega_im)	The function infield returns matrix EfieldM(3, geom. IndexF,geom.Trianglestotal), EfieldD(3, GEOM. IndexT, GEOM.Tetrahedratotal) and Index(4,1).EfieldM and EfieldD correspond to the electric fields generated in the feed due to the applied voltage (magnetic ring).The vector Index provides the index of the feed edges. The inputs to the function are three structures geom, GEOM and const in addition to the real and imaginary frequencies.

voltagem.cpp	VM = voltagem (geom, EfieldM)	The function `voltagem` returns a voltage vector '`VM`' (`geom.Edgestotal, 1`) which corresponds to the induced voltage in the metal structure generated due to the applied voltage at the feed (magnetic ring). The inputs to the function are structure `geom` and `EfieldM`. `EfieldM` corresponds to the field generated in the metal structure due to the applied voltage and is obtained from infield.
voltaged.cpp	VD = voltaged (GEOM, EfieldD)	The function `voltaged` returns a voltage vector `VD` (`GEOM.Edgestotal,1`) which corresponds to the induced voltage in the dielectric structure generated due to the applied voltage at the feed (magnetic ring). The inputs to the function are structure `GEOM` and `EfieldD`. `EfieldD` corresponds to the field generated in the dielectric structure due to the applied voltage and is obtained from infield.

slv.cpp	[I D] = slv (ZMM, ZDD, ZMD, VM, VD, omega_re, omega_im)	The function `slv` returns the current vector `I` (1, geom.Edgestotal) and D (1, `GEOM.EdgesTotal`) in the metal and dielectric respectively. The inputs to the function are the three impedance matrices
		`ZMM(geom.Edgestotal,geom.Edgestotal)`
		`ZDD(GEOM.Edgestotal,GEOM.Edgestotal)`
		`ZMD(geom.Edgestotal,GEOM.Edgestotal)`
		obtained from the functions `zmm`, `zdd` and `zmd` respectively. In addition, there are the two voltage vectors `VM` (1, `geom.Edgestotal`) and `VD` (1, `GEOM.Edgestotal`) obtained from the functions `voltagem` and `voltaged` and the real and imaginary part of the frequency at which the computations are to be carried out. The function `slv` uses LAPACK function `zsysv` (`zsysv_` for LINUX) for solving equations of the form [Z][I] = [V] where Z is a complex symmetric matrix.

fieldm.cpp	[Poynting, E, H] = fieldm (const, geom, I, f_re, Point, option)	The function `fieldm` computes the scattered electric field, `E` (3, Points where the field is calculated), magnetic field `H` (3, Points where the field is calculated) and the Poynting vector `Poynting` (3, Points where the field is calculated) for a metal structure. The inputs to the function are two structures `geom` and `const`. In addition to these, the other input parameters include the current vector, `I`, which is obtained using `slv`, frequency of operation, observation point where the field is to be measured and an addition parameter called `option`. If

		`option` is set to zero then the scattered filed is calculated while when the option is set to one the total field is calculated.
fieldd.cpp	[Poynting, E, H] = fieldd (const, GEOM, D, frequency, Points, option)	The function `fieldd` computes the scattered electric field, `E` (3, Points where the field is calculated), magnetic field, `H` (3, Points where the field is calculated) and the Poynting vector, `Poynting` (3, Points where the field is calculated) for a dielectric structure. The inputs to the function are two structures `GEOM` and `const`. In addition to these, the other input parameters include the current vector, `D`, in the dielectric which is obtained using `slv`, frequency of operation, the observation points where the field is to be measured and an addition parameter called `option`. If `option` is set to zero then the scattered field is calculated while when the option is set to one the total field is calculated.

The solver is tested for different structures, e.g. homogenous and inhomogeneous dielectric structures (section 5.1); resonators (section 5.2) and patch antennas (section 5.3). The solver is also tested by comparison with an analytical solution for dielectric scattering problem (Mie theory) in Appendix A.

The performance of the solver (radiation/scattering) is studied for different mesh configurations and different dielectric constants of the substrate. A typical structure has around 700 surface RWG basis functions and around 2,500 volume edge based basis functions. The time required for calculation of a single frequency step for the above structure is about 2 minutes on a P IV 3GHz processor. At present the solver can handle around 8,500 unknowns (metal + dielectric).

1.2 Review of common basis functions

Before going into the derivation and implementation of the MoM equations for the surface RWG and the edge based basis functions, we summarize the commonly used basis functions to simulate antennas. The basis functions chosen should have the ability to accurately represent and resemble the anticipated unknown field distribution, while minimizing the computational effort required when employing it.

a) Pulse basis function

It is defined as [11]

$$f_n(x) = \begin{cases} 1 & x_{n-1} \le x \le x_n \\ 0 & elsewhere \end{cases} \tag{1.2.1}$$

A sketch is given in Fig 1.2.1.

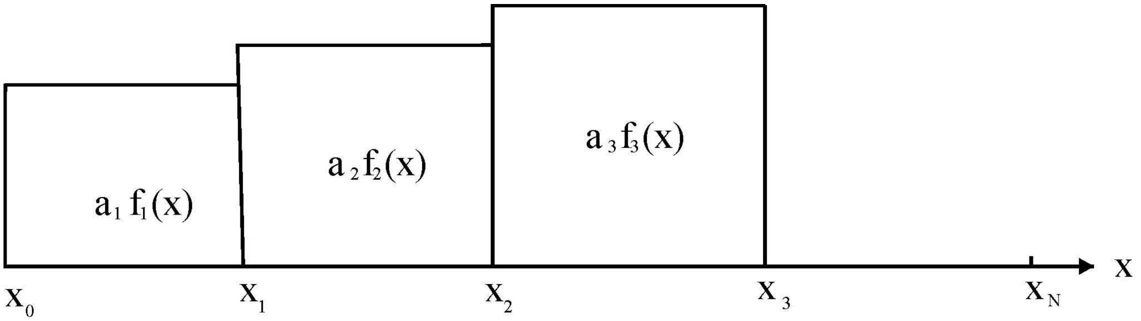


Fig 1.2.1 Pulse basis function.

b) Rooftop basis functions

Piecewise linear functions defined on rectangular sub-domain are referred as rooftop basis functions or RT [12]. In the simplest one dimensional case one has

$$f_n(x) = \begin{cases} \dfrac{x - x_{n-1}}{x_n - x_{n-1}}; & x_{n-1} \le x \le x_n \\ \dfrac{x_{n+1} - x}{x_{n+1} - x_n}; & x_n \le x \le x_{n+1} \\ 0; & elsewhere \end{cases} \tag{1.2.2}$$

A sketch is given in Fig 1.2.2.

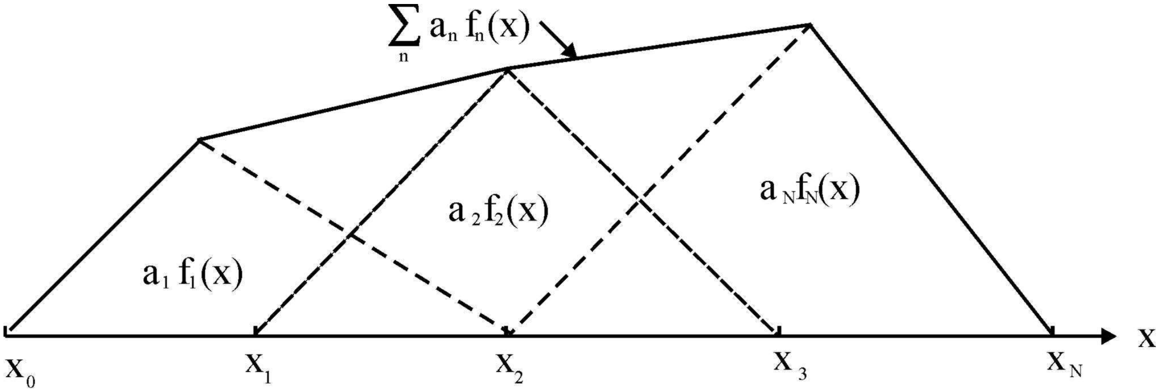


Fig 1.2.2 Rooftop basis function.

They are commonly used in the numerical solution of surface integral equations. The rooftop basis functions are well suited to modeling geometries that conform to Cartesian coordinates. These functions are much smoother than the pulse basis functions. There is also a 2D extension of the rooftop basis functions. In this case the basis functions are defined on two neighboring sub domains and the unknown is associated with the common edge between these sub domains.

c) Sinusoidal basis functions

Wire antennas can be solved using a moment's solution where the method of subsectional basis is applied with both the expansion and testing functions being sinusoidal distributions. It is defined as [11]

$$f_n(x) = \begin{cases} \dfrac{\sin[k(x - x_{n-1})]}{\sin[k(x_n - x_{n-1})]}; & x_{n-1} \leq x \leq x_n \\ \dfrac{\sin[k(x_{n+1} - x)]}{\sin[k(x_{n+1} - x_n)]}; & x_n \leq x \leq x_{n+1} \\ 0; & \textit{elsewhere} \end{cases} \tag{1.2.3}$$

A sketch is given in Fig 1.2.3.

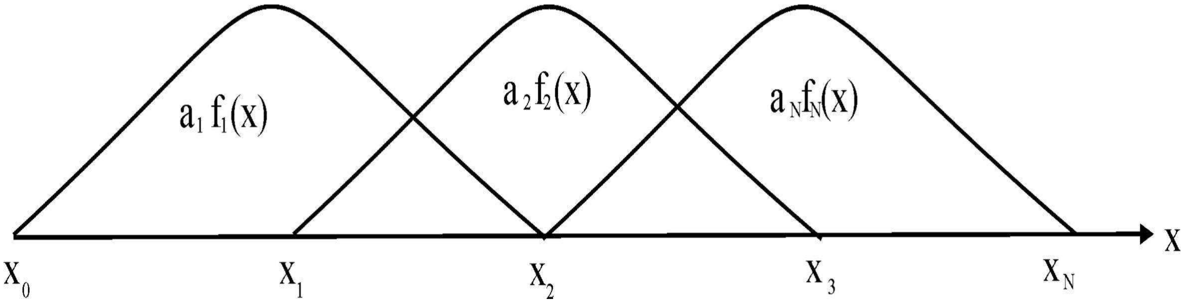


Fig 1.2.3 Sinusoidal basis function.

c) Rao-Wilton –Glisson (RWG) basis functions

RWG basis functions are capable of modeling flat faced approximations of arbitrary geometries. These are also edge functions. For any two triangular patches, t_n^+ and t_n^-, having areas A_n^+ and A_n^-, respectively, and sharing the common edge l_n, the n^{th} basis function is defined as [13]

$$f_n(\vec{r}) = \begin{cases} \dfrac{l_n}{2A_n^+}\vec{\rho}_n^{+S}; & \vec{r}\ in\ t_n^+ \\ \dfrac{l_n}{2A_n^-}\vec{\rho}_n^{-S}; & \vec{r}\ in\ t_n^- \end{cases} \qquad (1.2.4)$$

where $\vec{\rho}_n^{+S} = \vec{r} - \vec{r}_n^+$ is the vector drawn from the free vertex of triangle t_n^+ to the observation point; $\vec{\rho}_n^{-S} = \vec{r}_n^- - \vec{r}$ is the vector drawn from the observation point to the free vertex of triangle t_n^-. The basis function is zero outside two adjacent triangles t_n^+ and t_n^-. A sketch of these basis functions is given in Fig 1.2.4.

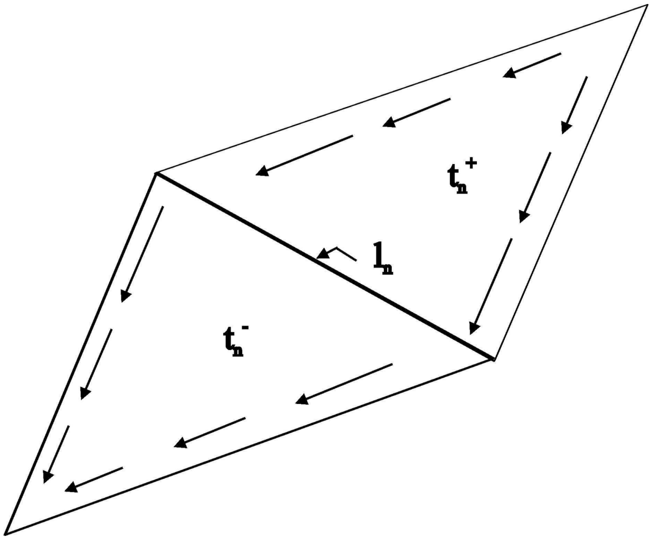


Fig 1.2.4 Surface RWG basis function.

e) Face-based basis functions

Volume RWG basis functions introduced in [2] are very similar to the surface RWG basis functions. Instead of two adjacent triangular patches sharing the common edge, one needs to consider two adjacent tetrahedra sharing the common face. For any two tetrahedra, T_n^+ and T_n^-, having volumes V_n^+ and V_n^-, respectively, and sharing the common face a_n, the n^{th} basis function becomes

$$f_n(\vec{r}) = \begin{cases} \dfrac{a_n}{3V_n^+}\vec{\rho}_n^{+V}; & \vec{r} \ in \ T_n^+ \\ \dfrac{a_n}{3V_n^-}\vec{\rho}_n^{-V}; & \vec{r} \ in \ T_n^- \end{cases} \tag{1.2.5}$$

where $\vec{\rho}_n^{+V} = \vec{r} - \vec{r}_n^{\,+}$ is the vector drawn from free vertex of tetrahedron T_n^+ to the observation point; $\vec{\rho}_n^{-V} = \vec{r}_n^{\,-} - \vec{r}$ is the vector drawn from the observation point to the free vertex of tetrahedron T_n^-. The basis function is zero outside two adjacent tetrahedra T_n^+ and T_n^-. A sketch of the basis function is given in Fig 1.2.5.

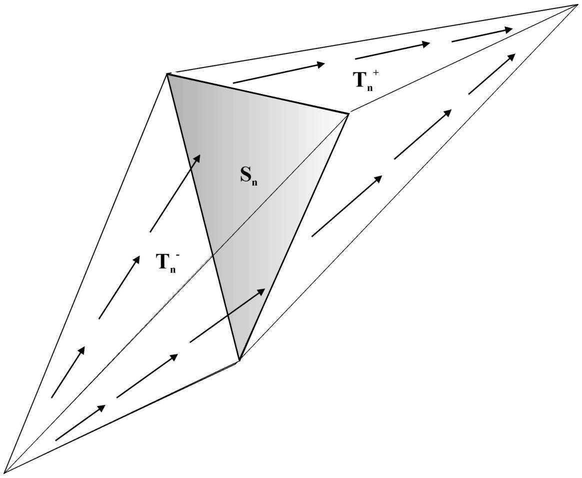


Fig 1.2.5 Volume RWG basis function.

e) Edge-based basis functions

The edge-based basis function $\vec{f}$ introduced in [14] is shown in Fig.1.2.6. The basis function is defined by a vector of the edge $\vec{p}$ (or *CD*), which is opposite to the base edge $\vec{l}$. The vector variation is essentially perpendicular to the base edge $\vec{l}$ (or *AB*). Within a tetrahedron, the basis function is a constant field given by $\vec{f} = c\vec{p}$ where *c* is a normalization coefficient.

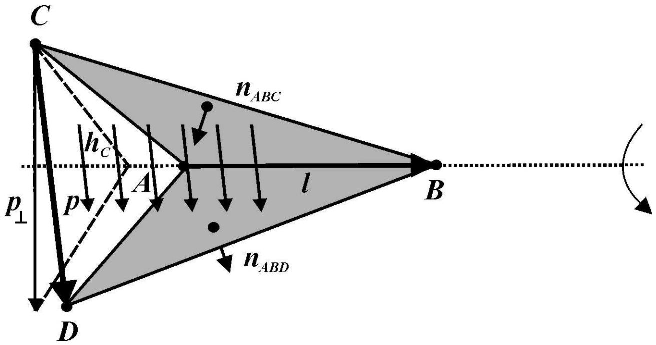


Fig 1.2.6 Edge based basis functions

1.3 RWG basis functions

In this section we discuss the surface RWG basis functions used for modeling a metal surface.

a. Definition

A metal surface is divided into triangular patches as shown in Fig.1.3.1.

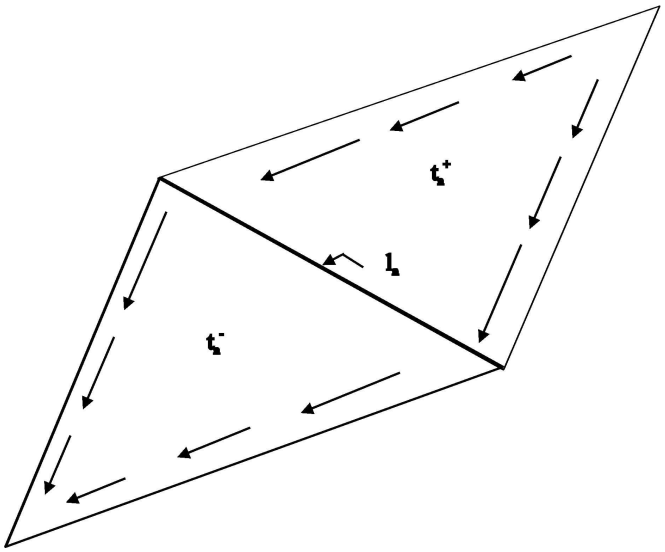


Fig 1.3.1 Surface RWG basis function

For any two triangular patches, t_n^+ and t_n^-, having areas A_n^+ and A_n^-, respectively, and sharing the common edge l_n, the n^{th} basis function is defined as

$$f_n^S(\vec{r}) = \begin{cases} \dfrac{l_n}{2A_n^+}\vec{\rho}_n^{+S}; & \vec{r} \ in \ t_n^+ \\ \dfrac{l_n}{2A_n^-}\vec{\rho}_n^{-S}; & \vec{r} \ in \ t_n^- \end{cases} \qquad (1.3.1)$$

where $\vec{\rho}_n^{+S} = \vec{r} - \vec{r}_n^{+}$ is the vector drawn from free vertex of triangle t_n^{+} to the observation point; $\vec{\rho}_n^{-S} = \vec{r}_n^{-} - \vec{r}$ is the vector drawn from the observation point to the free vertex of triangle t_n^{-}. The basis function is zero outside two adjacent triangles t_n^{+} and t_n^{-}.

1.4 Volume edge based basis functions

In this section we discuss the edge based basis functions used for modeling the dielectric volume.

a. Definition

The edge-based basis function $\vec{f}$ introduced in [14] is shown in Fig. 1.4.1 a. It is similar to the first Whitney form [7]. However, the vector variation is essentially perpendicular to the base edge $\vec{l}$ (or *AB*). The basis function is defined by a vector of the edge $\vec{p}$ (or *CD*), which is opposite to the base edge $\vec{l}$. Within a tetrahedron, the basis function is a constant field given by $\vec{f} = c\vec{p}$ where *c* is a normalization coefficient.

The basis function may include a different number of tetrahedra that share the same base edge $\vec{l}$. Three representative cases are depicted in Fig. 1.4.1. In the first case (Fig. 1.4.1a); both grayed faces of a tetrahedron are on the mesh boundary. The basis function includes only one tetrahedron. In the second case (Fig. 1.4.1b); two faces of two adjacent tetrahedra are on the mesh boundary. The basis function includes two tetrahedra and has one inner face. In the last case (Fig. 1.4.1c); all faces of all tetrahedra sharing the base edge are the inner faces of the mesh. The basis function only has the inner faces.

The component of the basis function $\vec{f}$ normal to face *ABC* in Fig. 1.4.1a is given by

$$f_{\perp} = c\vec{p} \cdot \vec{n}_{ABC} = c\vec{p}_{\perp} \cdot \vec{n}_{ABC} = 2c\bar{S} / h_C = cl\bar{S} / S_{ABC} \tag{1.4.1}$$

where $\vec{p}_{\perp}$ is the projection of $\vec{p}$ onto a plane perpendicular to the base edge; h_C is the height of triangle *ABC* perpendicular to the base edge; and $\bar{S}$ is the area of the projection of triangle *ACD* or triangle *BCD* onto a plane perpendicular to the base edge. The normalization coefficient is chosen in the form $c = 1/(l\bar{S})$. This guarantees that (i) the normal component of the basis function is continuous through the inner faces; and (ii) the total flux of the normal component through any face is equal to one.

$$f_{\perp} = \frac{1}{S_{ABC}} \tag{1.4.2}$$

b. Size of the function set

A naive guess is to assume that the number of basis functions is equal to the number of edges N_L of the tetrahedral mesh. This approach leads to the ill-conditioned Gram expansion matrix. In order to estimate the number of independent basis functions, let us first consider a mesh with one tetrahedron. Formally, there are six basis functions corresponding to six basis edges. Only three of them are linearly independent in $\Re^3$ and should therefore be retained. The number of basis function is

$$N = N_F - N_T \tag{1.4.3}$$

where N_F is the number of faces (four) and N_T is the number of tetrahedra (one) in the mesh.

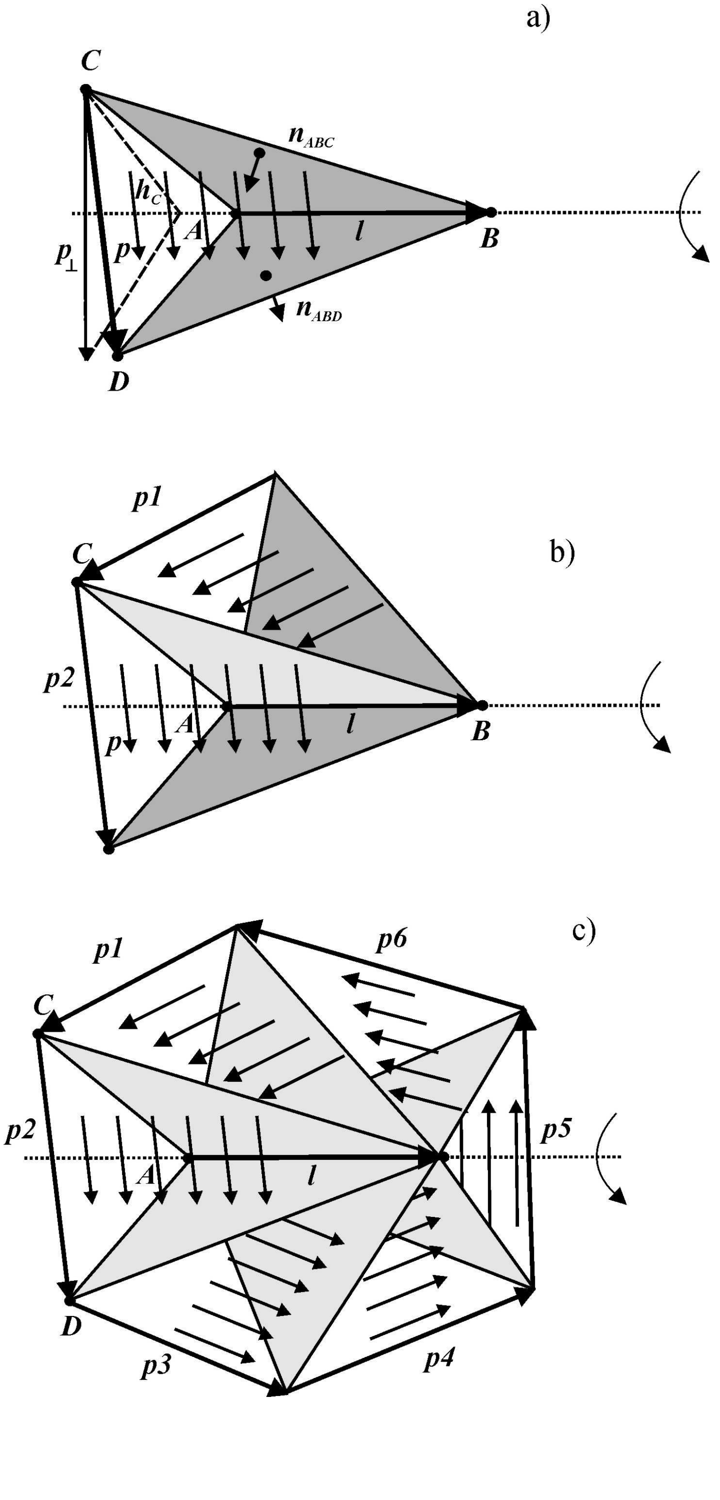


Fig 1.4.1. Three possible combinations of edge based basis functions. a) Two faces on the mesh boundary and no inner face; b) two faces on the mesh boundary and one (or more) inner face; c) only inner faces and no boundary faces.

Next, consider a mesh with two tetrahedra. The component of the electric flux perpendicular to the common face is the same in both tetrahedra and is supported by one basis function. The remaining component of the flux (parallel to the face) is different in both tetrahedra, and is supported by two basis functions in each tetrahedron. The number of independent basis functions (five) is again given by equation (1.4.3) with $N_F = 7, N_T = 2$.

In order to justify equation (1.4.3) in a general case the following can be mentioned. For any tetrahedral mesh, only one basis function is needed per face to support the normal flux component through the given face. This leads to the first term on the right-hand side of equation (1.4.3). On the other hand, any tetrahedron has four faces but needs only three linearly independent basis functions. Therefore, one basis function per tetrahedron must be subtracted. This leads to the second term on the right-hand side of equation (1.4.3). Equation (1.4.3) was validated directly for uniform rectangular meshes. For non uniform meshes the relation is $N < N_F - N_T$. Hence equation (1.4.3) provides a conservative estimate of the number of basis functions. The general expression for the number of basis functions is

$$N \le N_F - N_T \tag{1.4.4}$$

To remove the dependent basis functions for a given tetrahedral mesh, the Gram or "covariance" matrix of a set of the basis functions on the size $N_L \times N_L$ is set in the form

$$G = \begin{bmatrix} \left\langle \vec{f}_1, \vec{f}_1 \right\rangle & \dots & \left\langle \vec{f}_1, \vec{f}_{N_L} \right\rangle \\ \dots & & \dots \\ \left\langle \vec{f}_{N_L}, \vec{f}_1 \right\rangle & & \left\langle \vec{f}_{N_L}, \vec{f}_{N_L} \right\rangle \end{bmatrix}, \quad \left\langle \vec{f}_m, \vec{f}_n \right\rangle = \int_V \vec{f}_m \cdot \vec{f}_n d\upsilon \tag{1.4.5}$$

Matrix *G* is solved by using the LU factorization with partial pivoting using LAPACK'S *DGETRF* [15]. The matrix *G* is filled as a sparse matrix and LU operates on a sparse matrix. The diagonal elements are checked to determine the independent basis functions. The independent columns of matrix *G* correspond to independent basis functions.

Since the number of edges in a large tetrahedral mesh is smaller than the number of faces by typically 30-40%, the matrix *G* is smaller than the impedance matrix for the face-based basis functions. It is also real and symmetric. Therefore, the elimination of the null set requires approximately 25% CPU time compared to the factorization of the complex impedance matrix for the face-based basis functions. The critical point is that the elimination of the null set should be done only *once*. When a frequency sweep is applied, the corresponding CPU time becomes insignificant compared to the total CPU time necessary for the solution of MoM equations for every frequency.

2. Derivation of MoM equations

In this section, the MoM equations are derived for pure metallic, pure dielectric and combined metal-dielectric structure based on the surface RWG and volume edge-based basis functions. The derivations in this section form the core of the solver.

2.1 MoM equation for a pure metallic structure

a. Scattering problem

The total electric field (antenna or scattering problem) is a combination of the incident field (labeled by superscript *i*) and the scattered field (labeled by superscript *s*), i.e.

$$\vec{E} = \vec{E}^i + \vec{E}^s \tag{2.1.1}$$

The incident electric field is either the incoming signal (scattering problem) or the excitation electric field in the antenna feed (radiation problem). The scattered field has a straightforward interpretation for the scattering problem. For the antenna radiation, the "scattered" field is just the field radiated by the antenna feed.

The scattered electric field $\vec{E}^s$ is due to surface currents and free charges on the metal surface *S* (the so-called mixed-potential formulation) [13]

$$\vec{E}^s = -j\omega\vec{A}_S(\vec{r}) - \nabla\Phi_S(\vec{r}); \quad \vec{r} \text{ on } S \tag{2.1.2}$$

Index S denotes the surface related quantities.

The magnetic vector potential $\vec{A}_S(\vec{r})$ describes current radiation whereas the electric potential $\Phi_S(\vec{r})$ describes charge radiation. In the near field, the Φ-contribution is somewhat more critical than the $\vec{A}$-contribution. On the metal surface S, $\vec{E}_{\tan} = 0$, thus giving the electric field integral equations (EFIE),

$$E^i_{\tan} = \left(j\omega\vec{\mathrm{A}}_S + \nabla\Phi_S\right)_{\tan}; \quad \vec{r} \text{ on } S \tag{2.1.3}$$

b. Test functions

Assume that some test functions, $\vec{f}^{\,S}_m(\vec{r})$ $m = 1\ldots N_M$, cover the entire surface S and do not have a component normal to the surface. Multiplication of (2.1.3) by $\vec{f}^{\,S}_m$ and integration over S gives N_M equations

$$\int_S \vec{f}^{\,S}_m \cdot \vec{E}^i ds = j\omega\int_S \vec{f}^{\,S}_m \cdot \vec{A}_S ds - \int_S \left(\nabla \cdot \vec{f}^{\,S}_m\right)\Phi_S ds \tag{2.1.4}$$

since, according to the Stoke's theorem,

$$\int_S \nabla\Phi_S \cdot \vec{f}^{\,S}_m ds = -\int_S \Phi_S \left(\nabla \cdot \vec{f}^{\,S}_m\right) ds \tag{2.1.5}$$

if $\vec{f}^{\,S}_m$ doesn't have a component perpendicular to the surface boundary or edge (if any).

c. Basis functions

The surface current density, $\vec{J}_S$ is expanded into basis functions (which usually coincide with the test functions) in the form

$$\vec{J}_S = \sum_{n=1}^{N_M} I_n \vec{f}_n^S \tag{2.1.6}$$

The magnetic vector potential has the form [1]

$$\vec{\mathrm{A}}_S(\vec{r}) = \frac{\mu_0}{4\pi} \int_S \vec{J}_S g ds' \tag{2.1.7}$$

where μ_0 is the permeability in vacuum and $g = \exp(-jkR)/R, R = |\vec{r} - \vec{r}'|$ is the free-space Green's function (time dependency $\exp(j\omega t)$ is assumed everywhere). In the expression for the Green's function $\vec{r}$ is the observation point and $\vec{r}'$ is the integration point which belongs to the surface.

After substitution of expansion (2.1.6), the above equation becomes

$$\vec{A}_S(\vec{r}) = \sum_{n=1}^{N_M} \left\{ \frac{\mu_0}{4\pi} \int_S \vec{f}_n^S(\vec{r}') g ds' \right\} I_n \tag{2.1.8}$$

Similarly, the electric potential has the form [1]

$$\Phi_S(\vec{r}) = \frac{1}{4\pi\varepsilon_0}\int_S \sigma_S g ds', \quad j\omega\sigma_S = -\nabla \cdot \vec{J}_S \tag{2.1.9}$$

It follows from equation (2.1.9) that σ_s can be expressed in terms of the current density. Hence the electric vector potential finally reduces to

$$\Phi_S(\vec{r}) = \sum_{n=1}^{N_M}\left\{\frac{1}{4\pi\varepsilon_0}\frac{j}{\omega}\int_S \nabla \cdot \vec{f}_n^{\,S}(\vec{r}\,')ds'\right\}I_n \tag{2.1.10}$$

d. Moment equations

The moment equations are obtained if we substitute expansions (2.1.8) and (2.1.10) into the integral equation (2.1.4). In terms of symbolic notations,

$$\sum_{n=1}^{N_M}\hat{Z}_{mn}^{MM} I_n = v_m^M; \quad m = 1,\ldots,N_M \tag{2.1.11}$$

where

$$v_m^M = \int_S \vec{f}_m^{\,S} \cdot \vec{E}^i ds; \quad m = 1,\ldots N_M \tag{2.1.12}$$

are the "voltage" or excitation components for every test/basis function. The integral expressions are the components of the impedance matrix $\hat{Z}^{MM}$ of the size (N_M x N_M).

$$Z_{mn}^{MM} = \left(\frac{j\omega\mu_0}{4\pi}\right)\int_S\int_S \vec{f}_m^{\,S}(\vec{r}) \cdot \vec{f}_n^{\,S}(\vec{r}\,')g ds' ds - \left(\frac{j}{4\pi\omega\varepsilon_0}\right)\int_S\int_S \left(\nabla \cdot \vec{f}_m^{\,S}\right)\left(\nabla \cdot \vec{f}_n^{\,S}\right)g ds' ds \tag{2.1.13}$$

Note that the impedance matrix is symmetric for any set of basis functions (test functions should be the same) when the corresponding surface integrals are calculated precisely. The components of the impedance matrix are the double surface integrals of the Green's function and they mostly reflect the geometry of the problem. In the matrix form, (2.1.11) becomes

$$\hat{Z}^{MM}\vec{I} = \vec{v} \qquad (2.1.14)$$

e. RWG basis functions

Below, we recall the following properties of the RWG basis functions [13]. For any two triangular patches, t_n^+ and t_n^-, having areas A_n^+ and A_n^-, and sharing the common edge l_n, the *n*-th basis function becomes

$$\vec{f}_n^S(\vec{r}) = \begin{cases} \dfrac{l_n}{2A_n^+}\vec{\rho}_n^+ & \vec{r}\ in\ t_n^+ \\ \dfrac{l_n}{2A_n^-}\vec{\rho}_n^- & \vec{r}\ in\ t_n^- \end{cases} \qquad (2.1.15)$$

and

$$\nabla\cdot\vec{f}_n^S(\vec{r}) = \begin{cases} \dfrac{l_n}{A_n^+} & \vec{r}\ in\ t_n^+ \\ -\dfrac{l_n}{A_n^-} & \vec{r}\ in\ t_n^- \end{cases} \qquad (2.1.16)$$

where $\vec{\rho}_n^+ = \vec{r} - \vec{r}_n^+$ is the vector drawn from the free vertex of triangle t_n^+ to the observation point; $\vec{\rho}_n^- = \vec{r}_n^- - \vec{r}$ is the vector drawn from the observation point to the free vertex of triangle t_n^-. The basis function is zero outside two adjacent triangles t_n^+ and t_n^-.

Substitution of equations (2.1.15), (2.1.16) into equation (2.1.13) gives the components of the impedance matrix in terms of surface RWG basis functions in the form

$$\begin{aligned}
&\int_S\int_S \vec{f}_m^S \cdot \vec{f}_n^S g ds' ds = \\
&+\frac{l_m l_n}{4A_m^+ A_n^+}\int_{t_m^+}\int_{t_n^+}\left(\vec{\rho}_m^+ \cdot \vec{\rho}_n'^+\right) g ds' ds + \frac{l_m l_n}{4A_m^+ A_n^-}\int_{t_m^+}\int_{t_n^-}\left(\vec{\rho}_m^+ \cdot \vec{\rho}_n'^-\right) g ds' ds \\
&+\frac{l_m l_n}{4A_m^- A_n^+}\int_{t_m^-}\int_{t_n^+}\left(\vec{\rho}_m^- \cdot \vec{\rho}_n'^+\right) g ds' ds + \frac{l_m l_n}{4A_m^- A_n^-}\int_{t_m^-}\int_{t_n^-}\left(\vec{\rho}_m^- \cdot \vec{\rho}_n'^-\right) g ds' ds
\end{aligned} \qquad (2.1.17)$$

and

$$\begin{aligned}
&\int_S\int_S \left(\nabla \cdot \vec{f}_m^S\right)\left(\nabla \cdot \vec{f}_n^S\right) g ds' ds = \\
&+\frac{l_m l_n}{A_m^+ A_n^+}\int_{t_m^+}\int_{t_n^+} g ds' ds - \frac{l_m l_n}{A_m^+ A_n^-}\int_{t_m^+}\int_{t_n^-} g ds' ds \\
&-\frac{l_m l_n}{A_m^- A_n^+}\int_{t_m^-}\int_{t_n^+} g ds' ds + \frac{l_m l_n}{A_m^- A_n^-}\int_{t_m^-}\int_{t_n^-} g ds' ds
\end{aligned} \qquad (2.1.18)$$

f. Integral calculation

About 90% of the computations required for evaluation of the MoM impedance matrix for RWG basis functions is involved in the calculation of the surface integrals presented in equations (2.1.17), (2.1.18). Consider a structure where all triangular patches are enumerated by $p = 1,...,P$. Then, every integral in equation (2.1.17) is build upon the term

$$\vec{A}_{S\,pq}^{ij} = \int_{t_p}\int_{t_q} (\vec{\rho}_i \cdot \vec{\rho}'_j) g(|\vec{r} - \vec{r}'|) ds' ds \quad p,q = 1,...,P \quad i,j = 1,2,3 \tag{2.1.19}$$

Here, $\vec{\rho}_i = \vec{r} - \vec{r}_i$ for any vertex *i* of patch *p* whereas $\vec{\rho}'_j = \vec{r}' - \vec{r}_j$ for any vertex *j* of patch *q*. Similarly, every integral in equation (2.1.18) is build upon the term

$$\Phi_{S\,pq} = \int_{t_p}\int_{t_q} g(|\vec{r} - \vec{r}'|) ds' ds \quad p,q = 1,...,P \tag{2.1.20}$$

The integrals (2.1.19) and (2.1.20) can be found using a vectorized routine, which employs Gaussian integration of variable order (up to 7th) for both the surface integrals [16, 17]. Calculation is performed over all triangular patches, not over RWG basis functions. The corresponding formulas are given in Appendix B. Prior to the use of the Gaussian integration formulas; a singularity extraction must be done for atleast the self integrals. The singular potential integrals of the type $\int_{t_p} \frac{1}{|\vec{r} - \vec{r}'|}$ are calculated analytically.

g. Self-integrals

The self-integrals ($p = q$ in equations (2.1.19), (2.1.20)) are found precisely, using a number of analytical base integrals presented in [18]. Before doing that, the Taylor expansion is made for the Green's function

$$g = \exp(-jkR)/R \approx 1/R - jk + \mathrm{O}(R) \tag{2.1.21}$$

Therefore, equation (2.1.20) yields

$$\Phi_{Spp} = \int\limits_{t_p}\int\limits_{t_p} g(|\vec{r} - \vec{r}'|)ds'ds \approx \int\limits_{t_p}\int\limits_{t_p} \frac{1}{|\vec{r} - \vec{r}'|} ds'ds - jkA_p^2; \quad p = 1,...,P \tag{2.1.22}$$

where A_p is the area of the triangle t_p. Similarly equation (2.1.19) yields

$$\begin{gathered} \vec{A}^{ij}_{Spp} \approx \int\limits_{t_p}\int\limits_{t_q} \frac{(\vec{r} - \vec{r}_i)\cdot(\vec{r}\,' - \vec{r}_j)}{|\vec{r} - \vec{r}'|} ds'ds - jk\int\limits_{t_p}\int\limits_{t_q} (\vec{r} - \vec{r}_i)\cdot(\vec{r}\,' - \vec{r}_j)ds'ds \\ p = 1,...,P \quad i, j = 1,2,3 \end{gathered} \tag{2.1.23}$$

Introduction of the simplex or barycentric coordinates λ_1, λ_2 for the triangle t_p gives [13,18]

$$\vec{r} = \lambda_1(\vec{r}_1 - \vec{r}_3) + \lambda_2(\vec{r}_2 - \vec{r}_3) + \vec{r}_3, \quad \vec{r}\,' = \lambda_1'(\vec{r}_1 - \vec{r}_3) + \lambda_2'(\vec{r}_2 - \vec{r}_3) + \vec{r}_3 \tag{2.1.24}$$

and

$$\begin{aligned} &(\vec{r} - \vec{r}_i)\cdot(\vec{r}\,' - \vec{r}_j) = \\ &\qquad (\vec{r}_3 - \vec{r}_i)\cdot(\vec{r}_3 - \vec{r}_j)\times 1 + \\ &\qquad (\vec{r}_1 - \vec{r}_3)\cdot(2\vec{r}_3 - \vec{r}_i - \vec{r}_j)\times\lambda_1 + \\ &\qquad (\vec{r}_2 - \vec{r}_3)\cdot(2\vec{r}_3 - \vec{r}_i - \vec{r}_j)\times\lambda_2 + \\ &\qquad (\vec{r}_1 - \vec{r}_3)\cdot(\vec{r}_1 - \vec{r}_3)\times\lambda_1\lambda_1' + \\ &\qquad (\vec{r}_2 - \vec{r}_3)\cdot(\vec{r}_2 - \vec{r}_3)\times\lambda_2\lambda_2' + \\ &\qquad (\vec{r}_1 - \vec{r}_3)\cdot(\vec{r}_2 - \vec{r}_3)\times\lambda_1\lambda_2' + \\ &\qquad (\vec{r}_1 - \vec{r}_3)\cdot(\vec{r}_2 - \vec{r}_3)\times\lambda_1'\lambda_2 \end{aligned} \tag{2.1.25}$$

Substitution of equation (2.1.25) into the first term on the right-hand side of equation (2.1.23) results in seven integrals. Each of those is reduced to one of the four independent base integrals given in [18]. The remaining values are obtained using cyclic transformation. Integral (2.1.22) only needs the first base integral [5]. The second term on the right-hand side of equation (2.1.23) is calculated in the straightforward manner. Further details are given in Appendix B.

2.2 MoM equations for a pure dielectric structure

In this section, the MoM equations for a pure dielectric object (a scatterer) are derived from the EFIE [1], utilizing the volume edge-based basis functions [14].

a. Scattering problem

The total electric field (antenna or scattering problem) in a dielectric volume is a combination of the incident field (labeled by superscript *i*) and the scattered field (labeled by superscript *s*), i.e.

$$\vec{E} = \vec{E}^i + \vec{E}^s \tag{2.2.1}$$

Let V denote the volume of a lossy, inhomogeneous, dielectric body with (complex) dielectric constant $\hat{\varepsilon}(\vec{r}) = \varepsilon(\vec{r}) - j\sigma(\vec{r})/\omega$, where ε and σ are the medium electric permittivity and conductivity when $\vec{r}$ is in$V$. The total electric field in that case can be expressed in terms of the electric flux density, $\vec{D}(\vec{r})$, as

$$\vec{E} = \vec{D}(\vec{r})/\hat{\varepsilon}(\vec{r})$$

The incident field is the incoming signal for the scattering problem. The scattered electric field $\vec{E}^s$ is found using the volume equivalence principle [1]. Namely, the dielectric material is removed and replaced by equivalent volume polarization currents. The scattered field is due to the volume polarization currents in dielectric volume V (bounded by surface Ω)

$$\vec{E}^s = -j\omega\vec{A}(\vec{r}) - \nabla\Phi(\vec{r}); \quad \vec{r} \text{ in } V \tag{2.2.2}$$

The magnetic vector potential $\vec{A}(\vec{r})$ describes radiation of volume polarization currents, whereas the electric potential $\Phi(\vec{r})$ describes radiation of the associated bound charges. Thus, from the expressions for $\vec{E}$ and $\vec{E}^s$, we can write the EFIE as

$$\vec{E}^i = \frac{\vec{D}(\vec{r})}{\hat{\varepsilon}(\vec{r})} + j\omega\vec{A}(\vec{r}) + \nabla\Phi(\vec{r}); \quad \vec{r} \text{ in } V \tag{2.2.3}$$

b. Test functions

Assume that some volume test functions, $K(\vec{r})\vec{f}_m(\vec{r})$ m = 1… N_D, cover the entire dielectric volume V where $K(\vec{r})$ is an arbitrary factor. Each function $\vec{f}_m(\vec{r})$ is assumed to be piecewise continuous, divergence free, and has a continuous normal component everywhere within a smaller definition volume V_n, including its boundary. Multiplication of equation (2.2.3) by $K(\vec{r})\vec{f}_m(\vec{r})$ and integration over volume V gives N_D equations

$$\int_V K(\vec{r})\vec{f}_m(\vec{r})\cdot\vec{E}^i dv = \left[\int_V K(\vec{r})\vec{f}_m(\vec{r})\frac{\vec{D}(\vec{r})}{\hat{\varepsilon}(\vec{r})}dv + j\omega\int_V K(\vec{r})\vec{f}_m(\vec{r})\cdot\vec{A}dv + \int_V K(\vec{r})\vec{f}_m(\vec{r})\cdot\nabla\Phi(\vec{r})dv\right] \tag{2.2.4}$$

Transforming the last volume integral by using Stokes theorem for every individual tetrahedron T_p we obtain

$$\int_{V_p} K(\vec{r})\vec{f}_m(\vec{r})\cdot\nabla\Phi dv = -\int_{V_p} K(\vec{r})\Phi\left(\nabla\cdot\vec{f}_m(\vec{r})\right)dv + \int_{\Omega_p} K(\vec{r})\Phi\left(\hat{n}(\vec{r})\cdot\vec{f}_m(\vec{r})\right)ds$$

where Ω_p is the boundary surface for tetrahedra T_p. However the volume basis functions are divergenceless i.e. $\nabla \cdot \vec{f}_m(\vec{r}) = 0$. Hence

$$\int_{V_p} K(\vec{r})\vec{f}_m(\vec{r}) \cdot \nabla\Phi dv = \int_{\Omega_p} K(\vec{r})\Phi \left(\hat{n}(\vec{r}) \cdot \vec{f}_m(\vec{r})\right) ds = \int_{\Omega_p} K(\vec{r})\Phi \cdot f_{\perp m}(\vec{r}) ds \qquad (2.2.5)$$

where $\hat{n}$ is the unit outer normal to the surface Ω_p and $f_{\perp m}(\vec{r})$ is the outer normal component of the basis function $\vec{f}_m(\vec{r})$ on surface Ω_p .

Substituting the equation (2.2.5) in (2.2.4) gives the expression of the form

$$\int_V K(\vec{r})\vec{f}_m(\vec{r}) \cdot \vec{E}^i dv = \left[\int_V K(\vec{r})\vec{f}_m(\vec{r}) \frac{\vec{D}(\vec{r})}{\hat{\varepsilon}(\vec{r})} dv + j\omega \int_V K(\vec{r})\vec{f}_m(\vec{r}) \cdot \vec{A} dv + \sum_{p=1}^{P} \int_{\Omega} K(\vec{r})\Phi \cdot \vec{f}_{\perp m}(\vec{r}) ds \right] \qquad (2.2.6)$$

where P is the total number of tetrahedra contained by the basis functions.

c. Differential contrast

We need to transform the surface integral in (2.2.6) by considering each face of the tetrahedron separately. To achieve this we consider three representative cases depicted in Fig. 2.2.1. In the first case (Fig. 2.2.1a); S_1 is the common face between the dielectric tetrahedron ABCD and the fictitious tetrahedron in vacuum. In the second case two tetrahedra ABCD and ABCE share a common face S_2 while S_3 is the common face between tetrahedron ABCE and a fictitious tetrahedron in vacuum as shown in Fig 2.2.1c. We compute the surface integral in equation (2.2.6) by considering each of the above faces separately. The dielectric contrast $K(r) = \dfrac{\hat{\varepsilon}(\vec{r}) - \varepsilon_o}{\hat{\varepsilon}(\vec{r})}$ is a constant within every tetrahedron. The surface normal is directed from plus (or left) to minus (right) tetrahedron.

The face normal is aligned according to the right-hand rule for the base edge. Its direction follows the direction of the vector field. The differential contrast is obtained combining the contrast data for the left and right tetrahedron. If one of them does not exist (vacuum-dielectric boundary), then either K_+ or K_- becomes zero. For a homogeneous dielectric this guarantees that (i) every basis function with only inner faces doesn't create any surface charge; and (ii) every basis function with two boundary faces possesses the net surface charge zero.

Using the above theory we reduce the surface integral of (2.2.6) for each common face for the two tetrahedra ABCD and ABCE as shown in Fig 2.1.1.

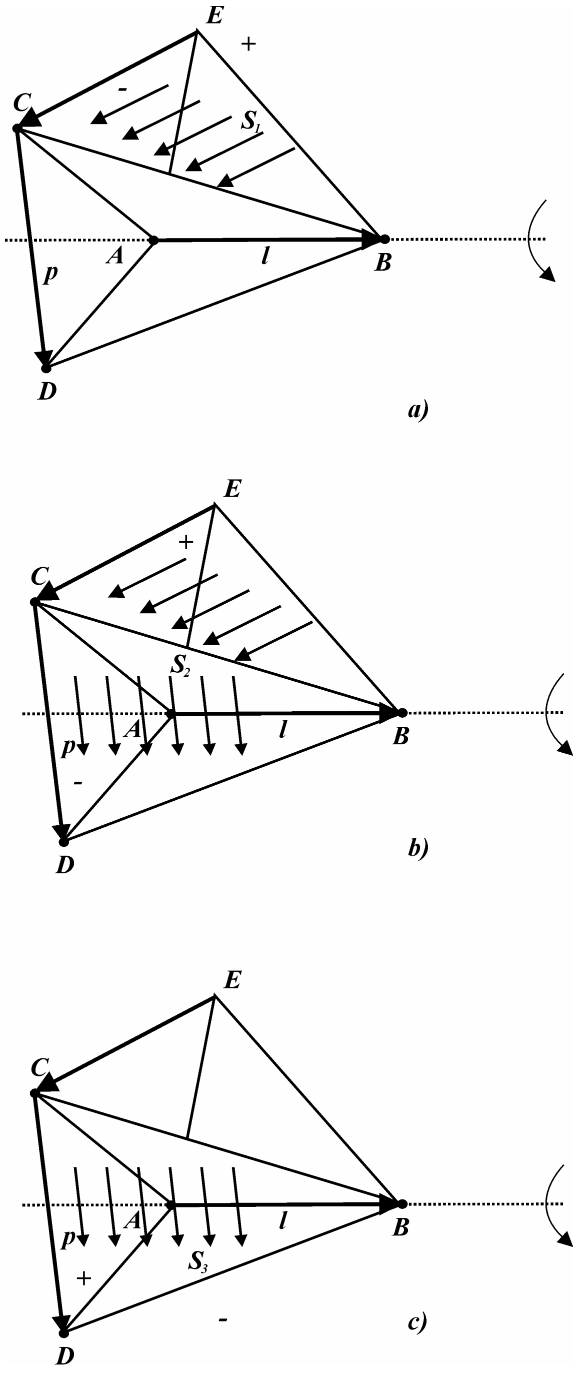


Fig. 2.2.1 Three representative cases to transform the surface integral by considering each face of the tetrahedron separately. a) S_1 is the common face between the dielectric tetrahedron ABCD and the fictitious tetrahedron in vacuum; b) S_2 is the common face between the dielectric tetrahedron ABCD and ABCE; c) S_3 is the common face between the dielectric tetrahedron ABCE and the fictitious tetrahedron in vacuum.

Face S_1:

$$\int_S K(\vec{r})\Phi \cdot f_{\perp mq}(\vec{r})ds = 0 - \int_{S_1} K_- \Phi \cdot f_{\perp mq}(\vec{r})ds = \int_{S_1} (0 - K_-)\Phi \cdot f_{\perp mq}(\vec{r})ds \tag{2.2.7}$$

Face S_2:

$$\int_S K(\vec{r})\Phi \cdot f_{\perp mq}(\vec{r})ds = \int_{S_2} K_+ \Phi \cdot f_{\perp mq}(\vec{r})ds - \int_{S_2} K_- \Phi \cdot f_{\perp mq}(\vec{r})ds = \int_{S_2} (K_+ - K_-)\Phi \cdot f_{\perp mq}(\vec{r})ds \tag{2.2.8}$$

Face S_3:

$$\int_S K(\vec{r})\Phi \cdot f_{\perp mq}(\vec{r})ds = \int_{S_3} K_+ \Phi \cdot f_{\perp mq}(\vec{r})ds - 0 = \int_{S_3} (K_+ - 0)\Phi \cdot f_{\perp mq}(\vec{r})ds \tag{2.2.9}$$

Summing up the above integrals we get for a tetrahedra pair the surface integral as

$$\int_{S_1} (0 - K_-)\Phi \cdot f_{\perp mq}(\vec{r})ds + \int_{S_2} (K_+ - K_-)\Phi \cdot f_{\perp mq}(\vec{r})ds + \int_{S_3} (K_+ - 0)\Phi \cdot f_{\perp mq}(\vec{r})ds$$

$$= \int_{S_1} \hat{K}_1 \Phi \cdot f_{\perp mq}(\vec{r})ds + \int_{S_2} \hat{K}_2 \Phi \cdot f_{\perp mq}(\vec{r})ds + \int_{S_3} \hat{K}_3 \Phi \cdot f_{\perp mq}(\vec{r})ds \tag{2.2.10}$$

Hence from equations (2.2.7)-(2.2.10)

$$\int_\Omega K(\vec{r})\Phi \cdot f_{\perp mq}(\vec{r})ds = \int_\Omega \hat{K}_q \Phi \cdot f_{\perp mq}(\vec{r})ds \tag{2.2.11}$$

where $\hat{K}_q = K_+ - K_-$ is the differential contrast on face *q*.

Hence equation (2.2.6) finally reduces to

$$\int_V K(\vec{r})\vec{f}_m(\vec{r})\cdot\vec{E}^i dv = \left[\int_V K(\vec{r})\vec{f}_m(\vec{r})\frac{\vec{D}(\vec{r})}{\hat{\varepsilon}(\vec{r})}dv + j\omega\int_V K(\vec{r})\vec{f}_m(\vec{r})\cdot\vec{A}dv + \int_\Omega \hat{K}_q\,\Phi\cdot f_{\perp mq}(\vec{r})ds\right] \quad (2.2.12)$$

d. Field and charge expansion

The total electric flux, $\vec{D}(\vec{r})$, is expanded in the form

$$\vec{D}(\vec{r}) = \sum_{n=1}^{N} D_n\vec{f}_n(\vec{r}) \qquad (2.2.13)$$

where $\vec{f}_n(\vec{r})$ is a set of volumetric basis functions.

Once equation (2.2.13) is given, the density of the surface bound charges is established following the continuity equation, in terms of the surface δ-functions [1]. The equivalent result can be obtained using Gauss theorem or the boundary condition on the dielectric interface. Consider two arbitrary tetrahedra (plus and minus) that share a common face (which includes the base edge) but have different dielectric constants $\hat{\varepsilon}^{\pm}$. The surface charge density, σ_Ω, from Gauss law,

$$\sigma_\Omega \equiv (K_+ - K_-)D_\perp = \hat{K}D_\perp \qquad (2.2.14)$$

where $D_\perp$ is the normal component of the total electric displacement on the boundary; K_+, K_- are the corresponding dielectric contrasts.

For every basis function $\vec{f}_n(\vec{r})$, and for every face, which supports the normal component of this basis function, the associated surface charge is enforced to follow equation (2.2.14). The normal component of $\vec{f}_n(\vec{r})$ is available from equation (1.4.1). The total surface charge density in the dielectric is obtained as a combination of the contributions of all basis functions

$$\sigma_\Omega(\vec{r}) \equiv \hat{K}(\vec{r})D_\perp(\vec{r}) = \sum_{n=1}^{N}\left\{\sum_{q=1}^{Q}\hat{K}_q f_{\perp nq}(\vec{r})\right\}D_n; \quad \vec{r} \in \Omega \tag{2.2.15}$$

where $\hat{K}_q = K_+ - K_-$ is the differential contrast on face q, $f_{\perp nq}(\vec{r})$ is the normal component of the basis function $\vec{f}_n(\vec{r})$ on face q. The inner summation in equation (2.2.15) is done over all Q faces, which support the normal component of the n-th basis function.

Along with (2.2.15), the volume polarization current density [1] in the dielectric volume, except for any of its boundaries, is given by

$$\vec{J}_V(\vec{r}) \equiv j\omega K(\vec{r})\vec{D}(\vec{r}) = j\omega\sum_{n=1}^{N}\left\{\sum_{p=1}^{P}K_p\vec{f}_{np}(\vec{r})\right\}D_n; \quad \vec{r} \in V, \vec{r} \notin \Omega \tag{2.2.16}$$

The inner summation in equation (2.2.16) is done over all P tetrahedra, which are contained by the n-th basis function. Every tetrahedron may possess its own dielectric contrast K_p.

e. MoM equations

According to the volume equivalence principle [1], the piecewise inhomogeneous dielectric material is removed and replaced by equivalent volume polarization currents in V and by the associated surface bound charges on Ω. The volume EFIE is written in the mixed-potential form [13], i.e. in form of equation (2.2.3)

$$\vec{E}^i = \vec{E} + j\omega\vec{A}(\vec{r}) + \nabla\Phi(\vec{r}); \quad \vec{r} \in V, \vec{r} \in \Omega \tag{2.2.17}$$

where $\vec{E} = \vec{D} / \hat{\varepsilon}$ is the net electric field and $\vec{E}^i$ is the incident field. The magnetic vector potential $\vec{A}(\vec{r})$ describes radiation of volume polarization currents given by equation (2.2.16), whereas the electric potential $\Phi(\vec{r})$ describes radiation of the associated bound charges given by equation (2.2.15). One has

$$\vec{A}(\vec{r}) = \frac{\mu_0}{4\pi}\int_V \vec{J}_V(\vec{r}\,')g(\vec{r},\vec{r}\,')dv'\,,\ \Phi(\vec{r}) = \frac{1}{4\pi\varepsilon_0}\int_\Omega \sigma_\Omega(\vec{r}\,')g(\vec{r},\vec{r}\,')ds' \tag{2.2.18}$$

where $g(\vec{r},\vec{r}\,') = \exp(-jkR)/R$, $R = |\vec{r} - \vec{r}\,'|$ is the free-space Green's function.

Substituting the values of the surface charge density and volume polarization currents form equations (2.2.15) and (2.2.16) in equation (2.2.18) yields

$$\vec{A}(\vec{r}) = \frac{j\omega\mu_0}{4\pi}\int_V \sum_{n=1}^{N}\left\{\sum_{p=1}^{P} K_{p'}\vec{f}_{np'}(\vec{r}\,')g(\vec{r},\vec{r}\,')dv'\right\}D_n \tag{2.2.19}$$

$$\Phi(\vec{r}) = \frac{1}{4\pi\varepsilon_0}\int_\Omega \sum_{n=1}^{N}\left\{\sum_{q=1}^{Q} \hat{K}_{q'}f_{\perp nq'}(\vec{r}\,')g(\vec{r},\vec{r}\,')ds'\right\}D_n \tag{2.2.20}$$

After substitution of (2.2.13), (2.2.19), and (2.2.20), equation (2.2.12) gives the MoM equations in the form

$$\sum_{n=1}^{N} \hat{Z}_{mn} D_n = \upsilon_m^D; \quad m = 1,..., N_D \tag{2.2.21}$$

where

$$\upsilon_m^D = \sum_{p=1}^{P} K_p \int_{V_p} \vec{f}_{mp}(\vec{r}) \cdot \vec{E}^i(\vec{r}) dv; \quad m = 1,...N_D$$

are the “voltage” or excitation components for every test/basis functions.

The impedance matrix $\hat{Z}$ is given by

$$\begin{aligned} Z_{mn} &= \sum_{p=1}^{P} \sum_{p'=1}^{P'} \frac{K_p}{\hat{\varepsilon}_p} \int_{V_p} \vec{f}_{mp}(\vec{r}) \cdot \vec{f}_{np'}(\vec{r}) dv \\ &- \frac{\omega^2 \mu_0}{4\pi} \sum_{p=1}^{P} \sum_{p'=1}^{P'} K_p K_{p'} \int_{V_p} \int_{V_{p'}} g(\vec{r}, \vec{r}') \vec{f}_{mp}(\vec{r}) \cdot \vec{f}_{np'}(\vec{r}') dv\, dv' \\ &+ \frac{1}{4\pi\varepsilon_0} \sum_{q=1}^{Q} \sum_{q'=1}^{Q'} \hat{K}_q \hat{K}_{q'} \int_{\Omega_q} \int_{\Omega_{q'}} g(\vec{r}, \vec{r}') f_{\perp mq}(\vec{r}) f_{\perp nq'}(\vec{r}') ds\, ds' \qquad m, n = 1,..., N \end{aligned} \tag{2.2.22}$$

The symmetric impedance matrix is thus written as a combination of individual volume and surface integrals. Since both the basis/test functions and their normal components are constant for a given tetrahedron/face, equation (2.2.22) may be notably simplified.

$$
\begin{aligned}
Z_{mn} &= \sum_{p=1}^{P}\sum_{p'=1}^{P'}\frac{K_p V_p}{\hat{\varepsilon}_p l_m l_n \vec{\Omega}_{mp}\vec{\Omega}_{np'}}\left(\vec{p}_{mp}\cdot\vec{p}_{np'}\right)\delta_{pp'} \\
&-\frac{\omega^2\mu_0}{4\pi}\sum_{p=1}^{P}\sum_{p'=1}^{P'}\frac{K_p K_{p'}}{l_m l_n \vec{\Omega}_{mp}\vec{\Omega}_{np'}}\left(\vec{p}_{mp}\cdot\vec{p}_{np'}\right)\int_{V_p}\int_{V_{p'}} g(\vec{r},\vec{r}\,')dv\,dv' \qquad (2.2.23) \\
&+\frac{1}{4\pi\varepsilon_0}\sum_{q=1}^{Q}\sum_{q'=1}^{Q'}\frac{\hat{K}_q\hat{K}_{q'}}{\Omega_q\Omega_{q'}}\int_{S_q}\int_{S_{q'}} g(\vec{r},\vec{r}\,')ds\,ds' \qquad m,n=1,\dots,N
\end{aligned}
$$

Note that the first term on the right-hand side of equation (2.2.23) is only different from zero when the p-th tetrahedron of basis function m coincides with the p'-th tetrahedron of basis function n.

f. Integral calculation

About 90% of the computations required for evaluation of the MoM impedance matrix for edge-based basis functions is involved in the calculation of the volume/volume and surface/surface integrals presented in equation (2.2.23). Consider a structure where all tetrahedral volumes are enumerated by $p=1,\dots,P$. Then, integral in equation (2.2.23) is built upon the term

$$
\vec{A}^{ij}_{Vpq} = \int_{V_p}\int_{V_q} g(|\vec{r}-\vec{r}\,'|)dv'dv \qquad p,q=1,\dots,P \quad i,j=1,2,3,4 \qquad (2.2.24)
$$

Also, integral in equation (2.2.23) is built upon the term

$$
\Phi^{ij}_{Spq} = \int_{\Omega_p}\int_{\Omega_q} g(|\vec{r}-\vec{r}\,'|)ds'ds \qquad p,q=1,\dots,P \qquad (2.2.25)
$$

The integrals (2.2.24)-(2.2.25) are found using a separate routine, which employs Gaussian integration of variable order [16, 17]. The singular portions of the surface to surface self integrals are found analytically, using the closed-form computation given in [18]. The singularity extraction is also done for volume self-integrals. The singular portions of the volume-to-volume self integrals are first integrated analytically using Gauss theorem [19]. The remaining integrals are non-singular and are evaluated numerically, using typically seven- point Gaussian quadrature for faces and the eight-point barycentric quadrature for volumes. Also, the Gaussian formulas for tetrahedra are used [20, 21, 22].Such an approach gives a reasonable compromise between accuracy and speed. The corresponding formulas are given in Appendix B.

2.3 MoM equations for combined metal-dielectric structure

In this section, the MoM equations for a combined metal-dielectric object (a scatterer) are accurately derived for the electric field integral equation (EFIE) [1], utilizing surface RWG basis functions [13] and the edge-based basis functions [14] following the approach as given in [4].

a. Scattering problem

The total electric field (scattering problem) is a combination of the incident field (labeled by superscript *i*) and the scattered field (labeled by superscript *s*), i.e.

$$\vec{E} = \vec{E}^i + \vec{E}^s \tag{2.3.1}$$

Let V (bounded by surface Ω) denote the volume of a lossy, inhomogeneous, dielectric body with (complex) dielectric constant $\hat{\varepsilon}(\vec{r}) = \varepsilon(\vec{r}) - j\sigma(\vec{r})/\omega$, where ε and σ are the medium permittivity and conductivity when $\vec{r}$ is in V. Let a metal surface *S* be attached to this dielectric object.

The incident field is the incoming signal for the scattering problem. The scattered electric field $\vec{E}^s$ in this case will have two components. One is due to volume polarization currents in the dielectric volume V and associated bound charges on the boundaries of an inhomogeneous dielectric region, and the other component is due to surface conduction currents and free charges on the metal surface *S*. Using the expressions for the scattered field in terms of the electric and magnetic potentials $\vec{A}$ and Φ , one has

$$\vec{E}^s = -j\omega\vec{A}(\vec{r}) - \nabla\Phi(\vec{r}) - j\omega\vec{A}_S(\vec{r}) - \nabla\Phi_S(\vec{r}); \quad \vec{r} \text{ in } V \tag{2.3.2}$$

$$\vec{E}^s = -j\omega\vec{A}_S(\vec{r}) - \nabla\Phi_S(\vec{r}) - j\omega\vec{A}(\vec{r}) - \nabla\Phi(\vec{r}); \quad \vec{r} \text{ on } S \tag{2.3.3}$$

where index S refers to metal surface. The magnetic vector potential $\vec{A}(\vec{r})$ and electric potential $\Phi(\vec{r})$ carry their usual meanings corresponding to metal and dielectric. Since

$$\vec{D} = \hat{\varepsilon}\vec{E} \quad \text{in the dielectric volume } V \tag{2.3.4}$$

$$\vec{E}_{\tan} = 0 \quad \text{on the metal surface } S \tag{2.3.5}$$

using the expressions for $\vec{E}$ and $\vec{E}^s$, we can write the EFIE as (see section 2.1 and 2.2)

$$\vec{E}^i = \frac{\vec{D}(\vec{r})}{\hat{\varepsilon}(\vec{r})} + j\omega\vec{A}(\vec{r}) + \nabla\Phi(\vec{r}) + j\omega\vec{A}_S(\vec{r}) + \nabla\Phi_S(\vec{r}); \quad \vec{r} \text{ in } V \tag{2.3.6}$$

$$\vec{E}^i_{\tan} = \left[+ j\omega\vec{A}_S(\vec{r}) + \nabla\Phi_S(\vec{r}) + j\omega\vec{A}(\vec{r}) + \nabla\Phi(\vec{r})\right]_{\tan}; \quad \vec{r} \text{ on } S \tag{2.3.7}$$

b. Test functions

Assume that some test functions, $K(\vec{r})\vec{f}_m(\vec{r})$, m = 1… N_D, cover the entire dielectric volume V. Multiplication of equation (2.3.6) by $K(\vec{r})\vec{f}_m(\vec{r})$ and integration over volume V gives N_D equations

$$\int_V K(\vec{r})\vec{f}_m(\vec{r})\cdot\vec{E}^i d\vec{r} = \left[\begin{array}{l} \int_V K(\vec{r})\vec{f}_m(\vec{r})\cdot\frac{\vec{D}(\vec{r})}{\hat{\varepsilon}(\vec{r})}dv + j\omega\int_V K(\vec{r})\vec{f}_m(\vec{r})\cdot\vec{A}dv + \int_V K(\vec{r})\vec{f}_m(\vec{r})\cdot\nabla\Phi(\vec{r})dv \\ + j\omega\int_V K(\vec{r})\vec{f}_m(\vec{r})\cdot\vec{A}_S dv + \int_V K(\vec{r})\vec{f}_m(\vec{r})\cdot\nabla\Phi_S(\vec{r})dv \end{array}\right] \tag{2.3.8}$$

Simplifying the last volume integral by applying Stokes theorem for every individual tetrahedron in the manner similar to the simplification of the last volume integral in Z^{DD} given in equation (2.2.5) yields

$$\int_V K(\vec{r})\vec{f}_m(\vec{r})\cdot\nabla\Phi_S dv = -\int_V K(\vec{r})\Phi_S\left(\nabla\cdot\vec{f}_m(\vec{r})\right)dv + \int_\Omega K(\vec{r})\Phi_S\left(\hat{n}(\vec{r})\cdot\vec{f}_m(\vec{r})\right)ds$$

However the volume basis functions are divergenceless. Hence

$$\int_V K(\vec{r})\vec{f}_m(\vec{r})\cdot\nabla\Phi dv = \int_\Omega K(\vec{r})\Phi_S\left(\hat{n}(\vec{r})\cdot\vec{f}_m(\vec{r})\right)ds = \int_\Omega K(\vec{r})\Phi_S\cdot f_{\perp mq}(\vec{r})ds \tag{2.3.9}$$

where $\hat{n}$ is the unit outer normal to the surface Ω and $f_{\perp mq}(\vec{r})$ is the outer normal component of the basis function $\vec{f}_m(\vec{r})$ on face *q*. Substituting the values from equations (2.2.5) and (2.3.9) in equation (2.3.8) gives

$$\int_V K(\vec{r})\vec{f}_m(\vec{r})\cdot\vec{E}^i dv = \left[\begin{array}{l} \underbrace{\int_V K(\vec{r})\vec{f}_m(\vec{r})\cdot\frac{\vec{D}(\vec{r})}{\hat{\varepsilon}(\vec{r})}dv + j\omega\int_V K(\vec{r})\vec{f}_m(\vec{r})\cdot\vec{A}dv + \int_\Omega \hat{K}_q\,\Phi\; f_{\perp mq}(\vec{r})ds}_{Z^{DD}} \\ \qquad\qquad \underbrace{+ j\omega\int_V K(\vec{r})\vec{f}_m(\vec{r})\cdot\vec{A}_S dv + \int_\Omega \hat{K}_q\,\Phi_S\; f_{\perp mq}(\vec{r})ds}_{Z^{DM}} \end{array}\right] \tag{2.3.10}$$

The process of converting the contrast, $K(\vec{r})$, to the differential contrast, $\hat{K}$, is exactly the same as explained in section 2.2c. The term on the right-hand side of equation (2.3.10), labeled Z^{DD}, is exactly the right-hand side of equation (2.2.12) from section 2.2 for pure dielectric. The term, labeled Z^{DM}, describes the contribution of radiation from the metal surface to the dielectric volume.

Now assume that the surface test functions, $\vec{f}_m^S(\vec{r})$, m = 1… N_M, cover the entire metal surface S and do not have a component normal to the surface. Multiplication of equation (2.3.7) by $\vec{f}_m^S(\vec{r})$ and integration over surface S gives N_M equations

$$\int_S \vec{f}_m^S(\vec{r})\cdot\vec{E}^i ds = \begin{bmatrix} \underbrace{+j\omega\int_S \vec{f}_m^S(\vec{r})\cdot\vec{A}_S ds - \int_S \left(\nabla\cdot\vec{f}_m^S(\vec{r})\right)\Phi_S ds}_{Z^{MM}} \\ \\ \underbrace{+j\omega\int_S \vec{f}_m^S(\vec{r})\cdot\vec{A} ds - \int_S \left(\nabla\cdot\vec{f}_m^S(\vec{r})\right)\Phi ds}_{Z^{MD}} \end{bmatrix} \tag{2.3.11}$$

since according to Stoke’s theorem,

$$\int_S \vec{f}_m^S(\vec{r})\cdot\nabla\Phi ds = -\int_S \Phi\left(\nabla\cdot\vec{f}_m^S(\vec{r})\right)ds \tag{2.2.12}$$

The term on the right-hand side of equation (2.2.11), labeled Z^{MM}, is exactly the right-hand side of equation (2.1.14) from section 2.1 for pure metal. The term, labeled Z^{MD}, describes the contribution of radiation from the dielectric volume to the metal surface.

c. Basis functions

The material in this section is essentially the combination of sections 2.1 and 2.2 where the MoM equations are obtained for the pure metal and pure dielectric structure. Given N_D independent edge-based basis functions, the total electric flux $\vec{D}(\vec{r})$ in the dielectric and the associated bound surface charge density σ_S, from Gauss law, are expanded in the form (see section 2.2)

$$\vec{D}(\vec{r}) = \sum_{n=1}^{N_D} D_n \vec{f}_n(\vec{r}), \qquad \sigma_\Omega(\vec{r}) \equiv \hat{K}(\vec{r}) D_\perp(\vec{r}) = \sum_{n=1}^{N_D} \left\{ \sum_{q=1}^{Q} \hat{K}_q f_{\perp nq}(\vec{r}) \right\} D_n; \quad \vec{r} \in \Omega \quad (2.3.13)$$

where $D_\perp$ is the normal component of the total electric flux on the boundary; $\hat{K}_q = K_+ - K_-$ is the differential contrast on face q between tetrahedra "+" and "-"; and $K_\pm = (\hat{\varepsilon}_\pm - \varepsilon_0) / \hat{\varepsilon}_\pm$ is the dielectric contrast, which is a constant within every tetrahedron. The associated surface normal is directed from plus (or left) to minus (right) tetrahedron. The surface normal is aligned according to the right-hand rule for the base edge. The differential contrast is obtained combining the contrast data for the left and right tetrahedron. If one of them does not exist (vacuum-dielectric boundary), then either K_+ or K_- becomes zero. For a homogeneous dielectric this guarantees that every basis function with two boundary faces possesses the net surface charge zero.

The volume polarization current density in the dielectric volume, except for any of its boundaries, is given by

$$\vec{J}_V(\vec{r}) \equiv j\omega K(\vec{r})\vec{D}(\vec{r}) = j\omega\sum_{n=1}^{N}\left\{\sum_{p=1}^{P} K_p \vec{f}_{np}(\vec{r})\right\}D_n; \quad \vec{r} \in V, \vec{r} \notin \Omega \tag{2.3.14}$$

The magnetic vector potential $\vec{A}(\vec{r})$ describes radiation of volume polarization currents given by equation (2.3.14), whereas the electric potential $\Phi(\vec{r})$ describes radiation of the associated bound charges given by equation (2.3.13). One has

$$\vec{A}(\vec{r}) = \frac{\mu_0}{4\pi}\int_V \vec{J}_V(\vec{r}')g(\vec{r},\vec{r}')dv' \qquad \Phi(\vec{r}) = \frac{1}{4\pi\varepsilon_0}\int_\Omega \sigma_\Omega(\vec{r}')g(\vec{r},\vec{r}')ds' \tag{2.3.15}$$

where $g(\vec{r},\vec{r}') = \exp(-jkR)/R$, $R = \left|\vec{r} - \vec{r}'\right|$ is the free-space Green's function.

Substituting the values of $\vec{J}_V(\vec{r})$ and $\sigma_\Omega(\vec{r})$ from equations (2.3.13) and (2.3.14) in (2.3.15) results in

$$\vec{A}(\vec{r}) = \frac{j\omega\mu_0}{4\pi}\int_V \sum_{n=1}^{N}\left\{\sum_{p=1}^{P} K_p \vec{f}_{np}(\vec{r})g(\vec{r},\vec{r}')\right\}D_n dv' \tag{2.3.16}$$

$$\Phi(\vec{r}) = \frac{1}{4\pi\varepsilon_0}\int_\Omega \sum_{n=1}^{N_D}\left\{\sum_{q=1}^{Q} \hat{K}_q f_{\perp nq}(\vec{r})g(\vec{r},\vec{r}')\right\}D_n ds' \tag{2.3.17}$$

Given N_M independent rooftop basis functions [13], the metal surface current density $\vec{J}(\vec{r})$ is expanded in the form (see section 2.1)

$$\vec{J}(\vec{r}) = \sum_{n=1}^{N_M} I_n \vec{f}_n^{\,S}(\vec{r}) \tag{2.3.18}$$

where $\vec{f}_n^S(\vec{r})$ are the RWG rooftop basis functions [13].

The magnetic vector potential [1] after substitution of expansion (2.3.18) becomes

$$\vec{A}_S(\vec{r}) = \sum_{n=1}^{N_M} \left\{ \frac{\mu_0}{4\pi} \int_S \vec{f}_n^S(\vec{r}\,') g(\vec{r},\vec{r}\,') ds' \right\} I_n \tag{2.3.19}$$

Similarly, the electric potential takes the form,

$$\Phi_S(\vec{r}) = \frac{1}{4\pi\varepsilon_0} \sum_{n=1}^{N_M} \left\{ \frac{j}{\omega} \int_S \left(\nabla \cdot \vec{f}_n^S(\vec{r}\,') \right) g(\vec{r},\vec{r}\,') ds' \right\} I_n \tag{2.3.20}$$

d. Moment equations

The moment equations are obtained if we substitute expansions (2.3.16), (2.3.17) and (2.3.19), (2.3.20), into equations (2.3.10), (2.3.11). In terms of symbolic notations,

$$\sum_{n=1}^{N_D} \hat{Z}_{mn}^{DD} D_n + \sum_{n=1}^{N_M} \hat{Z}_{mn}^{DM} I_n = \upsilon_m^D; \quad m = 1,...N_D \tag{2.3.21}$$

$$\sum_{n=1}^{N_M} \hat{Z}_{mn}^{MM} I_n + \sum_{n=1}^{N_D} \hat{Z}_{mn}^{MD} D_n = \upsilon_m^M; \quad n = 1,....N_M \tag{2.3.22}$$

where

$$\begin{aligned} \upsilon_m^D &= \int_V K(\vec{r}) \vec{f}_m(\vec{r}) \cdot \vec{E}^i dv; \quad m = 1,....,N_D \\ \upsilon_n^M &= \int_S \vec{f}_n^S(\vec{r}) \cdot \vec{E}^i{}_{\tan} ds; \quad n = 1,....N_M \end{aligned} \tag{2.3.23}$$

and

$$Z_{mn}^{MD} = -\frac{\omega^2 \mu_0}{4\pi}\sum_{p=1}^{2}\sum_{p'=1}^{P} K_{p'} \int_{S_p}\int_{V_{p'}} \vec{f}_n^{\,S}(\vec{r}) \cdot \vec{f}_{mp'}(\vec{r}\,') g(\vec{r},\vec{r}\,') dv' ds$$

$$-\frac{1}{4\pi\varepsilon_0}\sum_{p=1}^{2}\sum_{q=1}^{Q} \hat{K}_q \int_{S_p}\int_{\Omega_{q'}} (\nabla_S \cdot \vec{f}_n^{\,S}(\vec{r})) f_{\perp mq}(\vec{r}\,') g(\vec{r},\vec{r}\,') d\Omega' ds \quad m = 1,..., N_D ; n = 1,..., N_M \tag{2.3.24}$$

$$Z_{mn}^{DM} = \frac{j\omega\mu_0}{4\pi}\sum_{p=1}^{2}\sum_{p'=1}^{P} K_{p'} \int_{V_p}\int_{S_{p'}} \vec{f}_{np}(\vec{r}) \cdot \vec{f}_m^{\,S}(\vec{r}\,') g(\vec{r},\vec{r}\,') ds' dv$$

$$+\frac{j}{4\pi\varepsilon_0\omega}\sum_{p=1}^{2}\sum_{q=1}^{Q} \hat{K}_q \int_{\Omega_q}\int_{S_{p'}} f_{\perp nq}(\vec{r}) \cdot (\nabla_S \cdot \vec{f}_m^{\,S}(\vec{r}\,')) g(\vec{r},\vec{r}\,') ds' d\Omega \quad n = 1,..., N_D ; m = 1,..., N_M \tag{2.3.25}$$

From equations (2.3.24) and (2.3.25) we see that

$$\hat{Z}^{DM} = \left(\hat{Z}^{MD}\right)^T /(j\omega) \tag{2.3.26}$$

where the superscript T denotes the transpose matrix.

2.4 Total impedance matrix

The total impedance matrix is obtained by combining the metal impedance matrix $\hat{Z}^{MM}$, the dielectric impedance matrix $\hat{Z}^{DD}$ and the mutual impedance matrices $\hat{Z}^{DM}$ and $\hat{Z}^{MD}$.

$$\hat{Z} = \begin{bmatrix} \hat{Z}^{MM} & \hat{Z}^{MD} \\ \hat{Z}^{DM} & \hat{Z}^{DD} \end{bmatrix} \tag{2.4.1}$$

The impedance matrix Z can be converted into a symmetric matrix form by using trivial transformations. One way of achieving it is

$$\hat{Z} = \begin{bmatrix} j\omega\hat{Z}^{MM} & \hat{Z}^{MD} \\ j\omega\hat{Z}^{DM} & \hat{Z}^{DD} \end{bmatrix} \tag{2.4.2}$$

Once the Z matrix is obtained we solve the system of equation in the form

$$\vec{V} = \hat{Z}I \tag{2.4.3}$$

The system of equations is solved using the LAPACK's `zsysv` driver routine [15] which computes the solution to a complex system of linear equations $\hat{A}X = \vec{B}$, where $\hat{A}$ is an N-by-N symmetric matrix and X and B are column vectors of length N. The diagonal pivoting method is used to factor A. The factored form of $\hat{A}$ is then used to solve the system of equations $\hat{A}X = \vec{B}$. `zsysv` only uses the upper or the lower triangular matrix, hence Z^{DM} need not be calculated.

3. Scattering problem and fields

In this section, the voltage vectors and the scattered fields (electric and magnetic) are derived for metallic and dielectric structures.

3.1 Voltage vector

The voltage vector in equation (2.4.3) is a combination of the metal voltage vector and the dielectric voltage vector, i.e.

$$\vec{V} = \left[\vec{\upsilon}^{M} \quad \vec{\upsilon}^{D}\right] \tag{3.1.1}$$

where the metal vector is given by

$$\vec{\upsilon}^{M} = \left[\upsilon_1^{M}, \ldots\ldots\ldots \quad , \upsilon_m^{M}\right] \qquad m = 1, \ldots., N_M \,] \tag{3.1.2}$$

and the dielectric voltage vector is given by

$$\vec{\upsilon}^{D} = \left[\upsilon_1^{D}, \ldots\ldots\ldots \quad , \upsilon_n^{D}\right] \qquad n = 1, \ldots, N_D \tag{3.1.3}$$

The voltage expressions υ_m^M and υ_n^D are already derived in sections 2.1 and 2.2. Hence, we will not repeat the derivation but present the final results.

From equation (2.1.12) in section 2.1 we have

$$\upsilon_m^{M} = \int_S \vec{f}_n^{\,S}(\vec{r}) \cdot \vec{E}^{i}{}_{\tan}\, ds \qquad m = 1, \ldots. N_M \tag{3.1.4}$$

Similarly from equation (2.2.21) in section 2.2 we have

$$\upsilon_n^D = \int_V K(\vec{r})\vec{f}_m(\vec{r}) \cdot \vec{E}^i dv, \qquad n = 1,....N_D \tag{3.1.5}$$

3.2 Calculation of scattered fields due to metal structure

In this section, we calculate the scattered electric and magnetic fields due to the metal currents and the charges.

a. Electric field

The scattered electric field $\vec{E}^s$ is due to surface currents and free charges on the metal surface *S*. Hence, from (2.1.12)

$$\vec{E}^s = -j\omega\vec{A}_S(\vec{r}) - \nabla\Phi_S(\vec{r}); \quad \vec{r} \; in \; V \tag{3.2.1}$$

Substituting the values of $\vec{A}_S(\vec{r})$ and $\Phi_S(\vec{r})$ from equations (2.1.19) and (2.1.20) yields

$$\begin{aligned}\vec{E}^s = -j\omega\sum_{n=1}^{N_M}\left\{\frac{\mu_0}{4\pi}\int_S \vec{f}_n^S(\vec{r}')g(\vec{r},\vec{r}')ds'\right\}I_n - \\ \nabla_r\left(\sum_{n=1}^{N_M}\left\{\frac{1}{4\pi\varepsilon_0}\frac{j}{\omega}\int_S \nabla_S \cdot \vec{f}_n^S(\vec{r}')g(\vec{r},\vec{r}')ds'\right\}I_n\right)\end{aligned} \tag{3.2.2}$$

We mention that the gradient of $\Phi(\vec{r})$ is just the gradient of the Green's function ($g(\vec{r},\vec{r}') = \exp(-jkR)/R$, $R = |\vec{r} - \vec{r}'|$) as the other terms are constant. Thus, one has

$$\vec{E}^s = \frac{-j\omega\mu_0}{4\pi}\sum_{n=1}^{N_M}\left\{\int_S \vec{f}_n^{\,S}(\vec{r}\,')g(\vec{r},\vec{r}\,')ds'\right\}I_n - \frac{j}{4\pi\omega\varepsilon_0}\sum_{n=1}^{N_M}\left\{\int_S \nabla\cdot\vec{f}_n^{\,S}(r')\nabla_r g(\vec{r},\vec{r}\,')ds'\right\}I_n \tag{3.2.3}$$

The gradient of the Green's function is obtained as

$$\nabla_r g(\vec{r},\vec{r}\,') = \nabla_r \frac{e^{-jk|\vec{r}-\vec{r}\,'|}}{|\vec{r}-\vec{r}\,'|} = -jk\nabla_r|\vec{r}-\vec{r}\,'|g(\vec{r},\vec{r}\,') - \frac{1}{|\vec{r}-\vec{r}\,'|}\nabla_r|\vec{r}-\vec{r}\,'|g(\vec{r},\vec{r}\,')$$

$$\nabla_r g(\vec{r},\vec{r}\,') = \left(-jkg(\vec{r},\vec{r}\,') - \frac{g(\vec{r},\vec{r}\,')}{|\vec{r}-\vec{r}\,'|}\right)\left(\frac{\vec{r}-\vec{r}\,'}{|\vec{r}-\vec{r}\,'|}\right) \tag{3.2.4}$$

Substituting the above expression in equation (3.2.3) we obtain

$$\vec{E}^s = \frac{-j\omega\mu_0}{4\pi}\sum_{n=1}^{N_M}\left\{\int_S \vec{f}_n^{\,S}(\vec{r}\,')g(\vec{r},\vec{r}\,')ds'\right\}I_n - \frac{j}{4\pi\omega\varepsilon_0}\sum_{n=1}^{N_M}\left\{\int_S \nabla\cdot\vec{f}_n^{\,S}(r')\left(-jkg(\vec{r},\vec{r}\,') - \frac{g(\vec{r},\vec{r}\,')}{|\vec{r}-\vec{r}\,'|}\right)\left(\frac{\vec{r}-\vec{r}\,'}{|\vec{r}-\vec{r}\,'|}\right)ds'\right\}I_n \tag{3.2.5}$$

This is the final result for the scattered electric field due to metal radiation.

b. Magnetic field

The scattered magnetic field is expressed as

$$\vec{H}^s = \frac{1}{\mu_0}\nabla\times\vec{A}_S(\vec{r});\quad \vec{r}\text{ in }V \tag{3.2.6}$$

Substituting the value of $\vec{A}_S(\vec{r})$ from equation (2.1.19) yields

$$\vec{H}^s = \frac{1}{\mu_0}\nabla_r \times \sum_{n=1}^{N_M}\left\{\frac{\mu_0}{4\pi}\int_S \vec{f}_n^S(\vec{r}')g(\vec{r},\vec{r}')ds'\right\}I_n$$

$$\vec{H}^s = \frac{1}{4\pi}\sum_{n=1}^{N_M}\left\{\int_S \nabla_r \times \vec{f}_n^S(\vec{r}')g(\vec{r},\vec{r}')ds'\right\}I_n \tag{3.2.7}$$

Applying the standard vector identity $\nabla \times u\vec{A} = u\nabla \times \vec{A} - \vec{A} \times \nabla u$ to reduce the right hand side of the above expression yields

$$\nabla_r \times \vec{f}_n^S(\vec{r}')g(\vec{r},\vec{r}') = g(\vec{r},\vec{r}')\nabla_r \times \vec{f}_n^S(\vec{r}') - \vec{f}_n^S(\vec{r}') \times \nabla_r g(\vec{r},\vec{r}') = -\vec{f}_n^S(\vec{r}') \times \nabla_r g(\vec{r},\vec{r}') \tag{3.2.8}$$

The last term $\nabla_r g(\vec{r},\vec{r}')$ in the above equation is already calculated in equation (3.2.4). Substituting from (3.2.4) in (3.2.8) and finally in (3.2.7) leads to

$$\vec{H}^s = \frac{1}{4\pi}\int_S \sum_{n=1}^{N_M}\left\{\left(-\vec{f}_S(\vec{r}') \times \left(-jkg(\vec{r},\vec{r}') - \frac{g(\vec{r},\vec{r}')}{|\vec{r}-\vec{r}'|}\right)\left(\frac{\vec{r}-\vec{r}'}{|\vec{r}-\vec{r}'|}\right)\right)ds'\right\}I_n \tag{3.2.9}$$

This is the expression for the scattered magnetic field due to metal radiation.

3.3 Calculation of scattered fields due to dielectric structure

In this section we calculate the scattered electric and magnetic fields due to volume polarization currents and charges.

a. Electric field

The scattered electric field is due to volume polarization currents in the dielectric volume V (and associated charges on its boundary Ω) as given by equation (2.2.2)

$$\vec{E}^s = -j\omega\vec{A}(\vec{r}) - \nabla\Phi(\vec{r}); \quad \vec{r} \text{ in } V \tag{3.3.1}$$

Substituting the values of $\vec{A}(\vec{r})$ and $\Phi(\vec{r})$ from equations (2.2.19) and (2.2.20) in (3.3.1)

$$\begin{aligned}\vec{E}^s = &-j\omega\left(\frac{j\omega\mu_0}{4\pi}\sum_{n=1}^{N}\left\{\sum_{p=1}^{P}K_{p'}\int_V \vec{f}_{np'}(\vec{r}')g(\vec{r},\vec{r}')dv'\right\}D_n\right)\\ &-\nabla_r\left(\frac{1}{4\pi\varepsilon_0}\sum_{n=1}^{N}\left\{\sum_{q=1}^{Q}\hat{K}_{q'}\int_\Omega f_{\perp nq'}(\vec{r}')g(\vec{r},\vec{r}')d\Omega'\right\}D_n\right); \quad \vec{r} \text{ in } V\end{aligned} \tag{3.3.2}$$

Taking into account the fact that the gradient of $\Phi(\vec{r})$ is just the gradient of the Green's function ($g = \exp(-jkR)/R,\ R = \left|\vec{r} - \vec{r}'\right|$) as the other terms are constant. Therefore (3.3.2) reduces to

$$\begin{aligned}\vec{E}^s = &\frac{\omega^2\mu_0}{4\pi}\sum_{n=1}^{N}\left\{\sum_{p=1}^{P}K_{p'}\int_V \vec{f}_{np'}(\vec{r}')g(\vec{r},\vec{r}')dv'\right\}D_n\\ &-\frac{1}{4\pi\varepsilon_0}\sum_{n=1}^{N}\left\{\sum_{q=1}^{Q}\hat{K}_{q'}\int_\Omega f_{\perp nq'}(\vec{r}')\nabla_r g(\vec{r},\vec{r}')d\Omega'\right\}D_n\end{aligned} \tag{3.3.3}$$

The gradient of the Green's function is obtained from equation (3.2.4). Substituting the corresponding expression from equation (3.2.4) in equation (3.3.3) we obtain

$$\begin{aligned}\vec{E}^s = {} & \frac{\omega^2\mu_0}{4\pi}\sum_{n=1}^{N}\left\{\sum_{p=1}^{P}K_{p'}\int_V \vec{f}_{np'}(\vec{r}')g(\vec{r},\vec{r}')dv'\right\}D_n \\ & -\frac{1}{4\pi\varepsilon_0}\sum_{n=1}^{N}\left\{\sum_{q=1}^{Q}\hat{K}_{q'}\int_\Omega f_{\perp nq'}(\vec{r}')\left(-jkg(\vec{r},\vec{r}')-\frac{g(\vec{r},\vec{r}')}{|\vec{r}-\vec{r}'|}\right)\left(\frac{\vec{r}-\vec{r}'}{|\vec{r}-\vec{r}'|}\right)d\Omega'\right\}D_n\end{aligned} \tag{3.3.4}$$

This is the final result for the scattered electric field created by a dielectric structure.

b. Magnetic field

The scattered magnetic field is expressed as

$$\vec{H}^s = \frac{1}{\mu_0}\nabla_r \times \vec{A}(\vec{r}); \quad \vec{r} \text{ in } V \tag{3.3.5}$$

Substituting the value of $\vec{A}(\vec{r})$ from equation (2.2.19) yields

$$\vec{H}^s = \frac{1}{\mu_0}\nabla_r \times \frac{j\omega\mu_0}{4\pi}\sum_{n=1}^{N}\left\{\sum_{p=1}^{P}K_{p'}\int_V \vec{f}_{np'}(\vec{r}')g(\vec{r},\vec{r}')dv'\right\}D_n$$

$$\vec{H}^s = \frac{j\omega}{4\pi}\sum_{n=1}^{N}\left\{\sum_{p=1}^{P}K_{p'}\int_V \nabla_r \times \vec{f}_{np'}(\vec{r}')g(\vec{r},\vec{r}')dv'\right\}D_n \tag{3.3.7}$$

Applying the standard vector identity $\nabla \times u\vec{A} = u\nabla \times \vec{A} - \vec{A} \times \nabla u$ to reduce the right hand side of the above expression, one has

$$\nabla_r \times \vec{f}_{np'}(\vec{r}')g(\vec{r},\vec{r}') = g(\vec{r},\vec{r}')\nabla \times \vec{f}_{np'}(\vec{r}') - \vec{f}_{np'}(\vec{r}')\times\nabla g(\vec{r},\vec{r}') = -\vec{f}_{np'}(\vec{r}')\times\nabla g(\vec{r},\vec{r}') \qquad (3.3.8)$$

The last term $\nabla_r g(\vec{r},\vec{r}')$ is already calculated in (3.3.4). Substituting from (3.3.4) into (3.3.8) and finally in (3.3.7) gives

$$\vec{H}^s = \frac{j\omega}{4\pi}\sum_{n=1}^{N}\left\{\sum_{p=1}^{P}K_{p'}\int_V\left(-\vec{f}_{np'}(\vec{r}')\times\left(-jkg(\vec{r},\vec{r}')-\frac{g(\vec{r},\vec{r}')}{|\vec{r}-\vec{r}'|}\right)\left(\frac{\vec{r}-\vec{r}'}{|\vec{r}-\vec{r}'|}\right)\right)dv'\right\}D_n \qquad (3.3.9)$$

This is the expression for the scattered magnetic field in the dielectric.

3.4 Poynting vector

In the far field, $\vec{E}$ and $\vec{H}$ are perpendicular to each other and the direction of propagation. Thus, a right handed coordinate system is formed by $\vec{E}$ and $\vec{H}$ field vectors and by the direction of propagation. Hence, we obtain the relation

$$\vec{E}(r) = \eta\vec{H}(r)\times\frac{\vec{r}}{r},\ \vec{H}(r) = \eta\frac{\vec{r}}{r}\times\vec{E}(r) \qquad (3.4.1)$$

where η is the intrinsic impedance of free space and is given by the expression $\eta = \sqrt{\frac{\mu_0}{\varepsilon_0}} \approx 377\ \Omega$.

The electric field is measured in V/m while the magnitude of the H field is measured in A/m. The cross product of $\vec{E}$ and $\vec{H}^*$ (complex conjugate) is a real quantity. This product divided by 2, is the time- averaging Poynting vector. The Poynting vector at a point gives the average power density of the radiated field, per unit area

$$\vec{W}(r) = \frac{1}{2}\mathrm{Re}\left[\vec{E}(r) \times \vec{H}^{*}(r)\right] \tag{3.4.2}$$

and has the units of $W/_{m^2}$.

4. Radiation problem and fields

In this section we review various feed models used for the radiation problem and calculate the voltage vectors for the feed models.

4.1 Feed Models

a. Delta gap

The delta gap source modeling is the simplest and most widely used excitation model [11, 20, 21]. However it is the least accurate, especially for the purpose of estimating the feed impedance. It is most accurate for smaller gap width [11]. In this model it is assumed that the excitation voltage at the feed terminals is a constant value (1 V) and zero elsewhere. Therefore the incident electric field is also constant over the feed gap and zero elsewhere, hence the name delta gap.

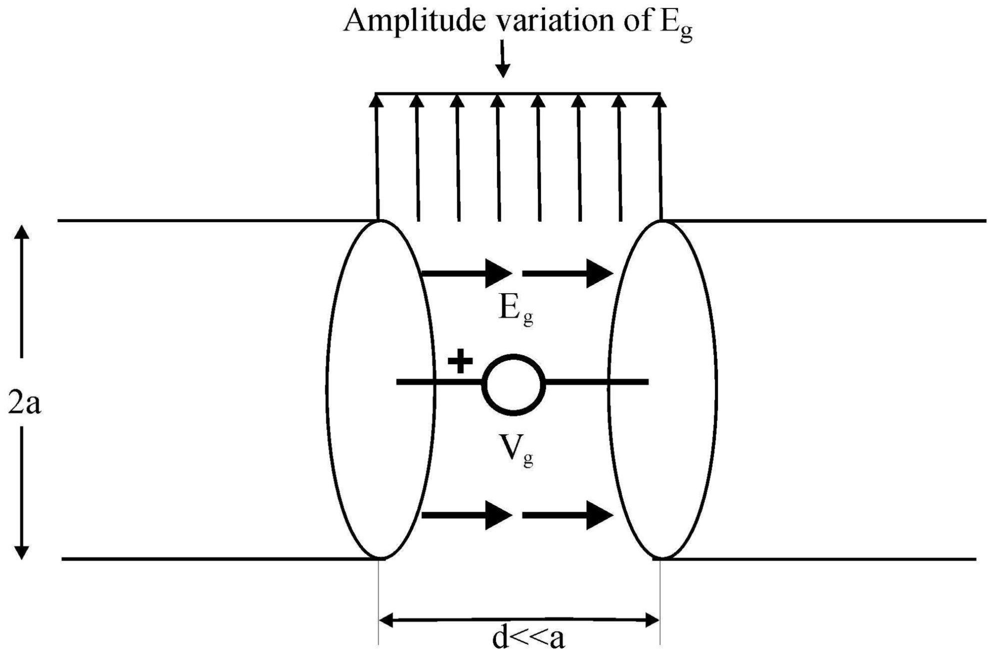


Fig 4.1.1 Delta gap source configuration.

Drawbacks of delta gap [23]

In a delta gap model placing a one in the generalized voltage matrix satisfies the 1V discontinuity in the scalar potential, but fails to satisfy the boundary condition on the impressed electric field, E^i. This failure stems form the fact that since E^i is tested only with the testing function, the surface area of the dipole which the tested field illuminates changes for each *N*. In other words a new boundary value problem is solved for each *N*. As a result the solution for the coefficients may not converge. Another major drawback of the delta function generator is that it predicts infinite "gap" capacitance. To circumvent this problem an extended delta gap source model is introduced.

b. Extended delta gap

The extended delta-gap model [23] assumes that the impressed electric field is distributed not only over the artificial gap of infinitesimally small width but also along the rest of the conductor in the vicinity of the gap. The length of this area is proportional to the physical diameter (width) of the conductor so that a convergence can be obtained for the MoM solution. Also, the field distribution along the conductor is assumed to be of the Gaussian type. The extended delta gap source model is illustrated in fig 4.1.2. In this case the unknown current is expanded in terms of series of subdomain basis functions.

$$J\hat{z} = \hat{z}\sum_{i=1}^{N} c_i J_i \tag{4.1.1}$$

where

$$J_i = \frac{T_i}{2\pi a_w} \tag{4.1.2}$$

The c_i are the unknown coefficients and N is the total number of basis functions, T is the triangle function and a_w is the dipole radius.

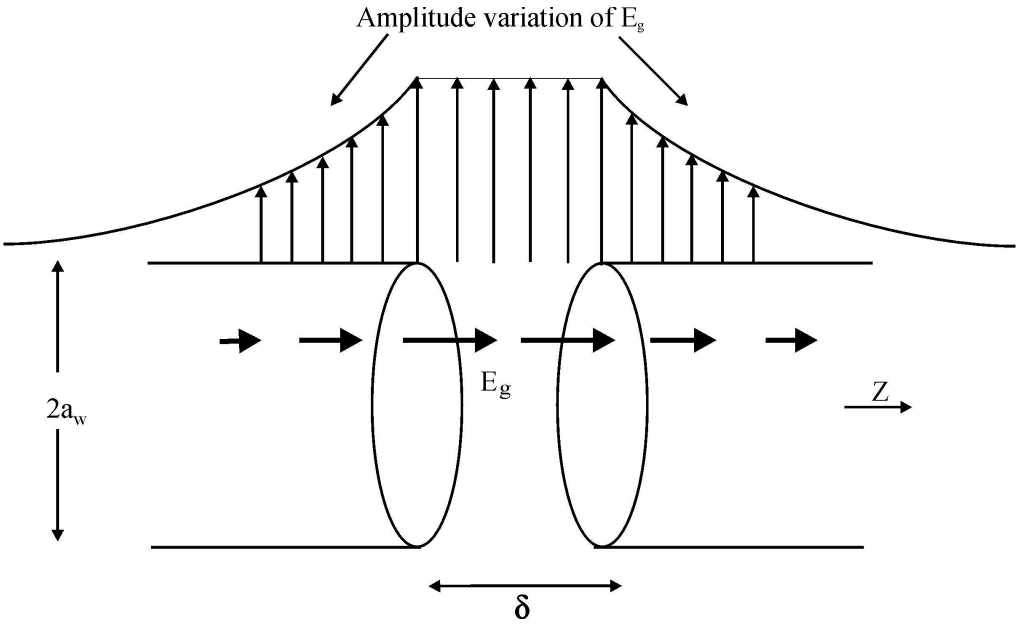


Fig 4.1.2 Modified Delta gap model.

E_g is a unit strength electric field of two variables impressed over an artificially constructed gap of length δ which may be written symbolically as

$$E_g^P = \hat{z}\frac{1}{\rho}\delta(\rho - a_w)\left[u\left(z+\frac{\delta}{2}\right) - u\left(z-\frac{\delta}{2}\right)\right] \tag{4.1.3}$$

The superscript P is used to indicate that the z variation of this field is just a unit impulse which spans the delta gap. The gap length δ is treated as an unknown constant to be determined by requiring the tested gap field to be such that $\sum\left\langle J_j, E_g^P\right\rangle = 1$ where J_j is the testing basis function. Formation of the inner product results in the following equations.

$$\sum\left\langle J_j, E_g^P\right\rangle = \sum_{j=1}^{N} \int_{\phi=0}^{2\pi} \int_{\rho=0}^{\infty} \int_{z=-\delta/2}^{\delta/2} \frac{T_j}{2\pi a_w} \frac{1}{\rho} \delta(\rho - a_w) \left[u\left(z + \frac{\delta}{2} \right) - u\left(z - \frac{\delta}{2} \right) \right] \rho dz d\rho d\phi = 1$$

(4.1.4)

Upon solution we have δ = a_w. The expanded delta source model is then implemented by replacing $E_g^P \big/ a_w$ by the Gaussian function

$$E_g^G = \frac{1}{\sqrt{2\pi}\delta_0} e^{\frac{-z^2}{2\delta_0^2}} \qquad (4.1.5)$$

where δ_0 is equal to a_w /2. The reason for this replacement is that the Gaussian function yields a solution which converges more rapidly than $E_g^P \big/ a_w$.

Since δ is a fixed length and the Gaussian function is smooth, a stable solution can be achieved since the electric field boundary condition will be satisfied for all *N*. This model is also appealing as this delta gap model will never have infinite capacitance. This is because as $\delta \to 0, a_w \to 0$ and hence there would be no feed.

c. Magnetic frill

The magnetic frill [24] model starts with the same uniform E-field distribution within the gap as shown in Fig 4.1.3. According to the surface equivalence principle [1] the radiation effect of this distribution is equivalent to the radiation of the coaxial rings of the surface magnetic current as shown in Fig 4.1.3.

In the generator representation sketched in Fig.4.1.3 the magnetic frill is thus the source of the impressed electric field, $\vec{E}_i$. This field can be computed in terms of the electric vector

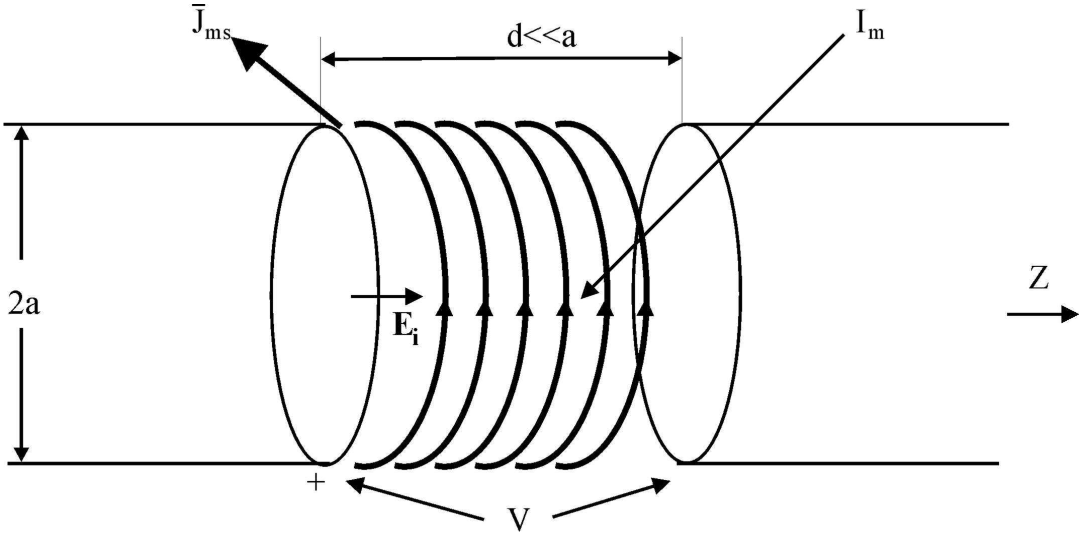


Fig 4.1.3 Magnetic frill model.

potential, $\vec{A}_e$, as

$$\vec{E}_i = -\frac{1}{\varepsilon}\nabla \times \vec{A}_e \tag{4.1.6}$$

where

$$\vec{A}_e = \frac{\varepsilon}{4\pi}\int_S \vec{J}_{ms}\, g(\vec{r})ds \tag{4.1.7}$$

S is the surface of the finite cylinder of length *d*, $\vec{r}$ is the distance between the element ds and the field point, $\vec{J}_{ms}$ is the magnetic current density and $g(\vec{r})$ is the Green's function given by $g(\vec{r}) = \exp(-jkR)/R,\ R = |\vec{r} - \vec{r}'|$. Combining equations (4.1.6) and (4.1.7), introducing the "curl" operator under the integral and noting that it operates only on $g(\vec{r})$ (as $\vec{J}_{ms}$ is constant over the surface) we get

$$\vec{E}_i = -\frac{1}{\varepsilon}\nabla\times\frac{\varepsilon}{4\pi}\int_S \vec{J}_{ms}\, g(\vec{r})ds = -\frac{1}{4\pi}\int_S \left(g(\vec{r})\nabla\times\vec{J}_{ms} - \vec{J}_{ms}\times\nabla g(\vec{r})\right)ds = \frac{1}{4\pi}\int_S \vec{J}_{ms}\times\nabla g(\vec{r})ds \tag{4.1.8}$$

The gradient of the Green's function is obtained from equation (3.2.4). Substituting the value form (3.2.4) in equation (4.1.8) we get

$$\vec{E}_i = \frac{1}{4\pi}\int_S \vec{J}_{ms}\times\left(-jkg(\vec{r},\vec{r}') - \frac{g(\vec{r},\vec{r}')}{|\vec{r}-\vec{r}'|}\right)\left(\frac{\vec{r}-\vec{r}'}{|\vec{r}-\vec{r}'|}\right)ds \tag{4.1.9}$$

4.2 Radiation pattern

The power of the radiated field per unit area as given by equation (3.5.3) is also the radiation density of the radiated signal. In the far-field region, the radiation density has only one radial component denoted by W. The radiation intensity is defined as the radiation density multiplied by a factor r^2. Now, the radiation intensity is introduced as

$$U = r^2 W \tag{4.2.1}$$

The radiation intensity has the units of watts per solid angle and is theoretically independent of the radius of the sphere surrounding the antenna if the sphere radius is large enough compared to the antenna size and wavelength. The total radiated power P_{rad} is obtained as the integral of the radiation density on the sphere surface. If a triangular mesh is used for the sphere, the total radiated power is obtained as the sum of the products of the radiation density in the middle of the mesh triangle and the triangle area. The summation is done over all the triangles of the sphere mesh. The antenna directivity is the normalized radiation intensity calculated in dB

$$D = 10\log_{10}\frac{U}{U_0} \tag{4.2.2}$$

The normalization factor U_0 is the total radiated power per unit solid angle

$$U_0 = \frac{P_{rad}}{4\pi} \tag{4.2.3}$$

Antenna radiation pattern is the directivity plot in terms of the polar coordinates.

5. Simulations results

The theory described in sections 2, 3 and 4 was implemented in the Matlab and C/C++ codes, compiled under Matlab environment as `mex` files [25]. The use of mixed C/Matlab codes helped in speeding up the algorithm and memory optimization. The solver was then tested for both homogenous and inhomogeneous pure dielectric (scattering) and combined metal-dielectric structures (radiation and scattering). This section provides a summary of the test results obtained in each case.

5.1 Test of simulations for a pure dielectric structure (scattering)

In this section, scattering of homogenous and inhomogeneous pure dielectric structures is tested for different mesh configurations. The simulation algorithm follows the model developed in section 2.2. The corresponding source codes are given in Appendix C.

a. Homogeneous dielectric plate

The first example is a homogeneous dielectric plate of the size 4cm×4cm×0.4cm (Fig. 5.1.1a), with $\varepsilon_r = 10$. The frequency of the *y*-polarized plane wave incident in the negative *z*-direction is either 750 MHz (plate size is $\approx \lambda_{\text{diel}}/3$) or 1,500 MHz (plate size is $\approx 2\lambda_{\text{diel}}/3$). A uniform tetrahedral mesh has one layer into the depth but has variable discretization in the *xy*-plane. The scattered electric field in the dielectric has the dominant *y*-component and is evaluated on a line $x = -0.1\,\text{cm}, \text{-}\ 2\,\text{cm} < y < 2\,\text{cm}, z = 0\,\text{cm}$, which is slightly off the *y*-axis. The offset is introduced in order to avoid field calculations exactly along tetrahedral faces.

Fig. 5.1.1b shows the scattered field as a function of the distance along the *y*-axis for the mesh with 864 tetrahedra at 750 MHz – face-based basis functions. Similar results for the edge-based functions are given in Fig. 5.1.1c. The Ansoft solution uses about 64,000 tetrahedra and is shown by a dashed curve.

The relative error E between MoM and FEM solutions

$$\mathrm{E} = \sqrt{\int_{-0.02}^{0.02} \left| E_y^{MoM} - E_y^{ANSOFT} \right|^2 dy \Big/ \int_{-0.02}^{0.02} \left| E_y^{ANSOFT} \right|^2 dy} \tag{5.1.1}$$

is shown in Fig. 5.1.2 a, b for both sets of basis functions. Total 14 uniform meshes were considered, ranging from 54 tetrahedra (3x3 plate) to 1,536 tetrahedra (16x16 plate).

Fig. 5.1.2a and 5.1.2b correspond to 750 MHz and 1,500 MHz, respectively. The error for the edge-based functions is shown by a bold curve (crosses); the error for the face-based functions is shown by a thin curve (circles). The edge-based functions possess nearly the same or a smaller error for the given mesh compared to the face-based functions. The number of unknowns is however, considerably smaller. The number of unknowns for all meshes used in this example is shown in Fig. 5.1.2c. Bold line indicates edge-based functions; thin line is related to the face-based functions. The ratio of unknowns is equal to 0.61 for the rough 3x3 mesh and then decreases to 0.58 for the finest 16x16 mesh.

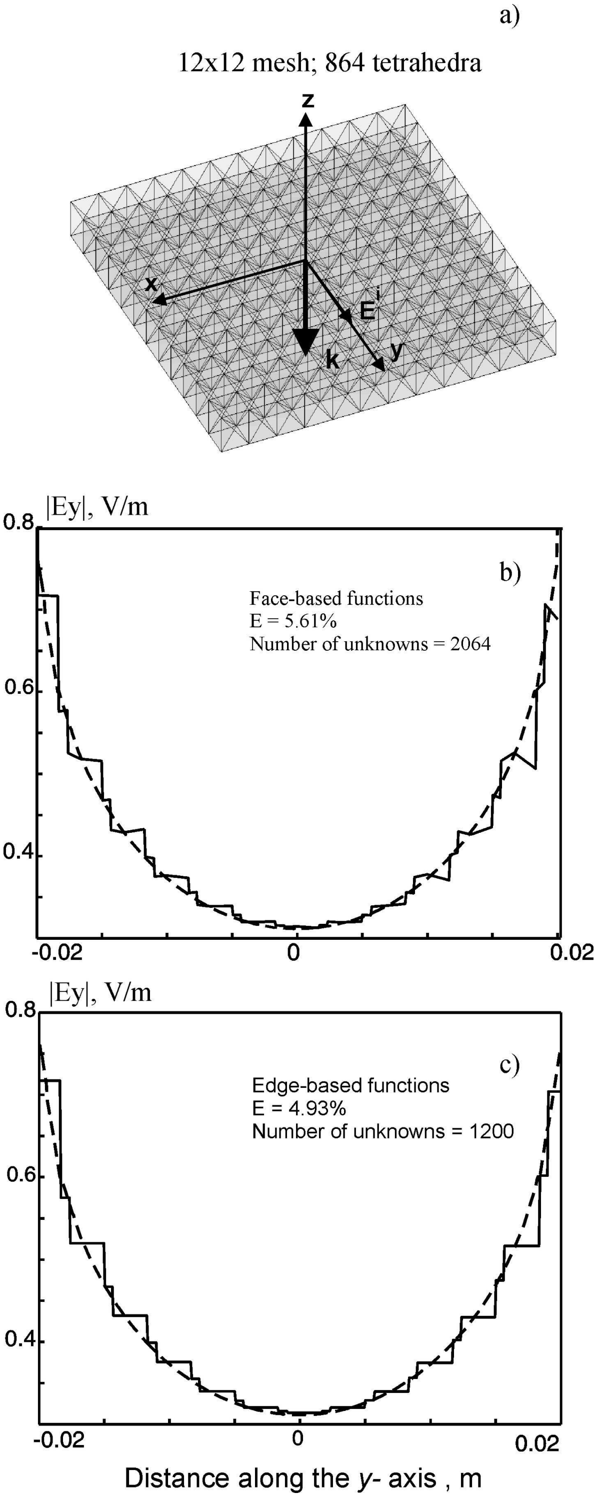


Fig. 5.1.1. a) Square dielectric plate meshed into 864 tetrahedra; b) dominant component of the scattered field $\left|E_y^s\right|$ within the dielectric plate on the *y*-axis compared to Ansoft HFSS solution (dashed curve) at 750 MHz – face-based functions; c) the same result for the edge-based functions.

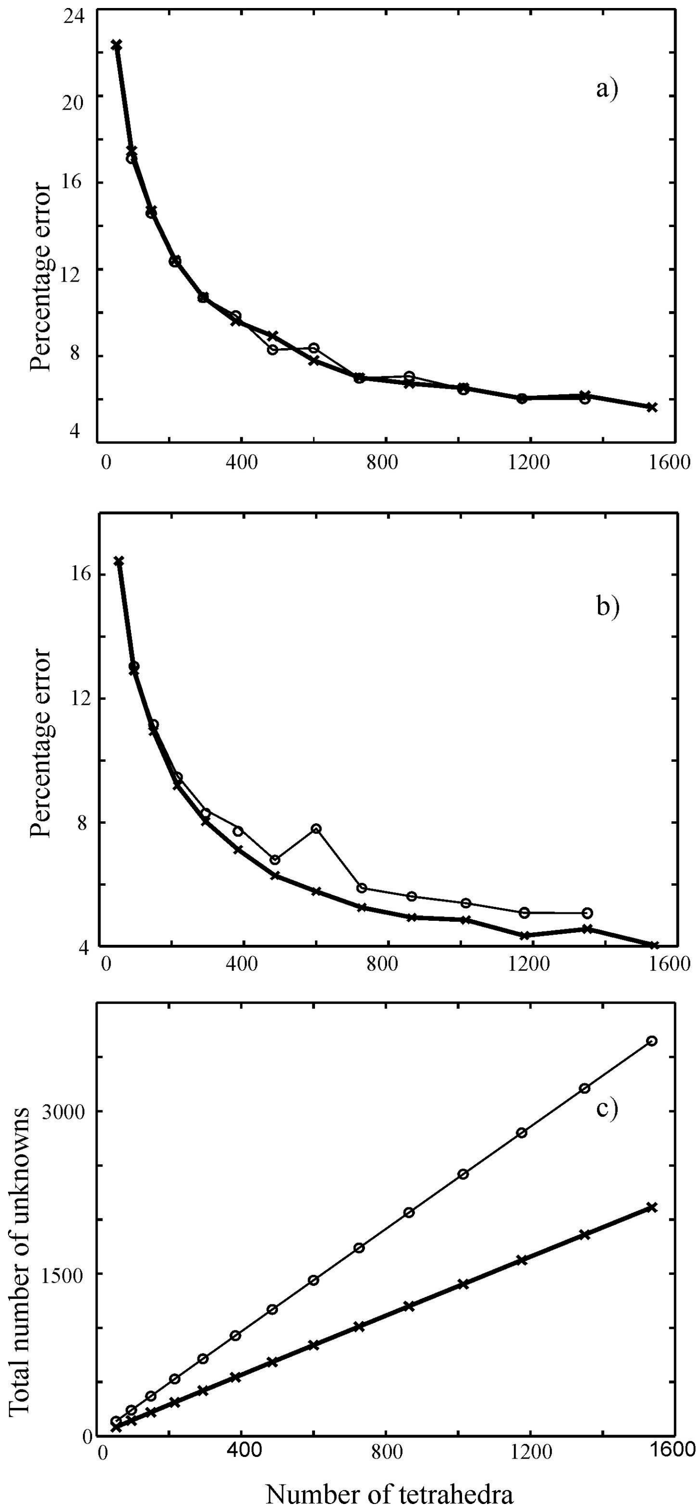
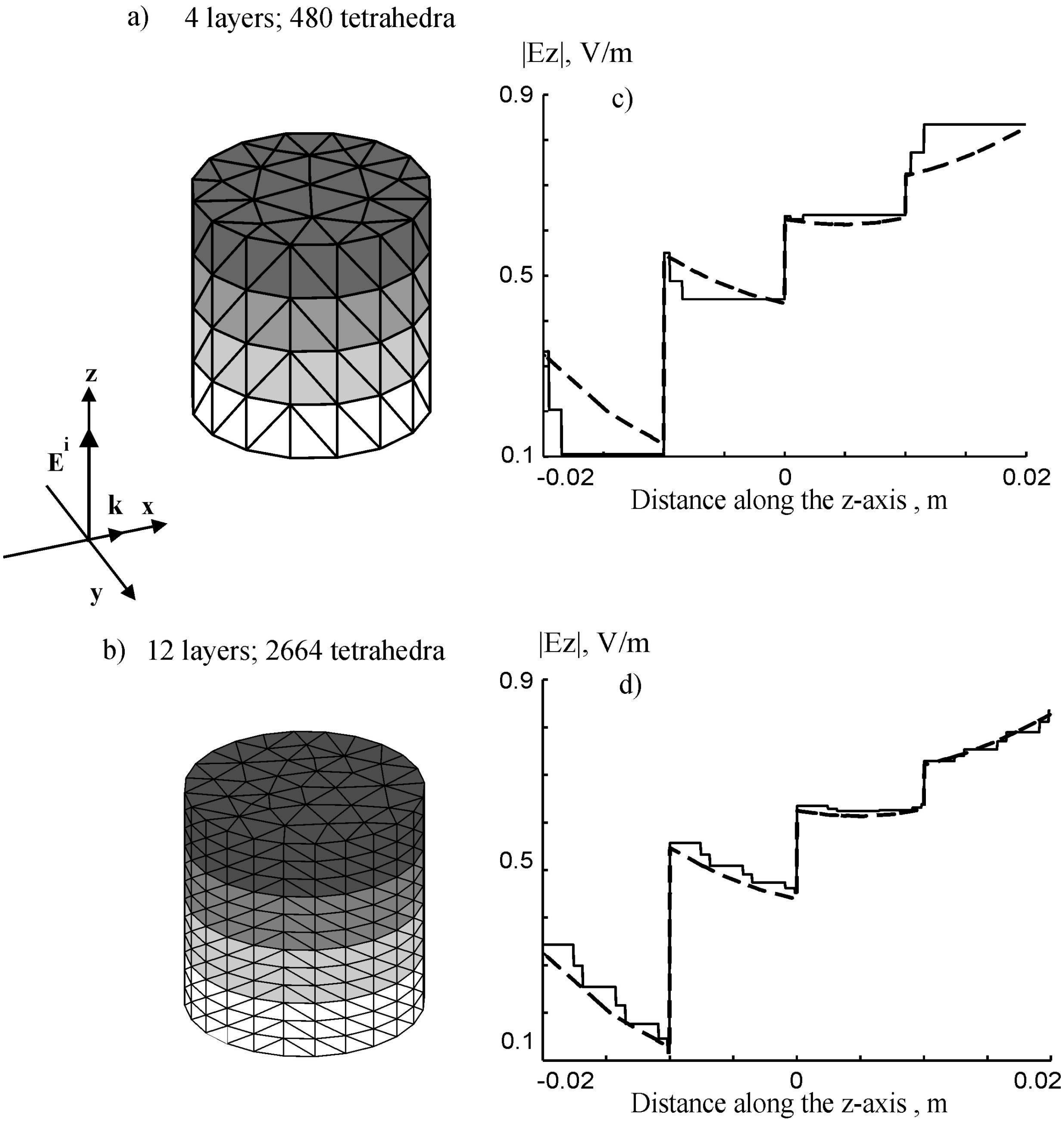


Fig. 5.1.2. a) L_2 error of the edge-based set (bold curve) and of the face-based set (thin curve) as a function of the total number of tetrahedra in the mesh for 14 different plate meshes at 750 MHz; b) the same data for 1,500 MHz; c) total number of unknowns for two sets of basis functions: thick line – edge-based functions; thin line – face-based functions.

b. Inhomogeneous dielectric cylinder

The second example is an inhomogeneous finite cylinder of radius 2 cm and the height of 4 cm. The cylinder has 4 horizontal layers, each with different dielectric constant (Fig. 5.1.3 a, b). The height of every layer is 1 cm. The dielectric constant varies from 10 for the top section to 2.5 for the bottom section, in steps of 2.5. The frequency of the *z*-polarized plane wave incident in the positive *x*-direction is 750 MHz. A tetrahedral mesh has variable discretization accuracy along the *z*-axis; the discretization accuracy of the cylinder cap also changes from a rough mesh in Fig. 5.1.3a to a finer mesh in Fig. 5.1.3b. The scattered field in the dielectric has the dominant *z*-component, which is compared to the Ansoft HFSS solution. The field is evaluated on the cylinder axis $x = 0\,\mathrm{cm}$, $y = 0\,\mathrm{cm}$, $-2\,\mathrm{cm} < z < 2\,\mathrm{cm}$. Fig. 5.1.3c, d show the scattered electric field as a function of the distance along the axis for two cylinders depicted in Fig. 5.1.3a, b with 480 and 2,664 tetrahedra (1,064 and 5,698 faces, respectively). The corresponding Ansoft HFSS solution is shown by a dashed curve. The MoM solution with the edge-based functions (solid line) is always stair like.

The error *E* between MoM and FEM solutions is calculated following equation (5.1.1). Fig. 5.1.4 shows the error E for both sets of the MoM basis functions. The bold line corresponds to the edge-based functions; thin line indicates the error of the face-based functions. The error reduces in both cases. However, the error for the edge-based elements is smaller and reaches 4% when the number of tetrahedra is about 2000. The number of unknowns for four different meshes used in this example is shown in Fig. 5.2.3. Bold line indicates edge-based functions; dashed line is related to face-based functions. The ratio of unknowns is 0.55 for the cylinder shown in Fig. 5.2.2a and 0.53 for the cylinder shown in Fig. 5.2.2b.



a) 4 layers; 480 tetrahedra

b) 12 layers; 2664 tetrahedra

Fig. 5.1.3 a, b) Inhomogeneous cylinder with four layers meshed into 480 and 2,664 tetrahedra, respectively; c) dominant component of the scattered field $\left|E_z^s\right|$ on the cylinder axis (solid curve) compared to Ansoft HFSS solution (dashed curve) for a cylinder with 480 tetrahedra; d) the same result for the cylinder with 2,664 tetrahedra.

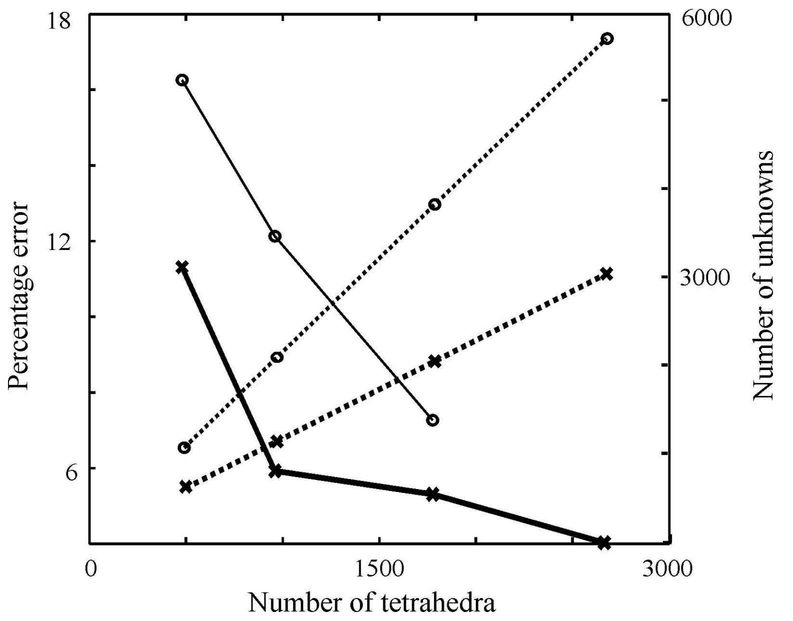


Fig. 5.1.4 Solid curves – L_2 error of the edge-based set (bold curve) and of the face-based set (thin curve) as a function of the total number of tetrahedra in the mesh for 4 different cylinder meshes at 750 MHz. Dotted lines – total number of unknowns for two sets of basis functions: thick dotted line – edge-based functions; thin dotted line– face-based functions.

c. Conclusion

Two given examples demonstrate a better performance of the edge-based basis functions. First, the edge-based basis functions have nearly the same or a faster convergence rate for equal tetrahedral meshes. Second, for the same tetrahedral mesh, the number of unknowns for the edge-based functions is considerably smaller. The ratio of unknowns (edge-based vs. face-based) ranges from 0.6 for rough plate meshes to approximately 0.5 for large volume meshes.

5.2 Test of simulations for a combined metal-dielectric structure

In this section, simulation of a combined metal-dielectric structure (a scatterer) is tested for different dielectric constants and different mesh configurations. The simulation algorithm follows the model developed in sections 2.1, 2.2 and 2.3. The corresponding source codes are given in Appendix C.

a. Dielectric plate with a ground plane

The structure under study is a homogeneous metal-backed dielectric plate on the size 4x4x0.4 cm centered at origin. The relative dielectric constant is 10. A typical structure is shown in Fig.5.2.1.

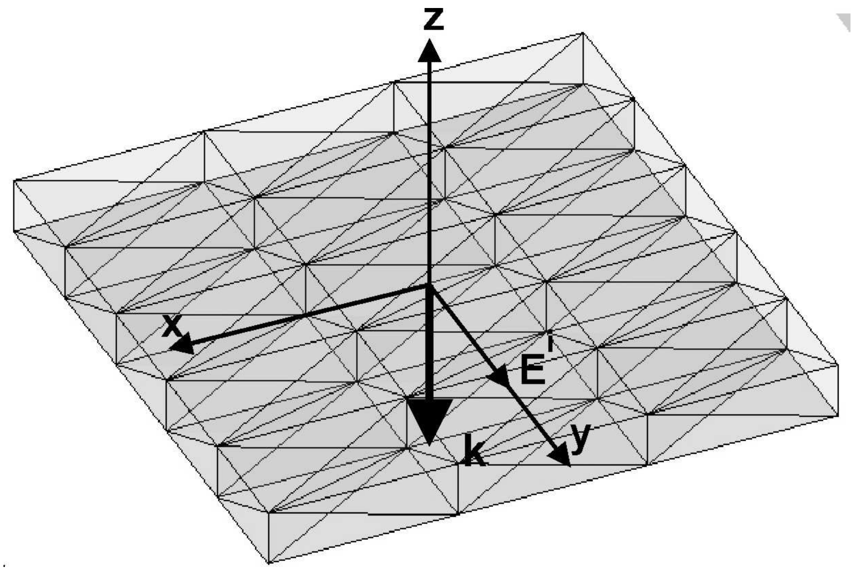


Fig.5.2.1 Metal - dielectric structure (Grid Size: 3x5).

The frequency of the *y*-polarized plane wave incident in the negative *z*-direction is either 750 MHz (the plate size is $\approx \lambda_{\text{diel}}/3$) or 1,500 MHz (the plate size is $\approx 2\lambda_{\text{diel}}/3$). The scattered electric field in the dielectric has the dominant *y*-component, but the *z*-component also becomes significant, due to free charges on the metal surface. The field is evaluated in the middle of the plate, on the plate axis at $x = 0, -2 < y < 2, z = 0\,\text{cm}$. After the MoM equations are solved, the electric field within the dielectric (the "near" field) is found for either set of basis functions. This near field is most sensitive to the numerical errors of the solver. The resulting near field is compared to the similar data obtained using Ansoft HFSS v9.0. The Ansoft solutions are computed using the PML layer and with fine tetrahedral meshes, with typically more than 60,000 tetrahedra.

Fig. 5.2.2 shows the magnitude of the *y*-component of the scattered field, $\left|E_y^s\right|$, as a function of the distance along the *y*-axis for three one-layer grids with the same number of tetrahedra (750 MHz). The left column is for the edge-based basis functions; the right column corresponds to the face-based functions. The Ansoft HFSS solution is given by a smooth thick curve.

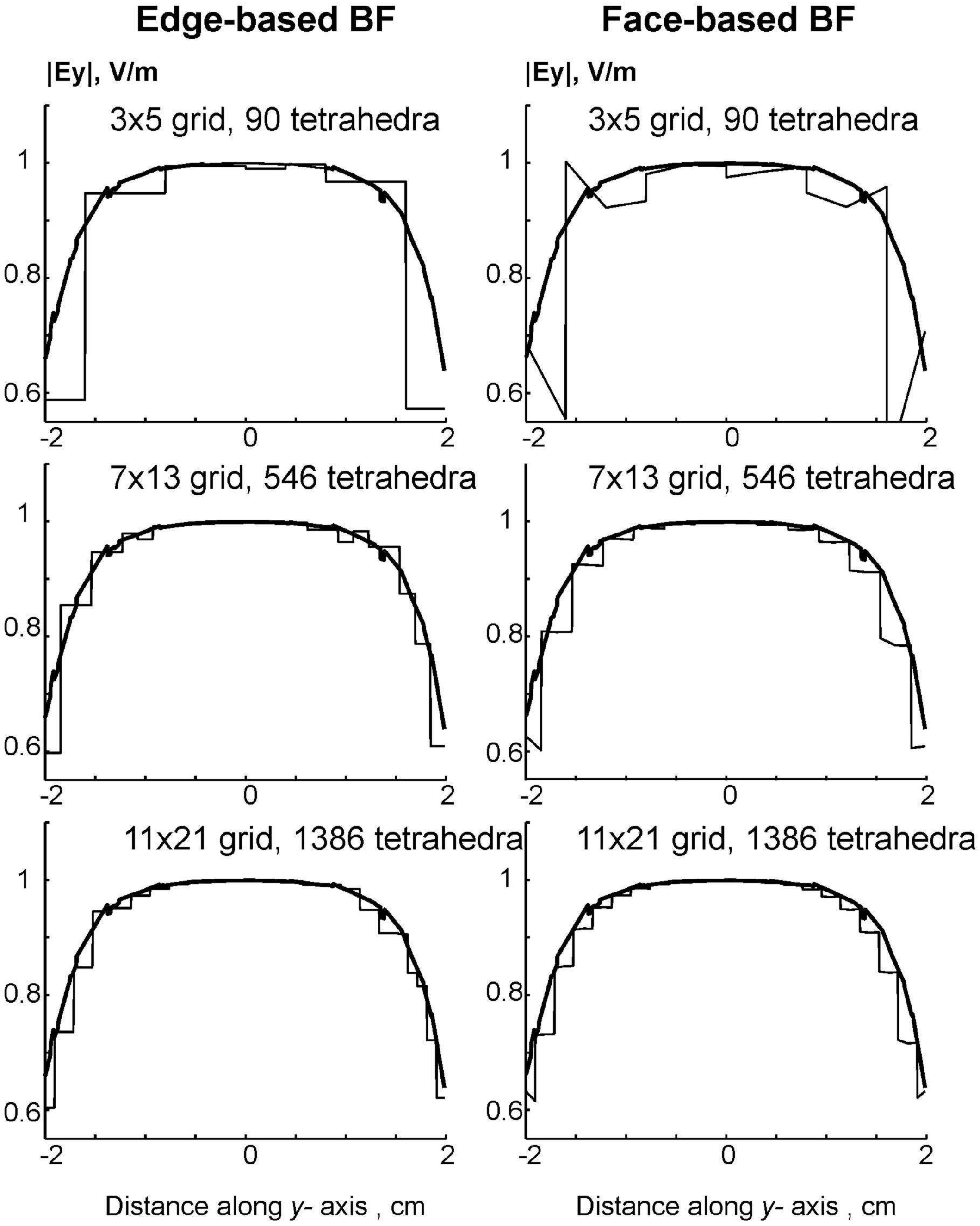


Fig. 5.2.2. Co-polar component of the scattered field $\left|E_y^s\right|$ within the dielectric plate along the y-axis compared to Ansoft solution (thick curve) at 750 MHz. Three groups of results are related to three different meshes. The left column corresponds to the edge-based functions; the right column corresponds to the face-based functions.

The same results for the component $\left|E_z^s\right|$ are presented in Fig. 5.2.3.

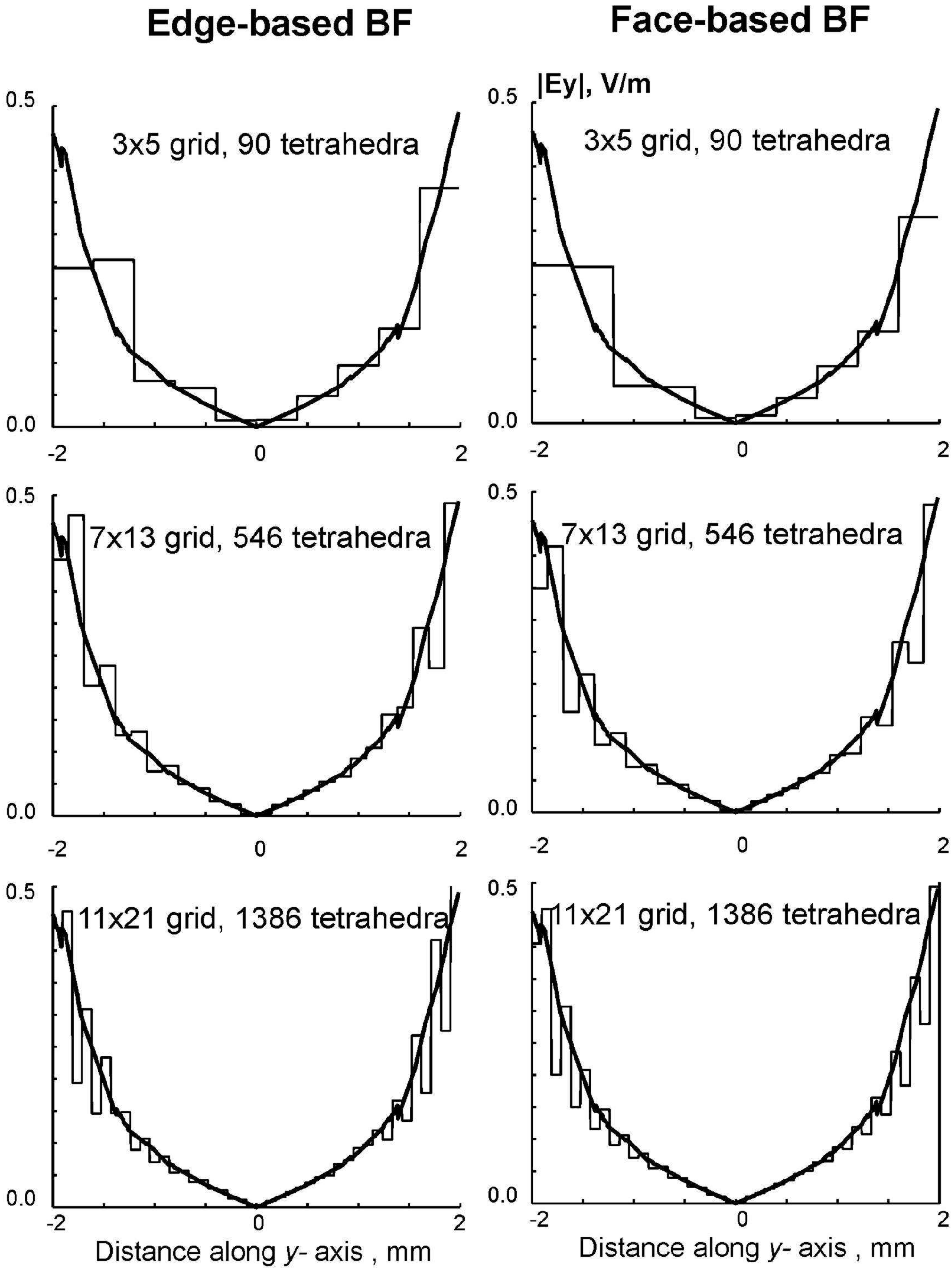


Fig. 5.2.3. Vertical component of the scattered field $\left|E_z^s\right|$ within the dielectric plate along the *y*-axis compared to Ansoft solution (thick curve) at 750 MHz. Three groups of results are related to three different meshes. The right column corresponds to the edge-based functions; the left column corresponds to the face-based functions.

In the above case, poor convergence is observed. The reason for that is the inability of both sets of basis functions to describe a significant variation of the field in the *z*-direction, using the one-layer grid. For a two-layer grid, the edge basis functions demonstrate a better performance as Fig. 5.2.4 shows.

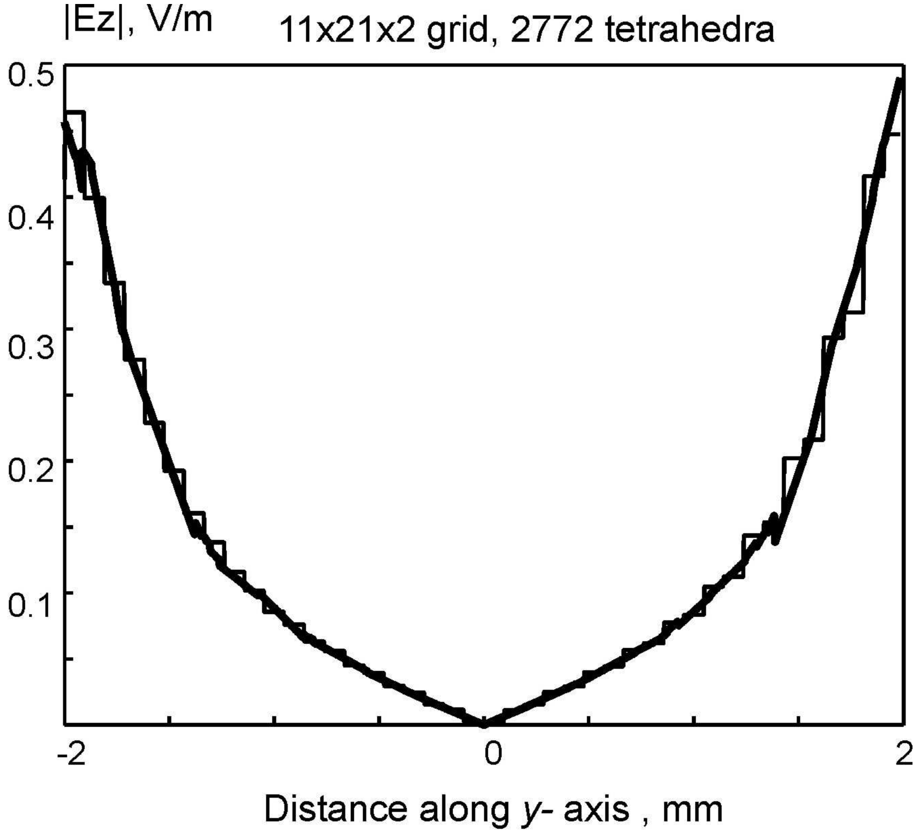


Fig. 5.2.4. Vertical component of the scattered field $|E_z^s|$ within the dielectric plate along the *y*-axis compared to Ansoft solution (thick curve) at 750 MHz (edge-based basis functions for a 11x21x2 mesh).

The L_2 error E of the MoM solution for $|E_y^s|$ is given by (5.1.1) and is shown in Fig. 5.2.5 a, b (thick line – edge-based functions; thin line – face-based functions). It is seen from Fig. 5.2.5 a, b that the edge-based basis functions possess nearly the same or an even smaller error for the given mesh, compared to the face-based functions. The total number of unknowns is however, considerably smaller.

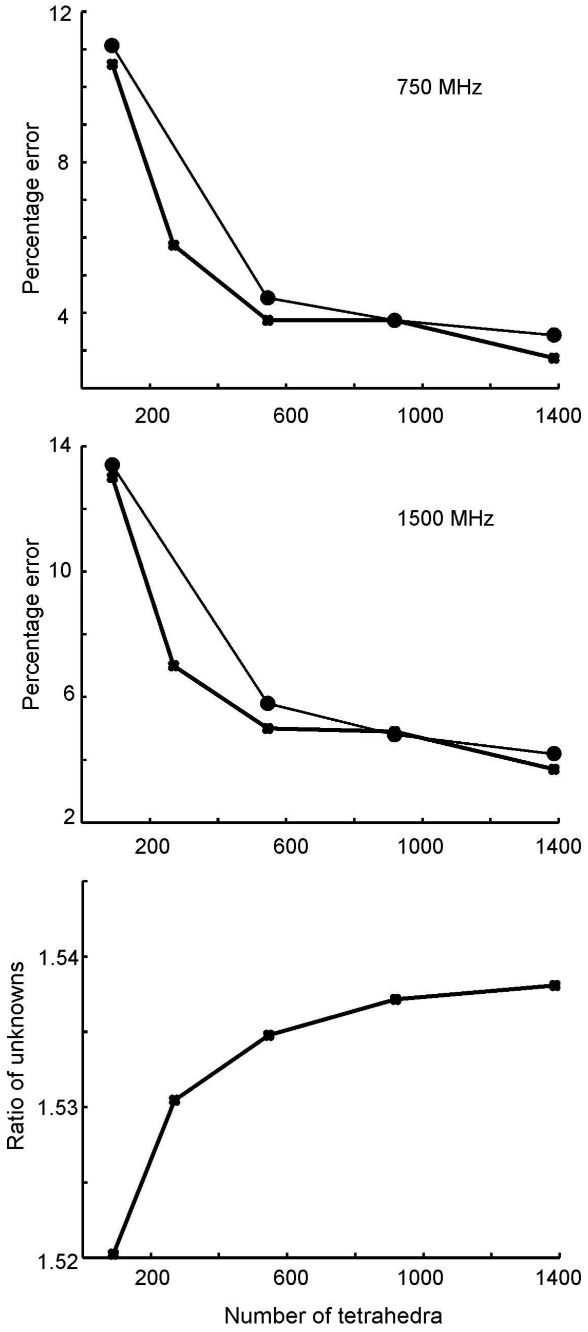


Fig. 5.2.5. a) L_2 error of the edge-based set (bold line) and of the face-based set (thin line) as a function of the total number of tetrahedra in the mesh at 750 MHz; b) the same data for 1,500 MHz; c) ratio of the number of unknowns (metal plus dielectric) for the face-based set and for the edge-based set: face-based vs. edge-based.

Fig. 5.2.5c indicates the ratio of the number of unknowns (metal plus dielectric) of the face-based model and of the edge-based model. This ratio varies at around 1.53 and increases with increasing the grid size. Note that the number of metal unknowns remains the same. Similar results are obtained for $\left|E_z^s\right|$.

These results indicate that the edge-based basis functions may be easily extended to model a metal-dielectric structure, in conjunction with the RWG basis functions for the metal sheet. They require a smaller number of unknowns compared to the face-based functions and give nearly the same or a better accuracy.

b. Resonator

We study three examples: the isolated parallel-plate metal/dielectric resonator with the relative dielectric constant $\varepsilon_r = 10$ and thickness-to-size ratio 1:20 (Fig. 5.2.6a), the same resonator but with thickness-to-size ratio 1:10 (Fig. 5.2.6b), and the isolated 2:2:1 dielectric resonator (Fig. 5.2.6c) with the relative dielectric constant $\varepsilon_r = 200$. For the parallel-plate metal-dielectric resonator, only the lowest TM_{110}^z mode is studied. Similarly, only the lowest $TE_{11\delta}^z$ resonant mode is considered for the dielectric resonator.

The eigenmode solution for the impedance matrix is obtained by the search for the local minimum of a cost function F of two variables – the reciprocal condition number of the impedance matrix

$$F(f, f') = \text{rcond}(Z(f + jf')) \tag{5.2.1}$$

using the LAPACK condition estimator. Here, f is the real part of frequency and $f' > 0$ is the imaginary part. The search procedure does not use a gradient method and implies direct evaluation of the cost function on the plane of complex frequency. The resonant frequency and the quality factor of the resonator are then obtained as

$$f_{res} = f_{\min}, Q = f_{\min} / 2f'_{\min} \tag{5.2.2}$$

Since the direct search procedure is safe but very time-consuming even for moderate meshes, its accuracy is limited by ± 0.1% for the real part of the resonant frequency f and by ± 0.2% for the imaginary part f'.

The test is made by comparison with the Ansoft HFSS v 9.0 eigenmode solver. The Ansoft solutions are obtained with a cubical PML enclosure whose size is optimized close to the first resonant frequency. Fine finite-element meshes are employed. The MoM meshes include a maximum of 1944 tetrahedra (2250 edge-based basis functions). The number of metal basis functions is considerably (up to two times) smaller. The parameter a in Fig. 5.2.6 is chosen to be 1.498 mm.

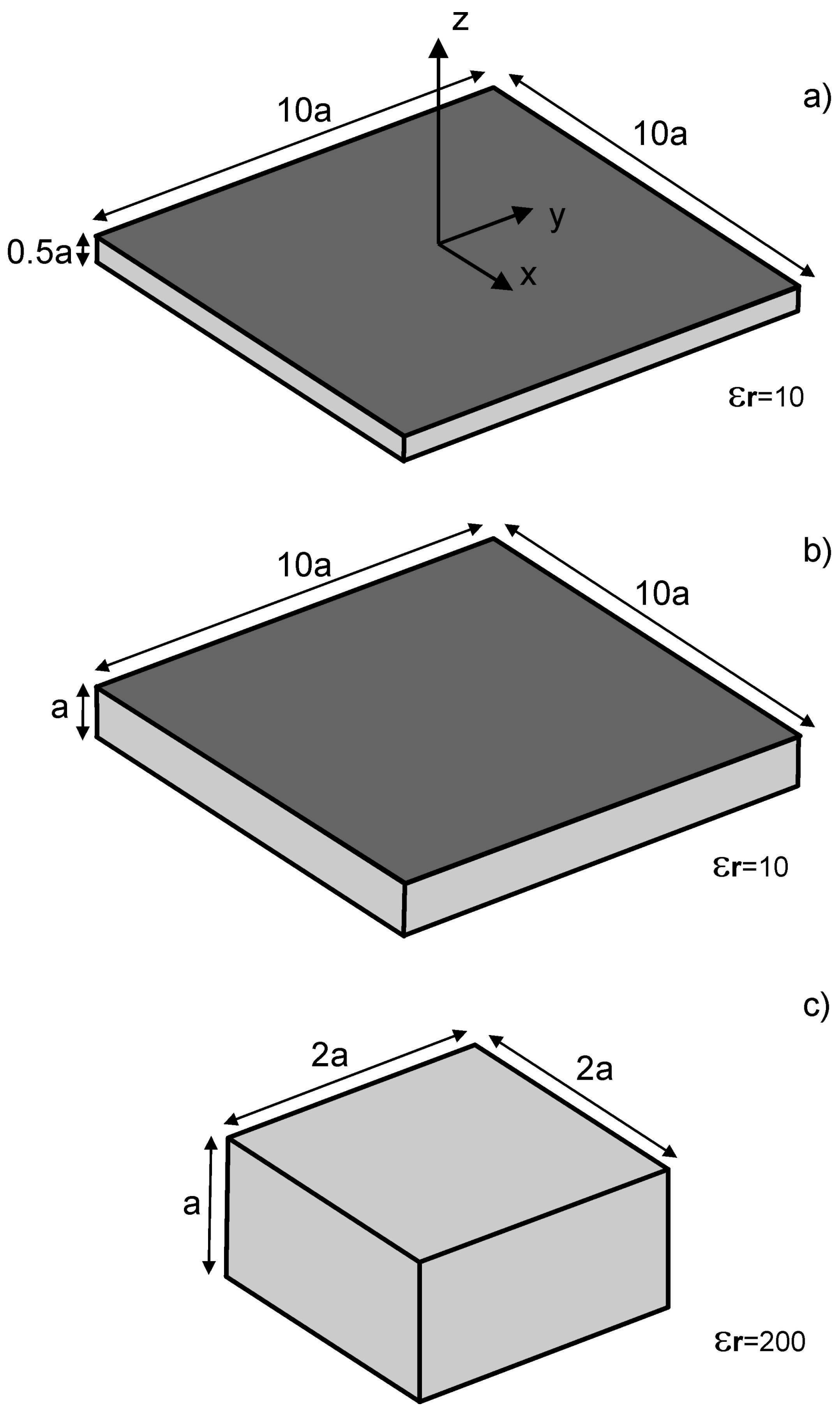


Fig. 5.2.6. Three computed resonator geometries: a) – metal-dielectric resonator with thickness-to-width ratio 1:20; b) – the same as a) but thickness-to-width ratio is 1:10; c) – dielectric rectangular resonator with a ratio of 2:2:1.

The first (metal/dielectric) resonator was studied with non-regular one-layer meshes (finer toward metal edges). The corresponding HFSS solution has a 27.5*a*×27.5*a*×14.5*a* PML enclosure, the final $\Delta f = 0.3\%$, and the final mesh uses 148,834 tetrahedra. It yields an eigen frequency of 3.2474 GHz and a *Q*-factor of 497. Fig. 5.2.7a shows the error percentage of the MoM solutions for 4 different tetrahedral/triangular MoM meshes (108, 264, 450, and 1158 tetrahedra). The dashed line corresponds to the error in the resonant frequency; the solid line gives the error of the *Q*-factor. While the error in the resonant frequency is always below 1%, the error of the *Q*-factor monotonously decreases and approaches 1% when the number of dielectric basis functions exceeds 1600. All MoM meshes have one layer of tetrahedra.

The second resonator was also studied with non-regular grids (finer toward metal edges). The corresponding HFSS solution has a 24.4*a*×24.4*a*×16.1*a* PML enclosure, the final $\Delta f = 0.2\%$, and the final mesh involves 102,582 tetrahedra. It yields an eigen frequency of 3.2970 GHz and a *Q*-factor of 226. Fig. 5.2.7b shows the error percentage of the MoM solutions for 4 different tetrahedral/triangular MoM meshes (108, 264, 450, and 1392 tetrahedra). The dashed line again corresponds to the error in the resonant frequency and the solid line gives the error of the *Q*-factor. Here again, the error in the resonant frequency is always below 1% and the error of the *Q*-factor monotonously decreases and approaches 1% when N_D exceeds 1600.

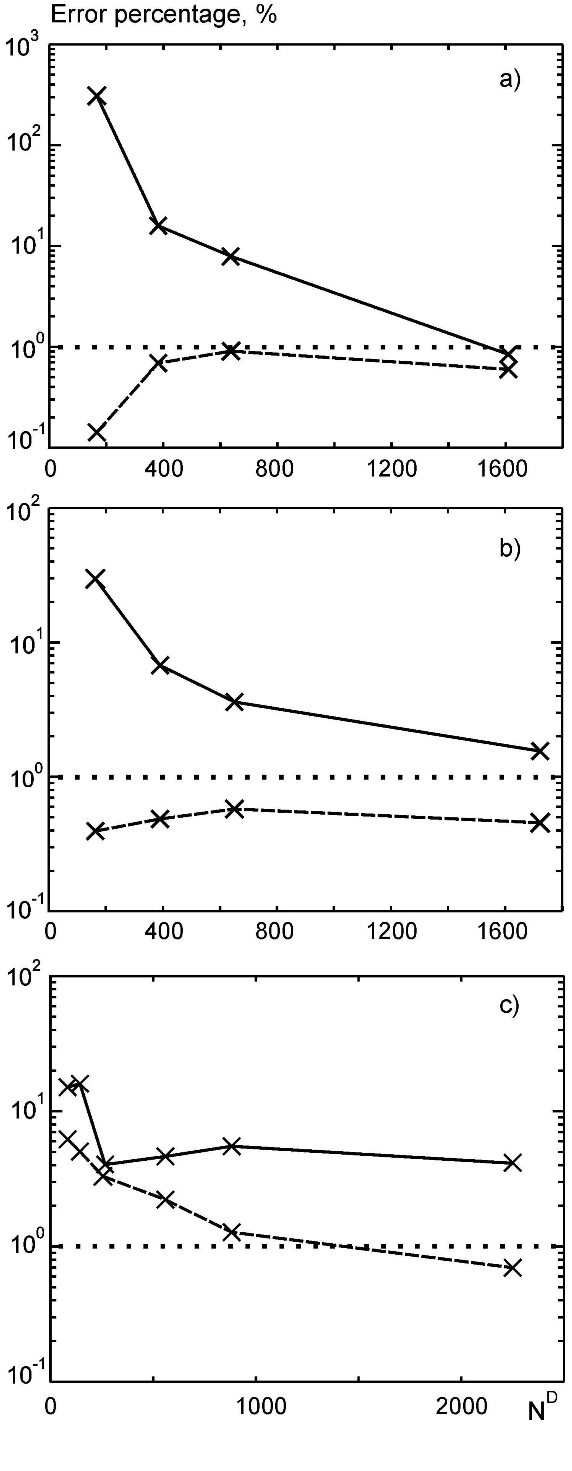


Fig. 5.2.7. Error percentage of the MoM solution as a function of the number of the dielectric basis functions N^D. Crosses denote the data for the individual meshes. a) – Parallel-plate resonator with thickness-to-width ratio 1:20; b) – parallel-plate resonator with thickness-to-width ratio 1:10; c) – dielectric rectangular resonator with $\varepsilon_r = 200$. Dashed line – error in the resonant frequency; solid line – error in the *Q*-factor.

The third (dielectric) resonator was studied with regular grids. The corresponding HFSS solution has a $54.2a\times 54.2a\times 37.3a$ PML enclosure, the final $\Delta f = 0.8\%$, and the final mesh uses 48,735 tetrahedra. We were unable to obtain a more accurate solution due to software crashes. It yields an eigen frequency equal to 1.3804 GHz and a Q-factor of 411. Fig. 5.2.7c shows the error percentage of the MoM solutions for 6 different tetrahedral/triangular MoM meshes based on regular cube grids of 3×3×1, 4×4×1, 4×4×2, 5×5×3, 7×7×3, and 9×9×4. The grids have 54, 96, 192, 450, 880, and 1944 tetrahedra. The error in the resonant frequency monotonously decreases and approaches 1% when the number of the basis functions becomes higher than 2000. The error of the Q-factor is not monotonous and is considerably higher than in the case of a thin metal/dielectric resonator. The bulk dielectric resonator required more significant computational efforts.

In order to prove the uniqueness of the eigenmode solution and the absence of artificial mode splitting we plot in Fig. 5.2.8 the inverse of the cost function for two plate meshes that gave the smallest Q-factor error in Fig. 5.2.7a and Fig. 5.2.7b, respectively. The extremum of the cost function is well developed and unique. More accurate eigenmode solutions were obtained by refining the search domain.

In order to prove the local field distribution within the resonator, we found the scattered field within each resonator when it is illuminated by a y-polarized plane wave incident in the negative z-direction, with amplitude of 1 V/m. The wave frequency is exactly the resonant frequency of the lowest eigenmode. The field evaluation line is shown in Fig. 5.2.8 a, b.

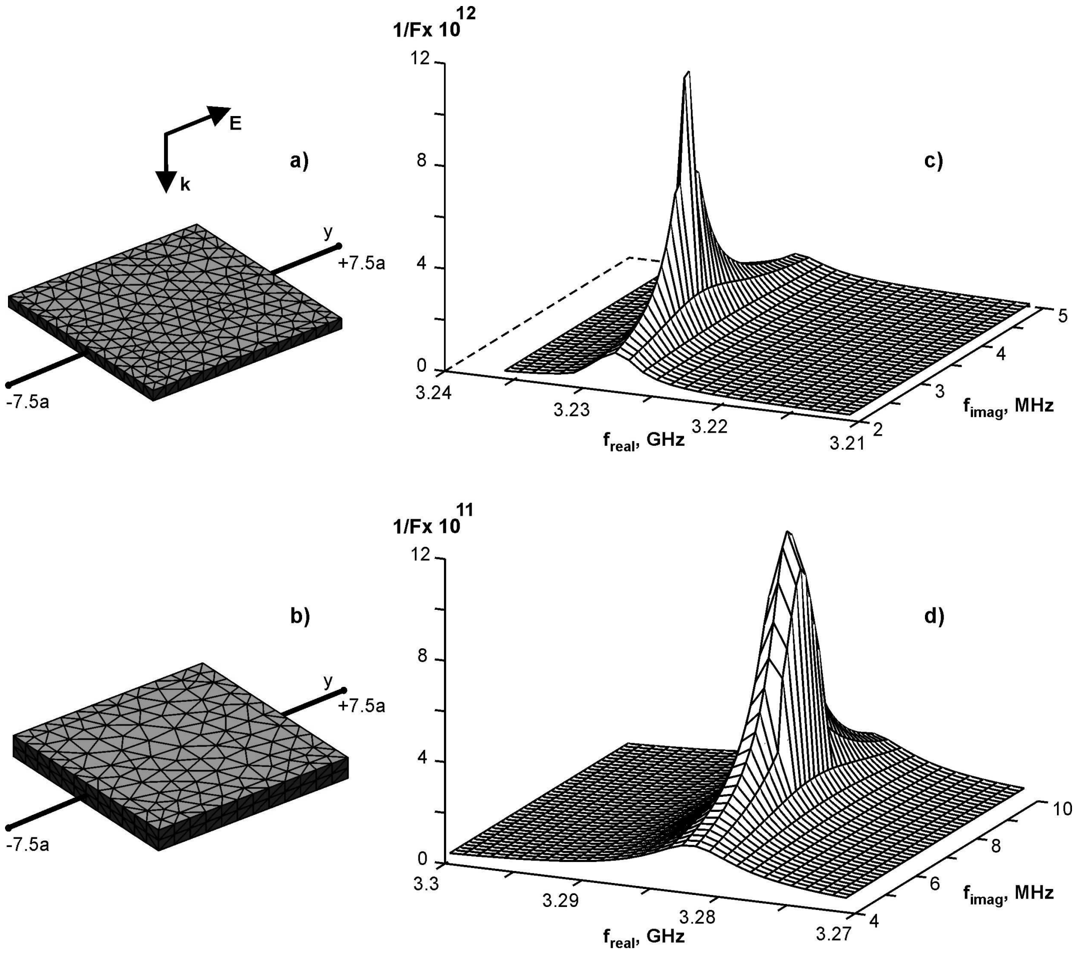


Fig. 5.2.8. a), b) Two parallel-plate resonator meshes based on the smallest eigenmode error in Fig. 5.2.7 a) and b), respectively; c), d) the related plots of the inverse of the cost function *F* over the plane of complex frequency.

As an example, the dominant *z*-component and the *y*-component of the scattered field are plotted in Fig. 5.2.9 for the 10:1 plate resonator along the *y*-axis over the length of 15*a*, which is 1.5 times the size of the resonator. The corresponding MoM mesh was shown in Fig. 5.2.8b. The MoM solution is given by the solid curve. The corresponding HFSS scattering solution obtained with 85,949 tetrahedra demonstrates good convergence and yields a final $\Delta E = 0.03\%$. This is shown by the dashed line. A good agreement is observed for the dominant *z*-component of the scattered field (Fig. 5.2.8a). However, the

HFSS solution for the co-polar scattered field indicates some oscillations close to the plate border (Fig. 5.2.8b). Within the resonator, both solutions are identical.

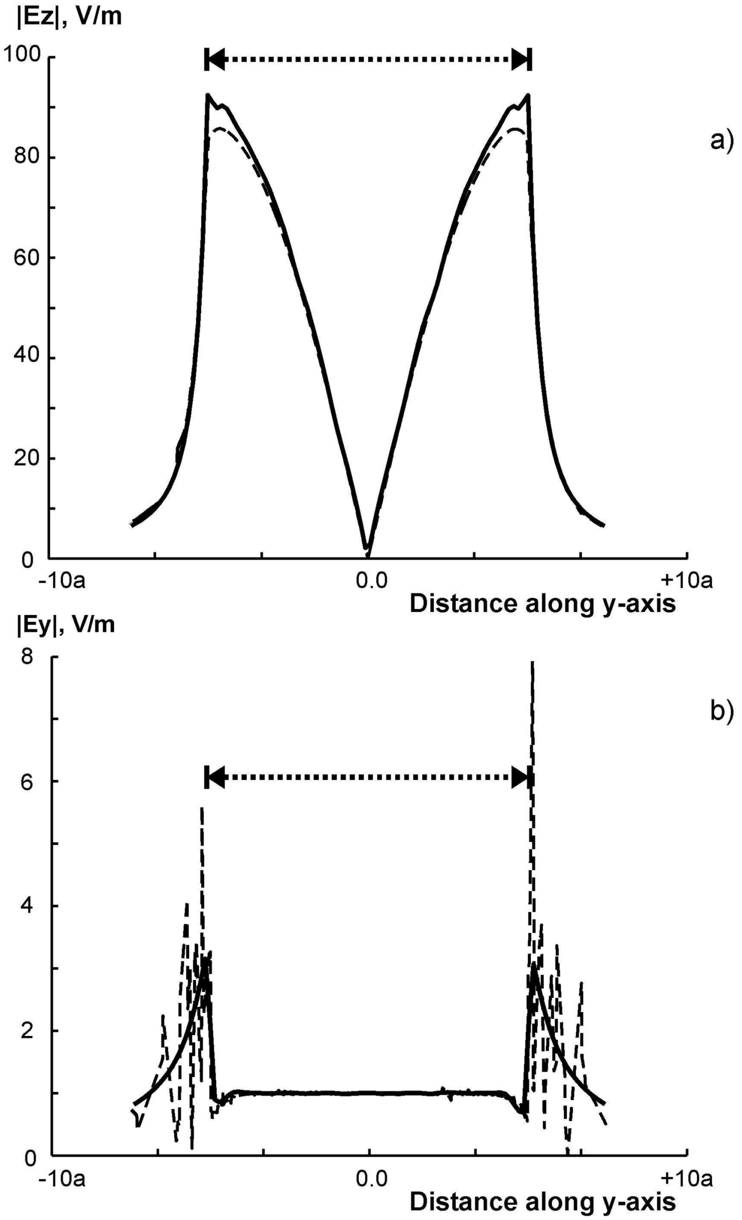


Fig. 5.2.8 a), b) Near scattered field of the *y*-polarized plane wave with amplitude 1 V/m at the resonant frequency of the 1:10 parallel-plate resonator. The field is given on the resonator axis. a) Magnitude of the dominant *z*-component of the electric field; b) magnitude of the co-polar component of the electric field. The MoM solution is shown by a solid line; the Ansoft HFSS solution is given by a dashed line. The plate size is indicated by a dotted line.

c. Conclusion

The method of volume integral equation coupled with the MoM zeroth-order edge-based basis functions gives a reliable solution for three basic isolated resonators with quality factors up to 500 and when the number of volume bases is reasonably small. Typically, 1000-2000 volume basis functions are necessary in order to obtain the quality factor with the accuracy on the order of 1-4%. The accuracy of the resonant frequency is considerably better.

Most interesting is the result for the thin (1:20) dielectric plate backed by two metal sheets. The simulation yields an accurate solution for the fundamental resonant mode (less than 1% error in the resonant frequency and a similar error in the quality factor) when the one–layer volume mesh is used. This observation supports the possible use of the edge-based functions for the MoM modeling of a patch antenna.

5.3 Test of simulations for a patch antenna

In this section, a metal patch antenna on a dielectric substrate is tested for different mesh configurations and different dielectric constants of the substrate. The simulations are based on the derivation given in sections 2.1, 2.2 and 2.3.

a. Thin patch antenna configuration

The structure under study is a thin patch antenna consisting of a single patch on a dielectric substrate with a finite ground plane. The dimensions of the patch are 30 mm x 40 mm on a 50 mm x 60 mm ground plane; with the thickness of the substrate 1.5 mm. Dielectric constant of the substrate was 2.55. A typical structure with 994 surface RWG elements and 2260 volume edge based basis functions is shown in Fig 5.3.1.

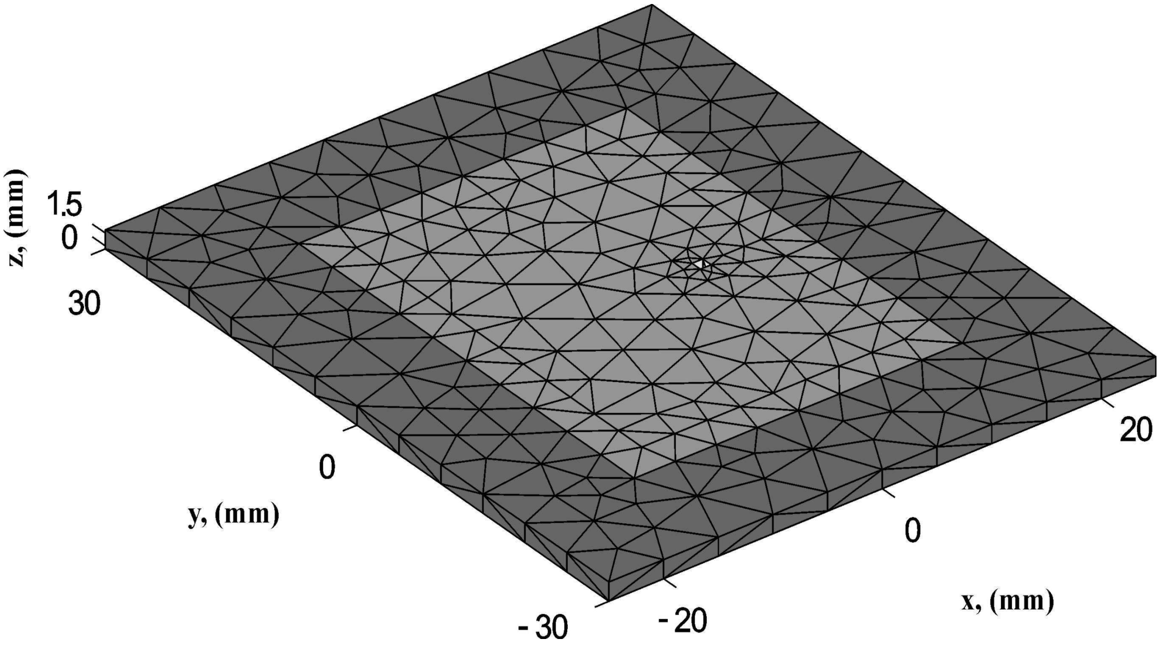


Fig. 5.3.1 Thin patch antenna structure with 994 surface RWG and 2260 volume edge-based basis functions.

The voltage gap feed model was used for the bottom edges of the feed to obtain all the results in this section. The variation of real and imaginary part of the impedance and the return loss are calculated over a suitable frequency range and plotted in Fig. 5.3.2a and Fig 5.3.2b respectively. The results were compared with the corresponding Ansoft HFSS v9 solution having about 102,000 tetrahedra.

The return loss is found based on the reflection coefficient, Γ, in the antenna feed verses the 50 Ω, termination namely

$$\Gamma = \frac{Z - 50\Omega}{Z + 50\Omega} \tag{5.3.1}$$

where Z is the complex input impedance. The return loss is the magnitude of the reflection coefficient in dB, i.e.

$$RL = 20\log_{10}|\Gamma| \tag{5.3.2}$$

The return loss (RL) is the most important parameter with respect to the load matching. It characterizes the antenna’s ability to radiate the power instead of reflecting it back to the generator. The antenna’s bandwidth is often defined as the band over which the return loss is sufficiently small (-10 dB and lower) [11].

Comparison results for the basic mesh configuration from Fig.5.3.1 can be seen in Fig.5.3.2.

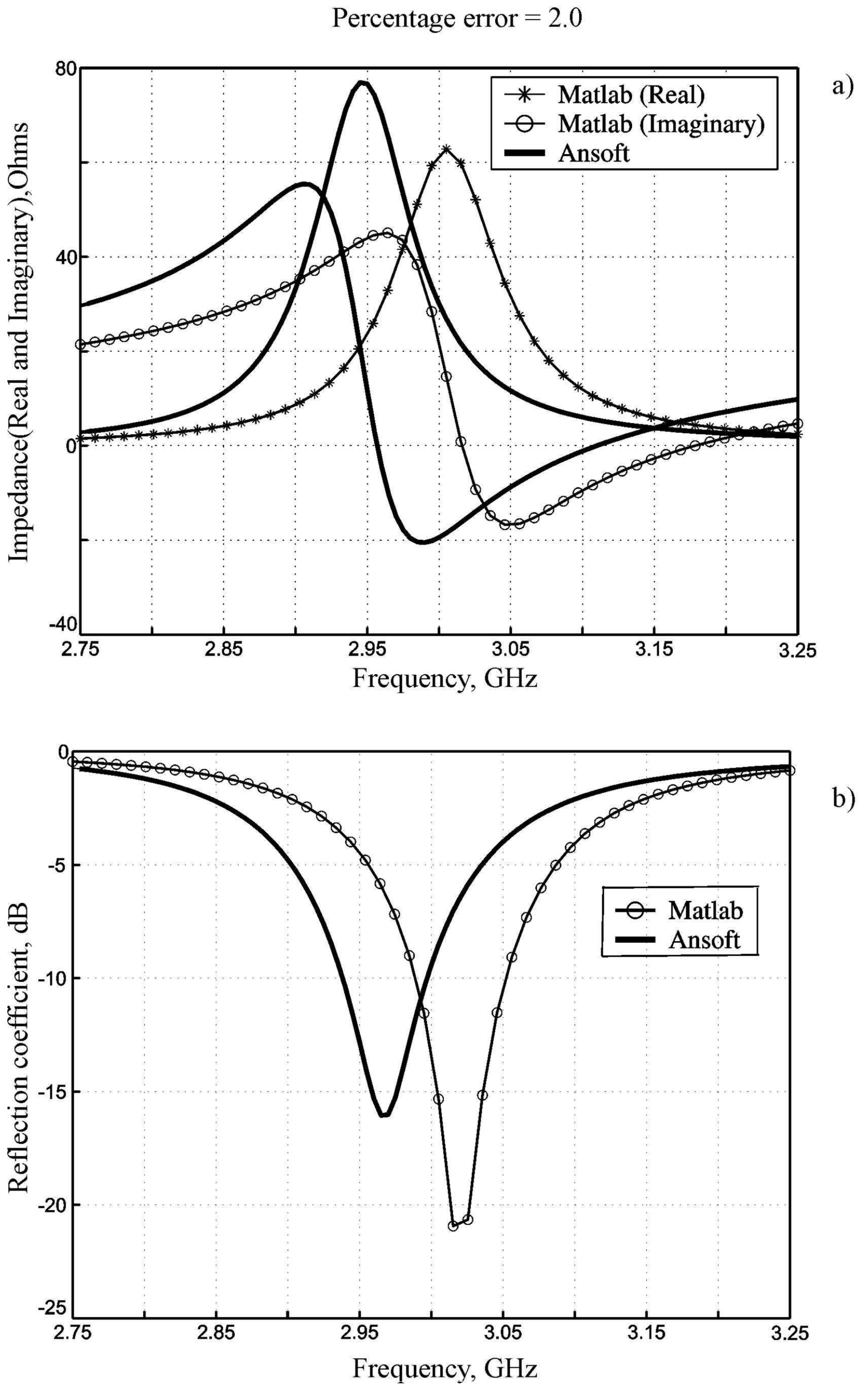


Fig. 5.3.2 Test result for the patch antenna in Fig 5.3.1 having $\varepsilon_r = 2.55$.
a) Impedance plot, b) return loss.

Tests were also performed for higher mesh discretization. Fig.5.3.2 shows one refined structure. It has two layers of tetrahedra and some rendering along the patch border. The structure has 1714 surface RWG elements and 6968 volume edge based basis functions. Fig.5.3.3 shows the result for this higher mesh refinement. Better agreement with HFSS was found for this refined structure in terms of both the amplitude and the resonant frequency. The error is observed to be 1.22 percent as compared to 2.0 percent as observed in Fig 5.3.1.

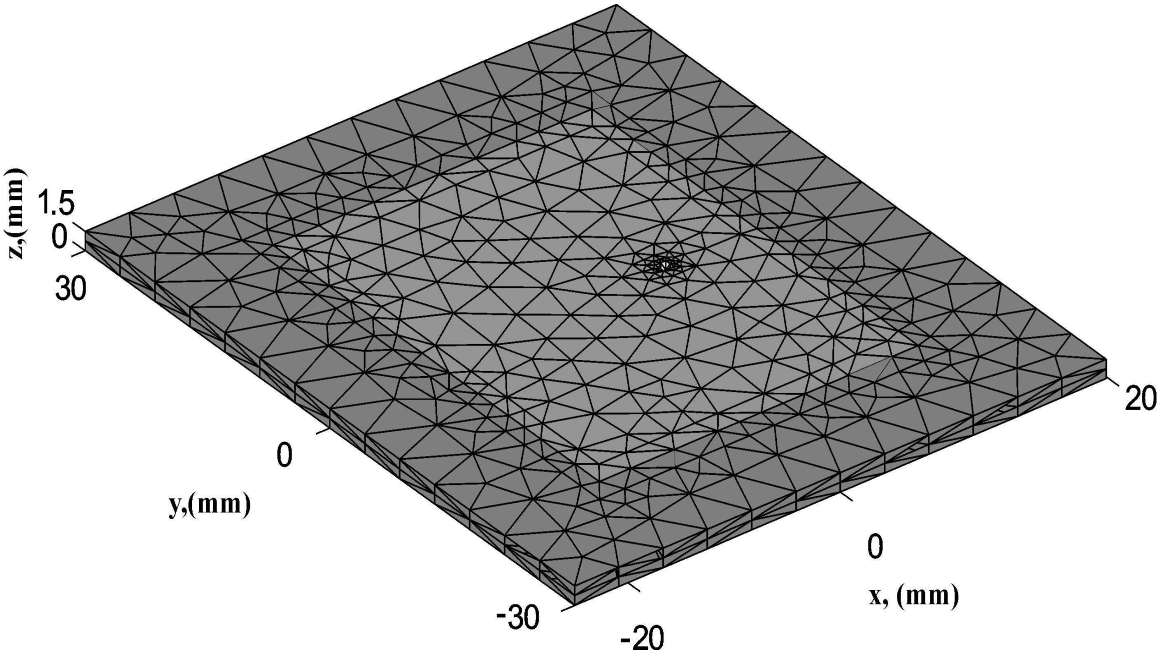


Fig.5.3.3 Refined patch antenna structure with 1714 surface RWG and 6968 volume edge-based basis functions.

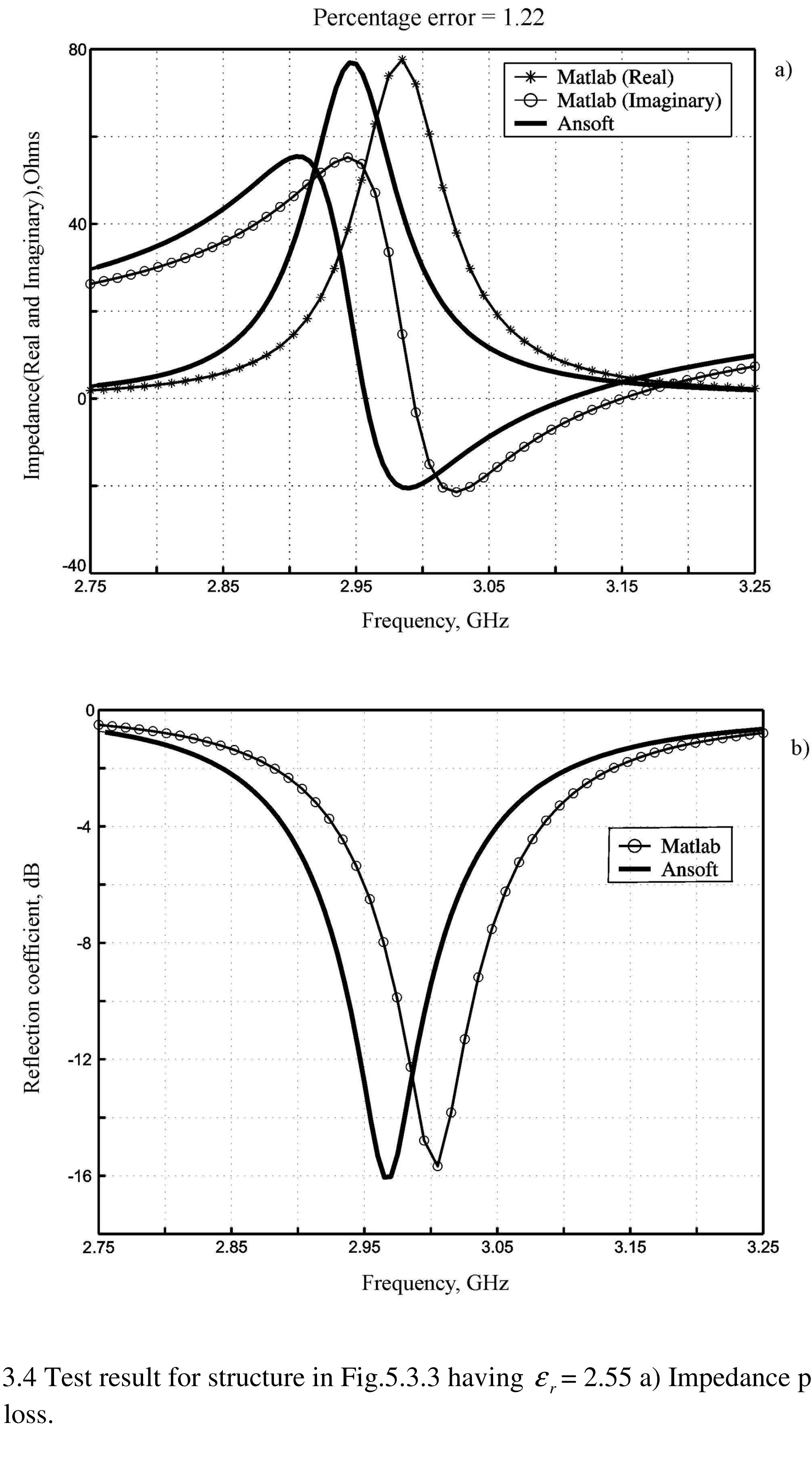

Percentage error = 1.22

Fig. 5.3.4 Test result for structure in Fig.5.3.3 having $\varepsilon_r = 2.55$ a) Impedance plot, b) return loss.

Table 5.3.1 provides the complete convergence history for this patch antenna structure. A steady improvement of the performance is observed when mesh is refined. The maximum possible mesh size was limited by the current solver performance.

Table 5.3.1 Convergence results for thin patch antenna structure having $\varepsilon_r = 2.55$.

Surface RWG basis functions	Volume edge basis functions	Calculated resonant frequency	Percentage error (vs. Ansoft HFSS)
498	1040	3.0284 GHz	2.41
510	1974	3.0221 GHz	2.20
994	2260	3.0162 GHz	2.00
1322	2622	3.0160 GHz	1.99
1006	3616	3.0033 GHz	1.56
1824	5844	2.9988 GHz	1.41
1848	6304	2.9971 GHz	1.35
1714	6968	2.9931 GHz	1.22

Next, we consider the convergence of the MoM solver as compared to the Ansoft HFSS solution at approximately the *same* number of tetrahedra. For the plot in Fig. 5.3.5 the Ansoft solution with 102,000 tetrahedra is considered as the exact solution and the convergence of Ansoft HFSS solution and the MoM solution is tested for the comparable number of tetrahedra. The convergence of the MoM solver (thin line) is observed to be better for smaller number of tetrahedra than the HFSS solution (thick line). However the HFSS solution converges nicely as the number of tetrahedra increase.

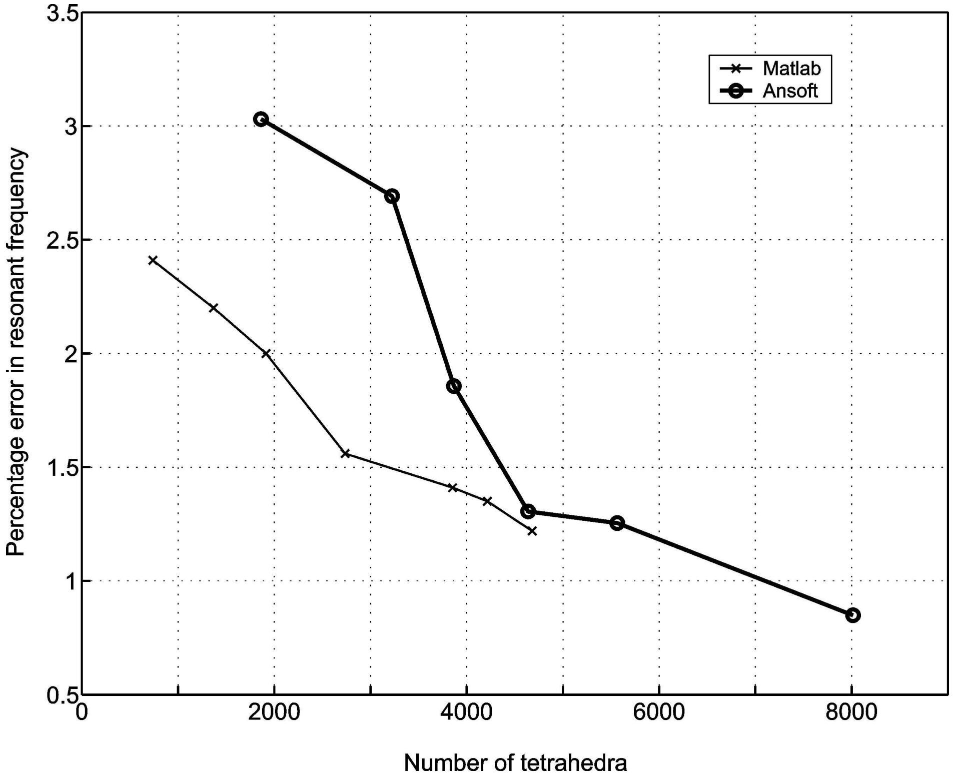


Fig. 5.3.5 Convergence of Ansoft HFSS solution (thick line) and MoM solution (thin line) for the comparable number of tetrahedra.

b. Thick patch antenna configuration

The structure under study is a patch antenna consisting of a single patch on a dielectric substrate with a finite ground plane. The dimensions of the patch are 2 x 4 cm on a 4 x 8 cm ground plane; with the thickness of the substrate 0.5 cm. A typical uniform mesh having a base grid size 8 by 16 (in *x, y* directions respectively,) is shown in Fig.5.3.6. This structure has 474 surface RWG elements and 1100 volume edge based elements.

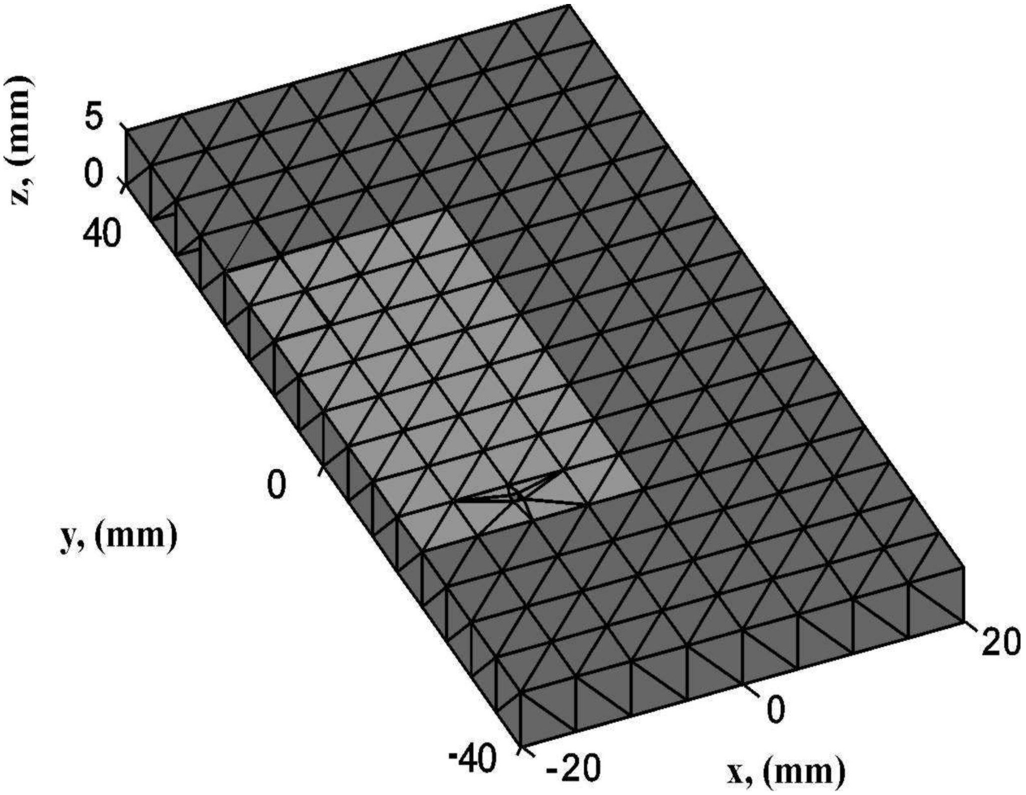


Fig. 5.3.6 Patch antenna structure with 474 surface RWG and 1100 volume edge basis functions.

The variation of real and imaginary part of the impedance over a suitable frequency range is calculated and plotted for different mesh configurations. The results are then compared with those obtained by using Ansoft HFSS v9. Different dielectric constants (2, 3, 5, and 10) for the substrate were considered. An impedance plot for $\varepsilon_r = 10$ is shown in Fig.5.3.7 for the mesh of Fig.5.3.6.

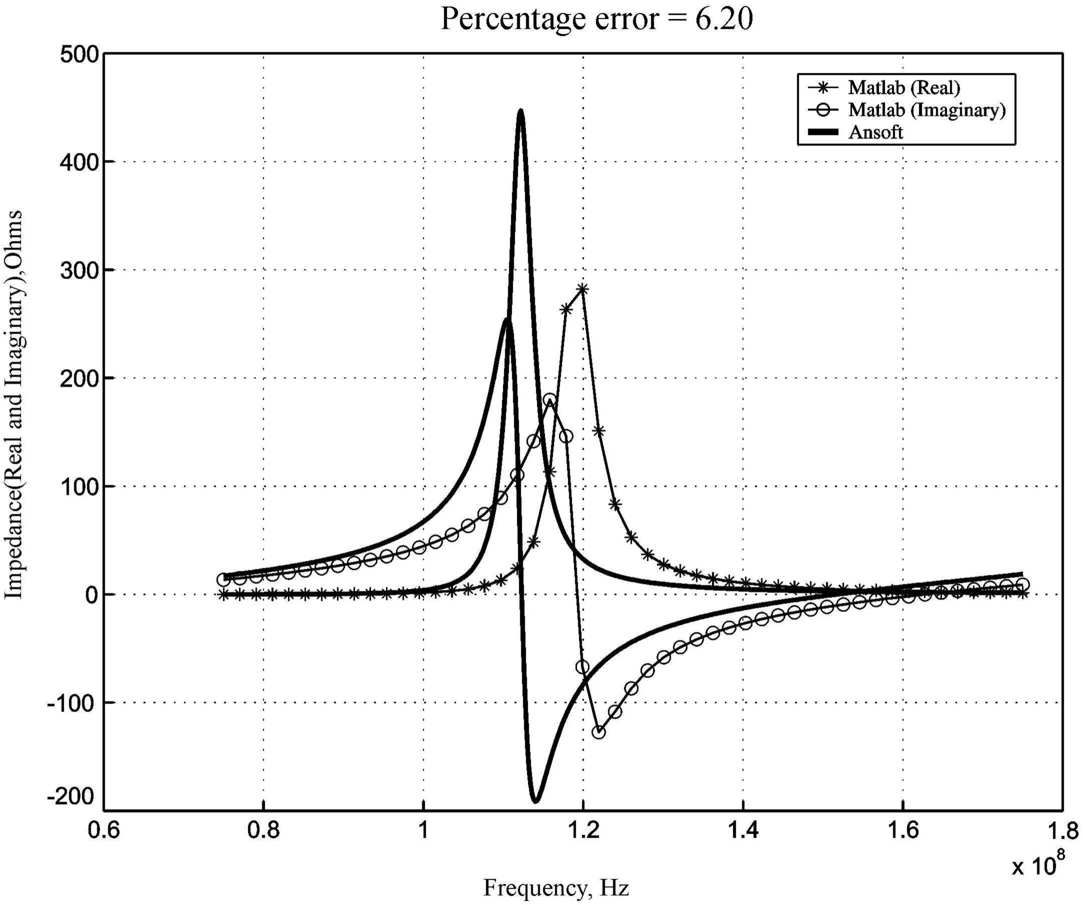


Fig. 5.3.6 Impedance plot for structure in Fig.5.3.5 having $\varepsilon_r = 10$.

Tests were also performed for higher mesh discretization and for different values of the dielectric constants of the substrate. Multiple layers of volume basis functions were introduced below the patch structure. Fig.5.3.7 shows the refined structure which has four layers of volume basis functions under the patch structure. A non uniform meshing is used in this case with patch border rendering. This structure has 1219 surface RWG elements and 5891 volume edge based elements.

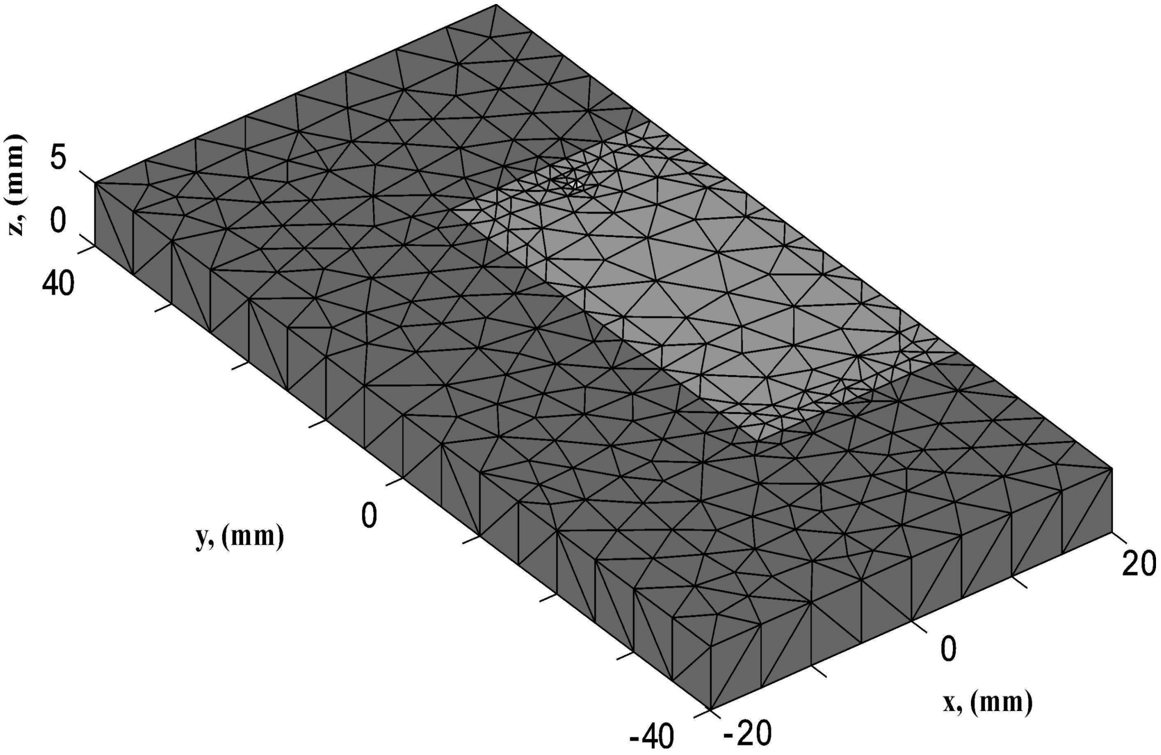


Fig. 5.3.7 Refined patch antenna structure with 1219 surface RWG elements and 5891 volume edge based elements.

Fig.5.3.8 shows the impedance plot for ε_r = 10 for the structure shown in Fig 5.3.7. Better agreement with ANSFOT is observed for this refined structure in terms of both the amplitude and the resonant frequency. The error in this case reduces to 2.16 percent as compared to 6.20 in the previous case.

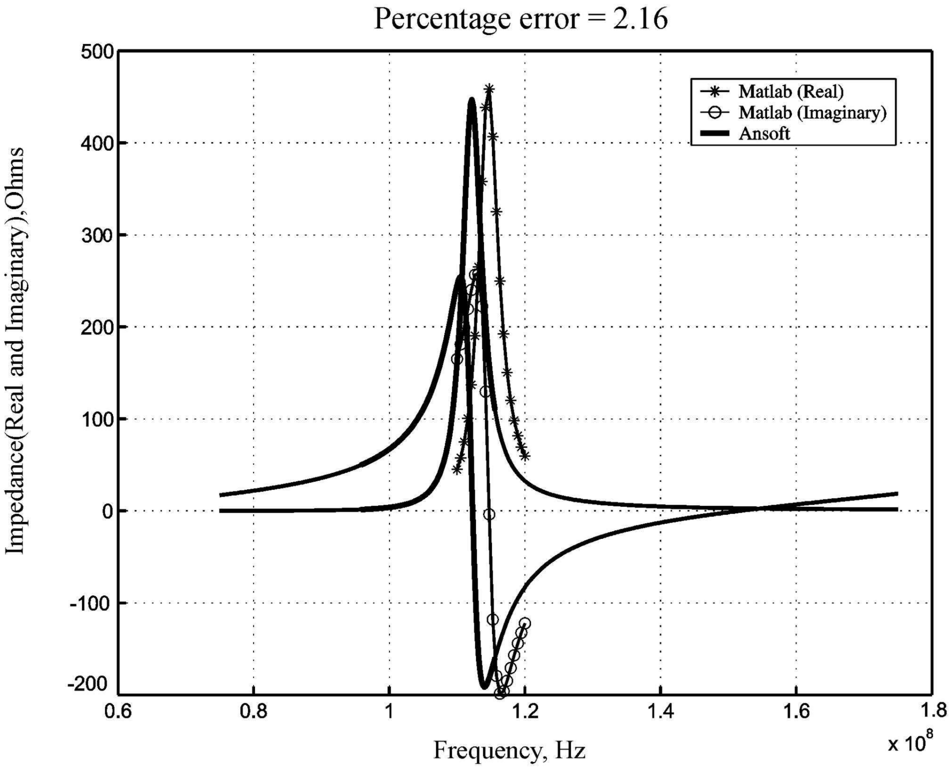


Fig. 5.3.8 Impedance plot for structure in Fig.5.3.7 having $\varepsilon_r = 10$.

Table 5.3.2 provides the complete convergence history for this structure. A steady improvement of the performance is observed with higher mesh refinement.

Table 5.3.2 Test results for the thick patch antenna.

ε_r	**Percent Error** RWG elements: 486 Edge elements: 1940	**Percent Error** RWG elements: 498 Edge elements: 2878	**Percent Error** RWG elements:1219 Edge elements: 5891
2	2.39	0.93	0.11
3	3.18	1.64	0.71
5	4.49	2.76	1.74
10	4.46	2.71	2.16

6. Analysis and Conclusion

Test results obtained in section 5 are discussed in this section. Possible methods to improve these results are proposed.

a. Non-resonant case

From the test results in Fig 5.1.2 for the homogenous dielectric plate and Fig 5.1.4 for the inhomogeneous dielectric cylinder it can be seen that, for the moderate size of the grid (around 500 edge elements), the error in the calculation of the near field is about 6 percent as compared to Ansoft HFSS. However a steady improvement in the performance is observed as we refine the mesh, with the lowest error of less than 0.5 percent achieved in case of the inhomogeneous cylinder with 2664 edge elements. The results of section 5.1 may be considered excellent. However, one needs to keep in mind that they are calculated at non-resonant frequencies.

The test results in section 5.2 with the metal- dielectric plate exhibit a similar behavior when mesh quality improves. A nearly perfect agreement for the vertical component of the scattered electric field is observed with the two layers of edge basis functions as shown in Fig. 5.2.4.

b. Resonant cases

The eigenmode solution for the solver is tested in section 5.2b for three different resonators. Good results were obtained in all the three cases (error less that 1% in the resonant frequency and a similar error in the quality factor) for a structure with around 2000 edge elements.

Keeping in mind these promising observations, we now take a look at the results for the patch antenna (radiation problem) in section 5.3. From Table 5.3.1, for a thin patch antenna configuration, it can be seen that for a structure with 1000 volume edge elements the error in the resonant frequency is 2.5%. With higher mesh discretization this error reduces to 1.22% (7000 volume edge elements) but a good agreement with Ansoft HFSS solution is not observed at this relatively large number of basis functions. Similarly, from Table 5.3.2, for a thicker patch antenna configuration, there is an improvement using higher mesh discretization but the error in the calculation of resonant frequency increases with increase in the dielectric constant of the substrate.

Thus, the following problem appears. While giving good results for some resonant/ non-resonant structures for a relatively small number of basis functions, the solver loses its performance for a typical patch antenna structure. The reason for such a behavior is not quite clear yet. Most probably it is connected to the non-adequate discretization of the edge singularity for the associated cavity resonator.

7. References

1. A. F. Peterson, S. L. Ray, and R. Mittra, *Computational Methods for Electromagnetics*, IEEE Press, Piscataway, New Jersey, 1998.

2. D. H. Schaubert, D. R. Wilton, and A. W. Glisson, "A tetrahedral modeling method for electromagnetic scattering by arbitrarily shaped inhomogeneous dielectric bodies," *IEEE Trans. Antennas and Propagation*, vol. AP-32, no. 1, pp. 77-85, January 1984.

3. Y. Liu, S. Safavi-Naeini, and S. K. Chaudhuri, "Comparison of SIE-MoM and VIE-MoM for determination of complex resonant frequency of dielectric resonators", in Antennas and Propagation Society, 2003 IEEE International Symposium, vol. 1, pp. 641 - 644, June 2003.

4. T. Sarkar, S. M. Rao, and A. R. Djordjević, "Electromagnetic scattering and radiation from finite microstrip structures", *IEEE Trans. Microwave Theory and Techniques*, vol. 38, pp. 1568-1575, Nov. 1990.

5. N. A. Ozdemir and J. F. Lee, "Single level dual rank SVD algorithm for volume integral equations of electromagnetic scattering", in Antennas and Propagation Society, 2003 IEEE International Symposium, vol. 2, pp. 302 - 305, June 2003.

6. M. L. Barton and Z. J. Cendes, "New vector finite elements for three dimensional magnetic field computation", *J. Appl. Phys.,* vol. 61, No 8, pp. 3919-3921, 1987.

7. A. Bossavit, “Whitney forms: a class of finite elements for three dimensional computations in electromagnetism”, *IEE Proceedings*, vol. 135, Pt. A, No 8, pp. 493-499, Nov. 1988.

8. Jin Fa Lee and R. Mittra, “A note on the application of edge elements for modeling three-dimensional inhomogeneously-filled cavities,” *IEEE Trans. Microwave Theory and Techniques*, vol. MTT-40, pp. 1767-1773, Sep. 1992.

9. J. B. Manges and Z. J. Cendes, “A generalized tree-cotree gauge for magnetic field computation,” *IEEE Trans. on Magnetics*, vol. 31, pp. 1342-1347, May 1995.

10. I. Tičar, O. Bíró, and K. Preis, “Vector potential expanded by edge basis functions associated with loops on finite-element facets,” *IEEE Trans. on Magnetics*, vol. 38, pp. 437-440, March 2002.

11. C. A. Balanis, *Antenna Theory: Analysis and Design*, Wiley & Sons Inc, 2nd edition, 1982.

12. L. Gurel, I. K. Sendur, and K. Sertel, “Quantitative Comparison of Rooftop and RWG Basis Functions,” *1997 IEEE AP-S International Symposium and URSI Radio Science Meeting*, Montreal, Canada, vol. 2, pp. 796-799, July 1997.

13. S. M. Rao, D. R. Wilton, and A. W. Glisson, “Electromagnetic scattering by surfaces of arbitrary shape,” *IEEE Trans. Antennas and Propagation*, vol. AP-30, no. 3, pp. 409-418, May 1982.

14. S. A. de Carvalho and L. de Souza Mendes, “Scattering of EM waves by inhomogeneous dielectrics with the use of the method of moments and the 3-D

solenoidal basis functions", *Microwave and Optical Technology Letters,* vol. 23, No. 1, pp. 42-46, Oct. 1999.

15. E. Anderson, z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorenson, " LAPACK User's Guide," 3rd ed. (SIAM 1999).

16. O.C. Zienkiewicz and R. L. Taylor, *The Finite Element Method*, v. *1 The Basics*, Butterworth-Heinemann, Oxford, 2000, 5th edition, p. 222.

17. G. R. Cowper, "Gaussian quadrature formulas for triangles," *In. J. Numer. Meth. Eng.*, pp. 405-408, 1973.

18. T. F. Eibert and V. Hansen, "On the calculation of potential integrals for linear source distributions on triangular domains," *IEEE Trans. Antennas and Propagation*, vol. AP-43, no. 12, pp. 1499-1502, Dec. 1995.

19. D. R. Wilton ,S. M. Rao, A. W. Glisson, D.H Schaubert, O.M. Al-Bundak, and C.M Butler, " Potential integrals for uniform and linear source distribution on polygon and polyhedral domains," *IEEE Trans. Antennas and Propagation*, vol. AP-32, no. 3, pp. 276-281, May 1984.

20. R. Cools, "An Encyclopedia of Cubature Formulas", *J. Complexity*, 19: 445-453, 2003

21. R. Cools, "Monomial cubature rules since ``Stroud": a compilation - part 2", *J. Comput. Appl. Math.*, 112(1-2): 21--27, 1999.

22. R. Cools, “Monomial cubature rules since ``Stroud": a compilation ”, *J. Comput. Appl. Math.*, 112(1-2): 21--27, 1999.

23. G. P. Junker, A. A. Kishk, and A. W. Glisson, “A novel delta gap source model for center fed cylindrical dipoles,” *IEEE Trans. Antennas Propagation*, vol. 43, pp. 537-540, May 1995.

24. B.D Popović, M.B Dragović and A.R. Djordjević, *Analysis and Synthesis of Wire Antenna*, John Wiley & Sons Inc, 1982.

25. See web link below for Matlab external interface/ API references. http://www.mathworks.com/access/helpdesk/help/techdoc/apiref/ch00intr.html.

Appendix A. Preliminary comparison of the MoM solution with the Mie series for the dielectric sphere

1. Problem statement

In order to test the accuracy of the solutions obtained by using the edge-based basis functions, it is desired to apply the method to a problem for which an analytical solution is available. Unfortunately, very few three-dimensional dielectric scattering problems can be solved analytically. The homogenous dielectric sphere and the layered, inhomogeneous sphere are the two cases where the scattered fields can be analytically determined [1].

Since the dielectric sphere with, possibly, axial inhomogeneity is an example where the scattered field can be analytically determined, this example is chosen below. The internal electric field distribution within the sphere is determined. The scattering cross section and the scattering radiation pattern of a dielectric sphere can be then calculated analytically from the Mie series expansion for the internal fields if necessary [2,3].

The present example is a homogeneous sphere of radius R=*2 cm.* An *x*-polarized plane wave is incident on it along the positive *z*-direction. The scattered electric field in the dielectric has the dominant *x*-component and is evaluated on a line x = *0 cm,* y = *0 cm* and at $-1.25R < z < 1.25R$. The results of the MoM solver are compared with Ansoft HFSS solution having from 78,000 to 88,000 tetrahedra and with the Mie series expansion.

The present Mie series solution calculates the scattered or total field inside the sphere only, while the Ansoft and MoM solutions calculate the fields outside the sphere as well. The Mie solution is based on the vector potential approach of Stratton [2] and was programmed by Mr. Rostislav Lemdiasov a research assistant and PhD student in the ECE department at WPI. Note that an alternative formulation of the Mie series exists [3].

2. MoM simulation parameters

Three different cases corresponding to three different values of the dielectric constant of the sphere are considered. Also, three different mesh configurations are considered having 302, 2348 and 4322 tetrahedra (Fig. A1.a) respectivley. The variation of the scattered field is obtained at or near the lowest resonant frequency of the sphere. The resonant frequency is computed from the characteristic equation of the TE_{111} mode, for the source free system [4]. Table A1 below gives the simulation data for the different test cases.

Table. A1. Simulation data for the homogenous sphere.

Dielectric constant	k_0a		
2	1.6	2.0 $\approx f_{res}$	2.4
10	0.8	1.0 $\approx f_{res}$	1.2
50	0.4	0.5 $\approx f_{res}$	0.6

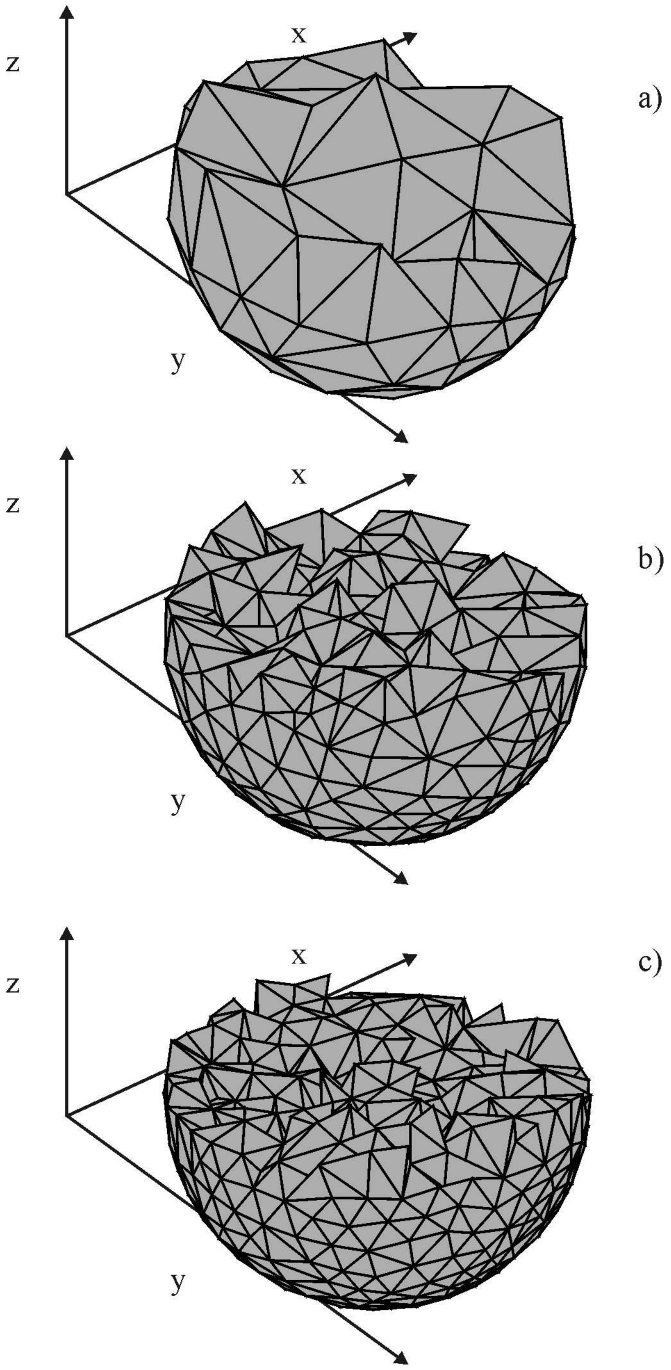


Fig. A1. a) – Tetrahedral meshes with a) 302 tetrahedra; b) 2348 tetrahedra; c) 4322 tetrahedra.

3. Scattered field

The scattered electric field in the MoM method is obtained by using the electric and magnetic potentials in the form (see section 3.2)

$$\begin{aligned}\vec{E}^{s} = &\frac{\omega^{2}\mu_{0}}{4\pi}\int_{V}\sum_{n=1}^{N}\left\{\sum_{p=1}^{P}K_{p'}\vec{f}_{np'}(\vec{r}\,')g(\vec{r},\vec{r}\,')d\vec{r}\,'\right\}D_{n} \\ &-\frac{1}{4\pi\varepsilon_{0}}\int_{S}\sum_{n=1}^{N}\left\{\sum_{q=1}^{Q}\hat{K}_{q'}f_{\perp nq'}(\vec{r}\,')\left(-jkg(\vec{r},\vec{r}\,')-\frac{g(\vec{r},\vec{r}\,')}{|\vec{r}-\vec{r}\,'|}\right)\nabla_{r}|\vec{r}-\vec{r}\,'|d\vec{r}\,'\right\}D_{n}\end{aligned} \tag{A1}$$

where D_n are the already obtained MoM solution for the total electric flux density in the dielectric – see section 3.2. Equation (A1) is valid everywhere in space, not only within the sphere. Unfortunately, it may have a large error when the observation point is close to the dielectric boundary, due to integration singularity (gradient of the free-space Green's function).

Alternatively, within the sphere, one can also find the total electric flux by

$$\vec{D} = \sum_{n=1}^{N} D_{n}\vec{f}_{n} \tag{A2}$$

and next find the scattered field by subtracting the incident signal. Equation (A2) uses the MoM solution directly. Since the basis functions $\vec{f}_n$ are piecewise constants, it leads to a "staircase" solution. Below, we will present the results corresponding to both approaches (A1) and (A2).

4. Results for a low value of the dielectric constant

Fig. A.2 gives the dominant component of the scattered field $\left|E_x^s\right|$ (found according to (A1)) within the dielectric sphere calculated along the z-axis compared to the Ansoft HFSS solution (dashed curve) and the Mie solution (thick curve) at $k_0a = 1.6$ and $\varepsilon_r = 2$. Three groups of results are related to three different MoM meshes a) 302 tetrahedra; b) 2348 tetrahedra; c) 4322 tetrahedra. Similar results for the same $\varepsilon_r = 2$ are shown in Fig A.3 for $k_0a = 2$ and in Fig. A.4 for $k_0a = 2.4$.

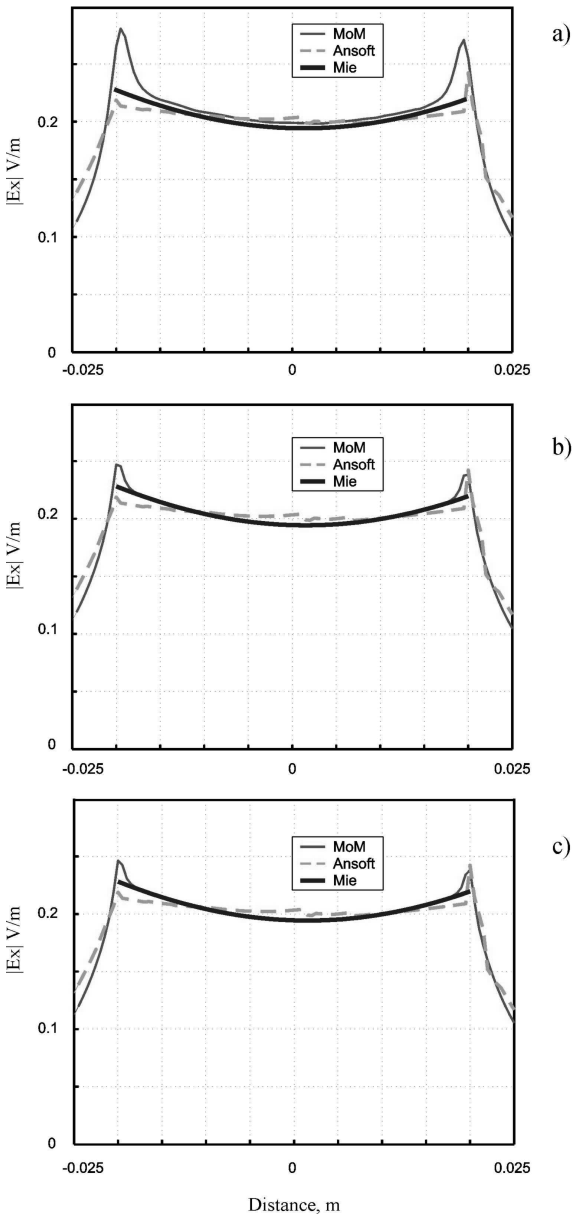


Fig. A.2 Dominant component of the scattered field $\left|E_x^s\right|$ within the dielectric sphere along the *z*-axis compared to Ansoft solution (dashed curve) and the Mie solution (thick curve) at $k_0 a = 1.6$ and $\varepsilon_r = 2$ (by using equation (A1)). Three groups of results are related to three different meshes a) 302 tetrahedra; b) 2348 tetrahedra; c) 4322 tetrahedra. The Ansoft HFSS solution has 79321 tetrahedra.

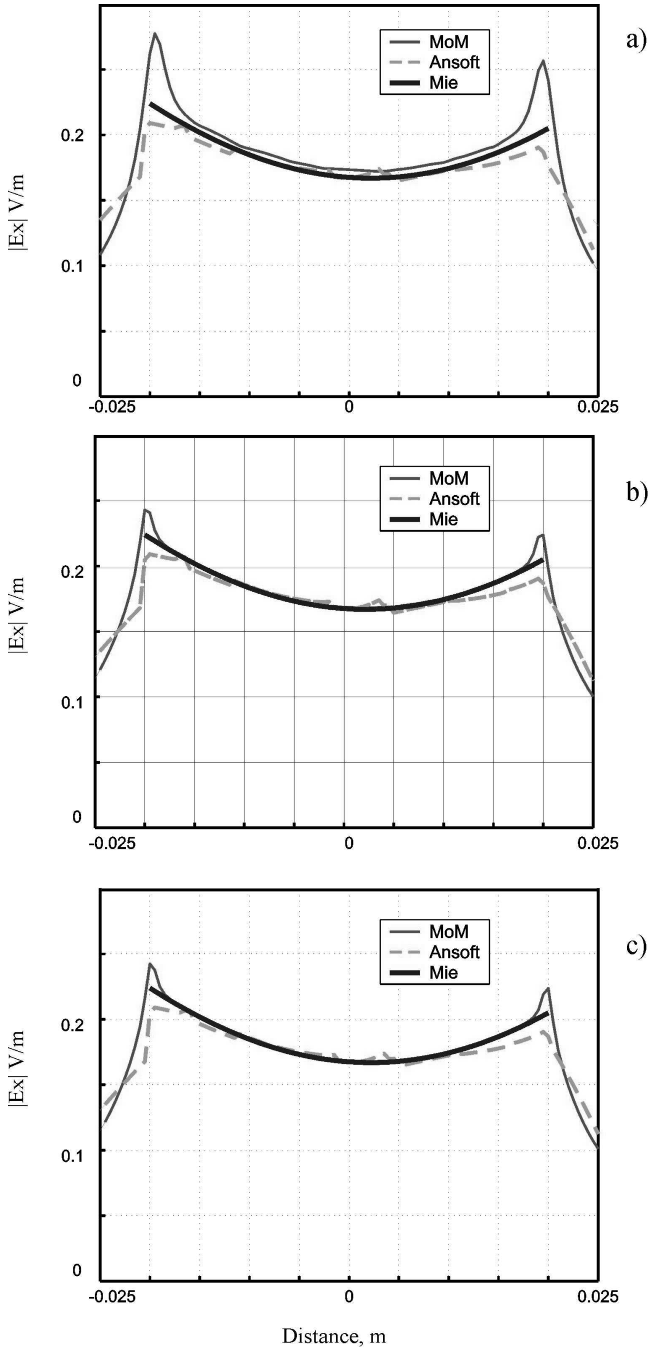


Fig. A.3 Dominant component of the scattered field $\left|E_x^s\right|$ within the dielectric sphere along the z-axis compared to Ansoft solution (dashed curve) and the Mie solution (thick curve) at $k_0 a = 2.0$ and $\varepsilon_r = 2$ (by using equation (A1)). Three groups of results are related to three different meshes a) 302 tetrahedra; b) 2348 tetrahedra; c) 4322 tetrahedra. The Ansoft HFSS solution has 82961 tetrahedra.

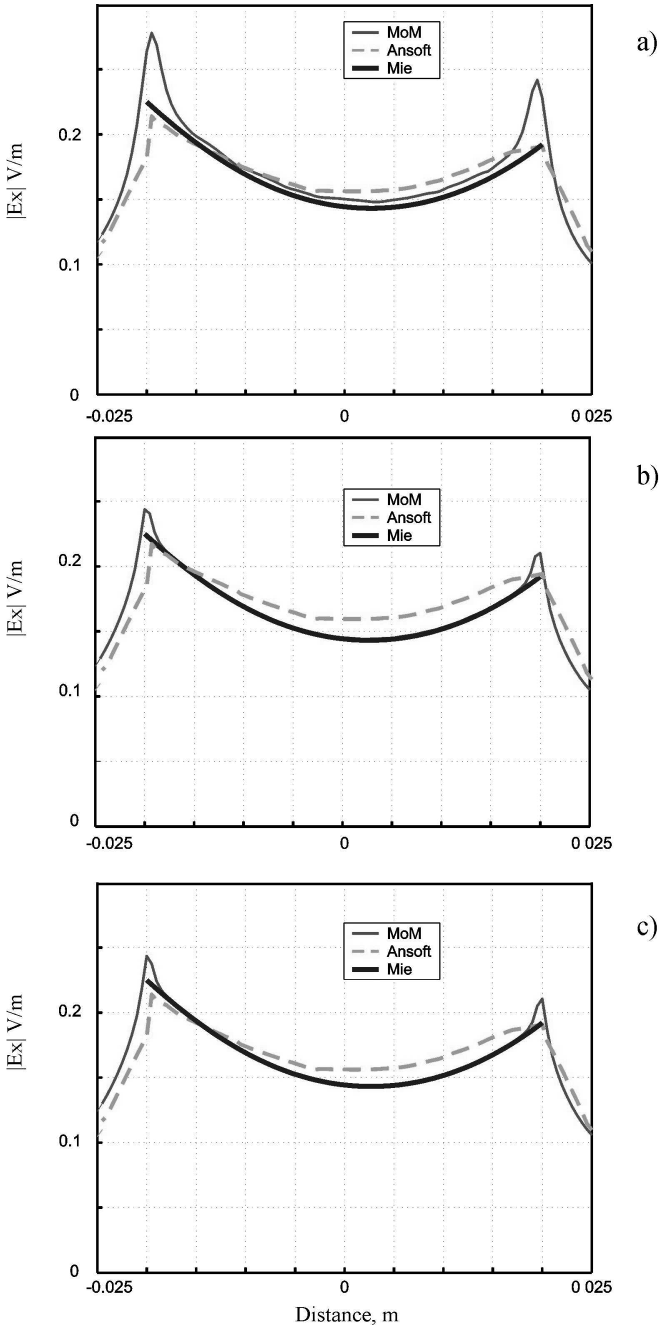


Fig. A.4 Dominant component of the scattered field $\left|E_x^s\right|$ within the dielectric sphere along the *z*-axis compared to Ansoft solution (dashed curve) and the Mie solution (thick curve) at $k_0 a = 2.4$ and $\varepsilon_r = 2$ (by using equation (A1)). Three groups of results are related to three different meshes a) 302 tetrahedra; b) 2348 tetrahedra; c) 4322 tetrahedra. The Ansoft HFSS solution has 86421 tetrahedra.

One can see the MoM results agree fairly well with the analytical solutions except two small “ears” that correspond to the points in the vicinity of the sphere boundary. It is believed that the ears are due to a numerical integration effect and must disappear once a more accurate formula for the potential integrals (with singularity extraction and analytical calculation of the self-integral) is used for the gradient of the free-space Green’s function. This work is currently underway.

To support this observation we present below in Figs. A.5-A.7 the same results, but obtained according to equation (A2). Figs. A.5-A.7 display a uniform convergence along the sphere diameter despite the non-continuous character of the solution.

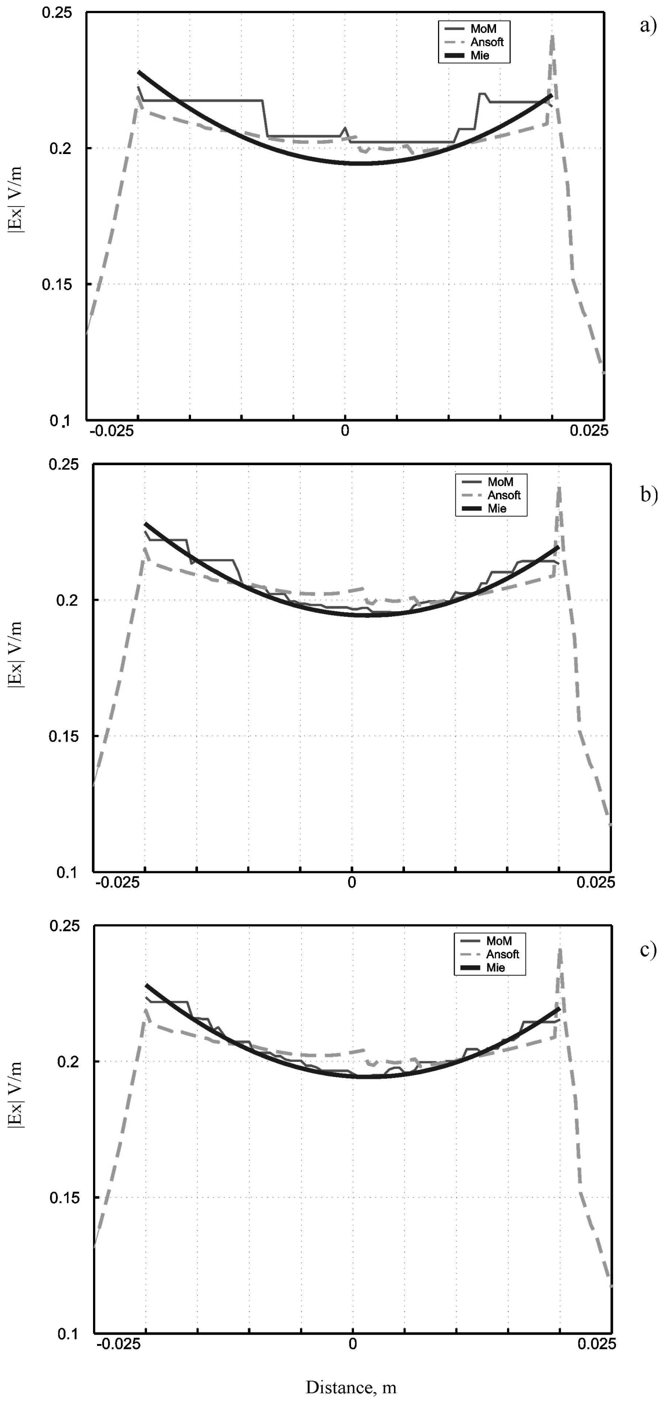


Fig. A.5 Dominant component of the scattered field $\left|E_x^s\right|$ within the dielectric sphere along the *z*-axis compared to Ansoft solution (dashed curve) and the Mie solution (thick curve) at $k_0 a = 1.6$ and $\varepsilon_r = 2$ (by using the direct approach equation (A2)). Three groups of results are related to three different meshes a) 302 tetrahedra; b) 2348 tetrahedra; c) 4322 tetrahedra. The Ansoft HFSS solution has 79321 tetrahedra.

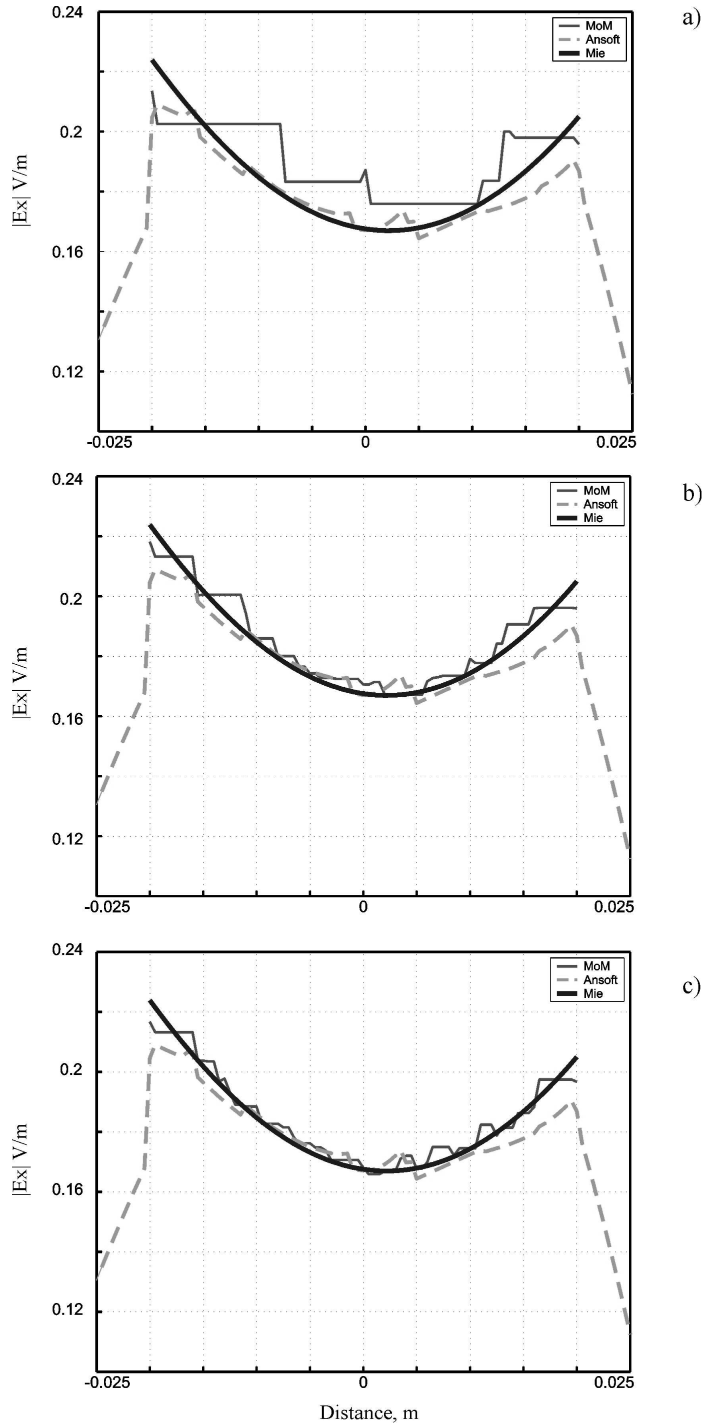


Fig. A.6 Dominant component of the scattered field $\left|E_x^s\right|$ within the dielectric sphere along the *z*-axis compared to Ansoft solution (dashed curve) and the Mie solution (thick curve) at $k_0 a = 2.0$ and $\varepsilon_r = 2$ (by using the direct approach equation (A2)). Three groups of results are related to three different meshes a) 302 tetrahedra; b) 2348 tetrahedra; c) 4322 tetrahedra. The Ansoft HFSS solution has 82961 tetrahedra.

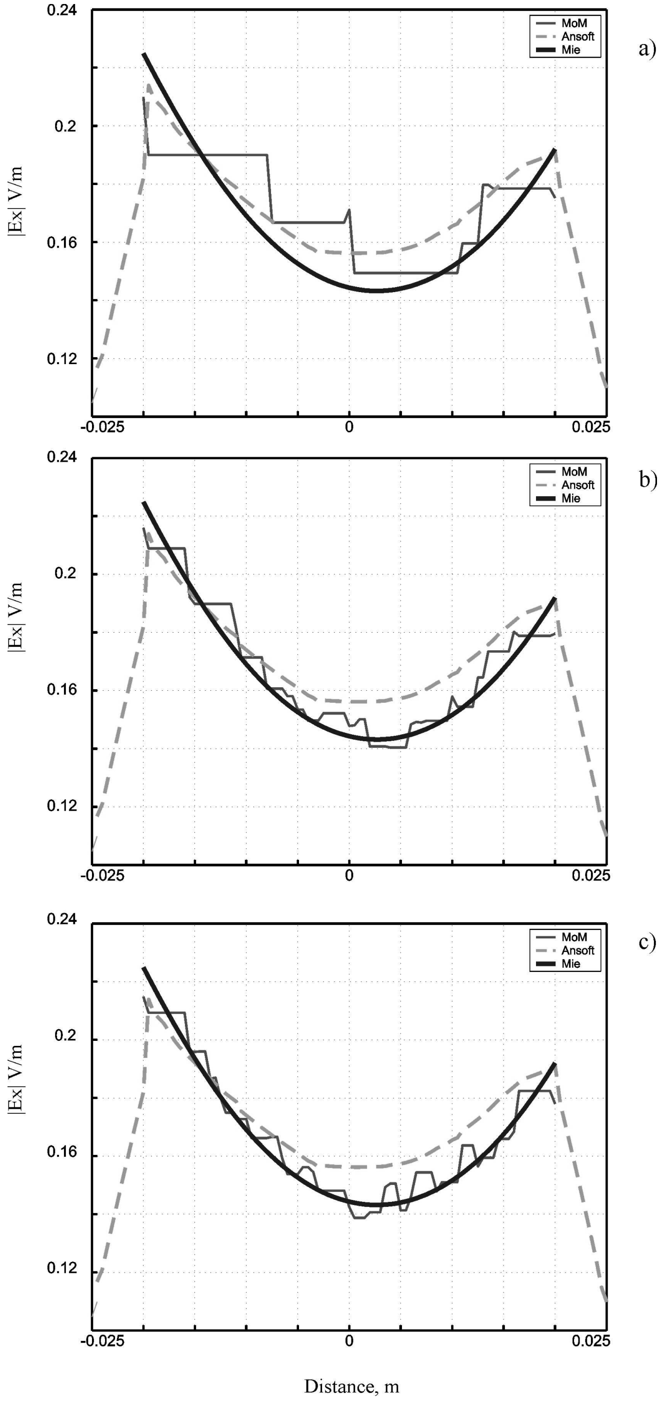


Fig. A.7 Dominant component of the scattered field $\left|E_x^s\right|$ within the dielectric sphere along the z-axis compared to Ansoft solution (dashed curve) and the Mie solution (thick curve) at $k_0a = 2.4$ and $\varepsilon_r = 2$ (by using the direct approach equation (A2)). Three groups of results are related to three different meshes a) 302 tetrahedra; b) 2348 tetrahedra; c) 4322 tetrahedra. The Ansoft HFSS solution has 86421 tetrahedra.

5. Results for a higher value of the dielectric constant

Similar results were also obtained for a homogenous sphere with higher dielectric constants. Fig. A.8 gives the dominant component of the scattered field $\left|E_x^s\right|$ (found according to (A1)) within the dielectric sphere with $\varepsilon_r = 10$ and $k_0 a = 0.8$. Three groups of results are related to three different meshes a) 302 tetrahedra; b) 2348 tetrahedra; c) 4322 tetrahedra. The results obtained in this case are compared to the Mie solution only. Similar results for the same $\varepsilon_r = 10$ are shown in Fig. A.9 for $k_0 a$ = 1.0 and in Fig.A.10 for $k_0 a$ = 1.2.

Fig. A.11 gives the dominant component of the scattered field $\left|E_x^s\right|$ (found according to (A1)) within the dielectric sphere with $\varepsilon_r = 50$ and $k_0 a = 0.4$. Three groups of results are related to three different meshes a) 302 tetrahedra; b) 2348 tetrahedra; c) 4322 tetrahedra. The results obtained in this case are compared to the Mie solution only. Similar results for the same $\varepsilon_r = 50$ are shown in Fig A.12 for $k_0 a$ = 1.0 and in Fig. A.13 for $k_0 a$ = 1.2.

In all the above cases results similar to that observed at lower dielectric constants are obtained.

6. Conclusion

The method of volume integral equation coupled with the MoM zeroth-order edge-based basis functions gives a reliable solution for the scattered field distribution inside the spheres having different dielectric constants (upto 50). A good convergence is obtained when mesh quality improves. The electric field is calculated by using two approaches:

(i) from the equation of the scattered field (equation A1), (ii) directly using the MoM solution for the electric flux density (equation A2). The MoM results agree fairly well with the analytical solutions in both the cases. However, for the first approach, we observe two small "ears" that correspond to the points in the vicinity of the sphere boundary. It is believed that the ears are due to a numerical integration effect and must disappear once a more accurate formula for the potential integrals (with singularity extraction and analytical calculation of the self-integrals) is used.

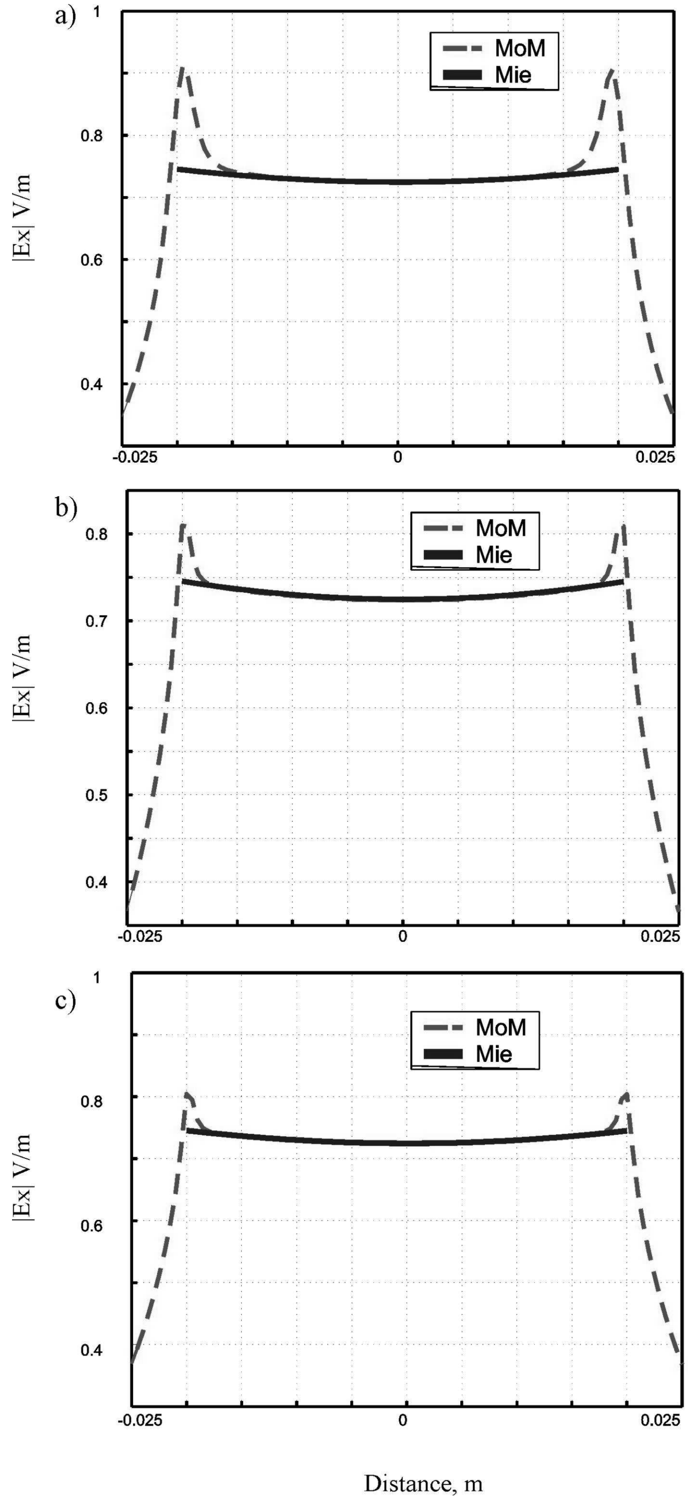


Fig. A.8 Dominant component of the scattered field $\left|E_x^s\right|$ within the dielectric sphere along the *z*-axis compared to the Mie solution (thick curve) at $k_0 a = 0.8$ and $\varepsilon_r = 10$ (by using equation (A1)). Three groups of results are related to three different meshes a) 302 tetrahedra; b) 2348 tetrahedra; c) 4322 tetrahedra.

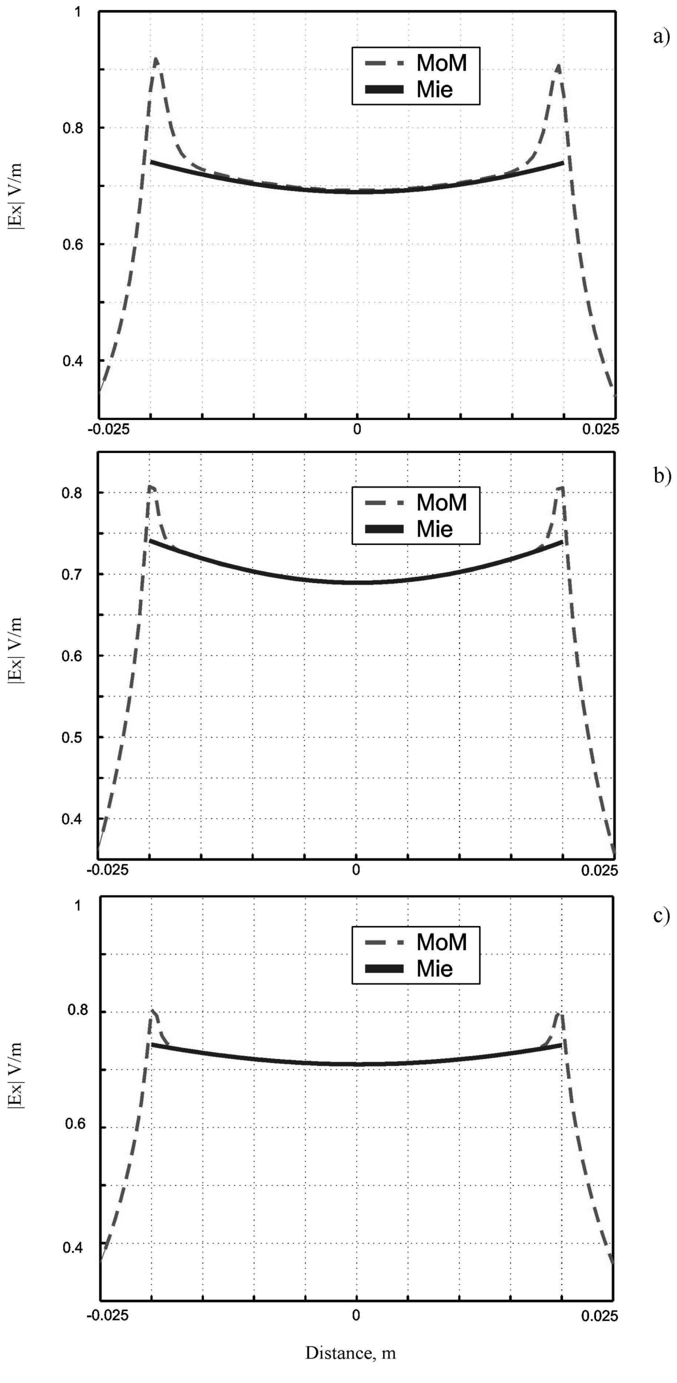


Fig. A.9 Dominant component of the scattered field $\left|E_x^s\right|$ within the dielectric sphere along the *z*-axis compared to the Mie solution (thick curve) at $k_0a = 1.0$ and $\varepsilon_r = 10$ (by using equation (A1)). Three groups of results are related to three different meshes a) 302 tetrahedra; b) 2348 tetrahedra; c) 4322 tetrahedra.

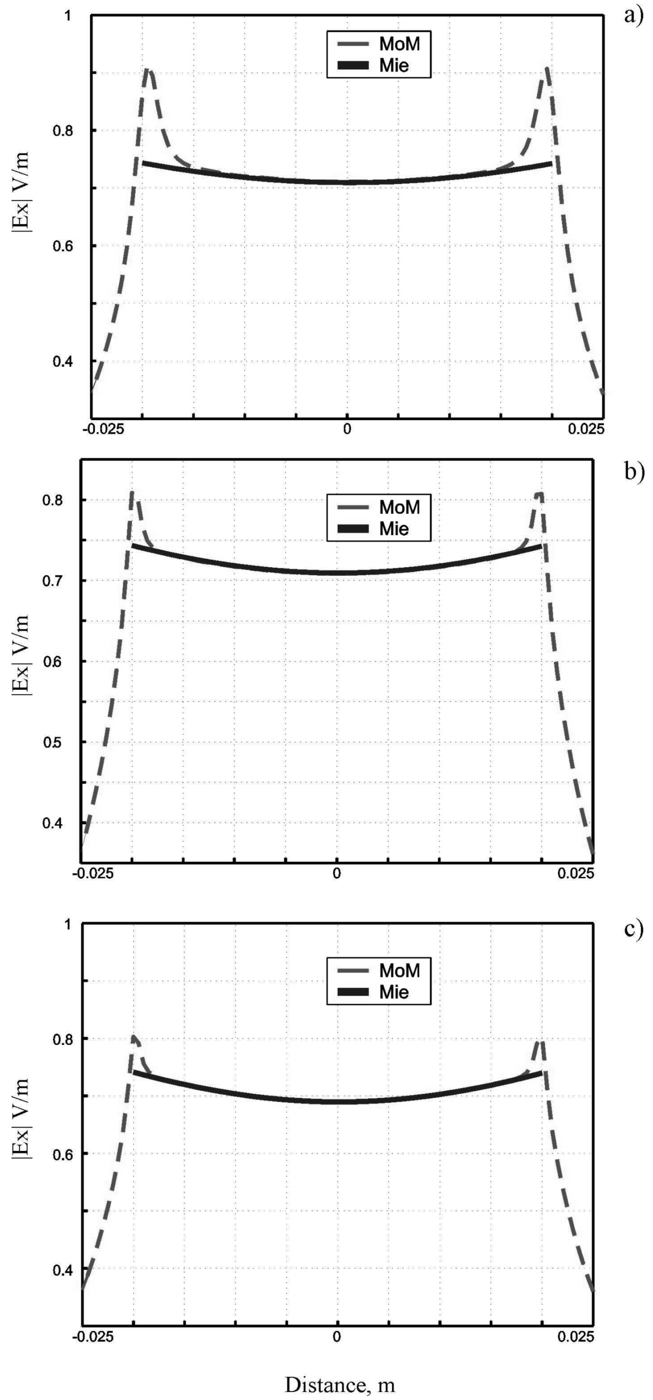


Fig. A.10 Dominant component of the scattered field $\left|E_x^s\right|$ within the dielectric sphere along the *z*-axis compared to the Mie solution (thick curve) at $k_0 a = 1.2$ and $\varepsilon_r = 10$ (by using equation (A1)). Three groups of results are related to three different meshes a) 302 tetrahedra; b) 2348 tetrahedra; c) 4322 tetrahedra.

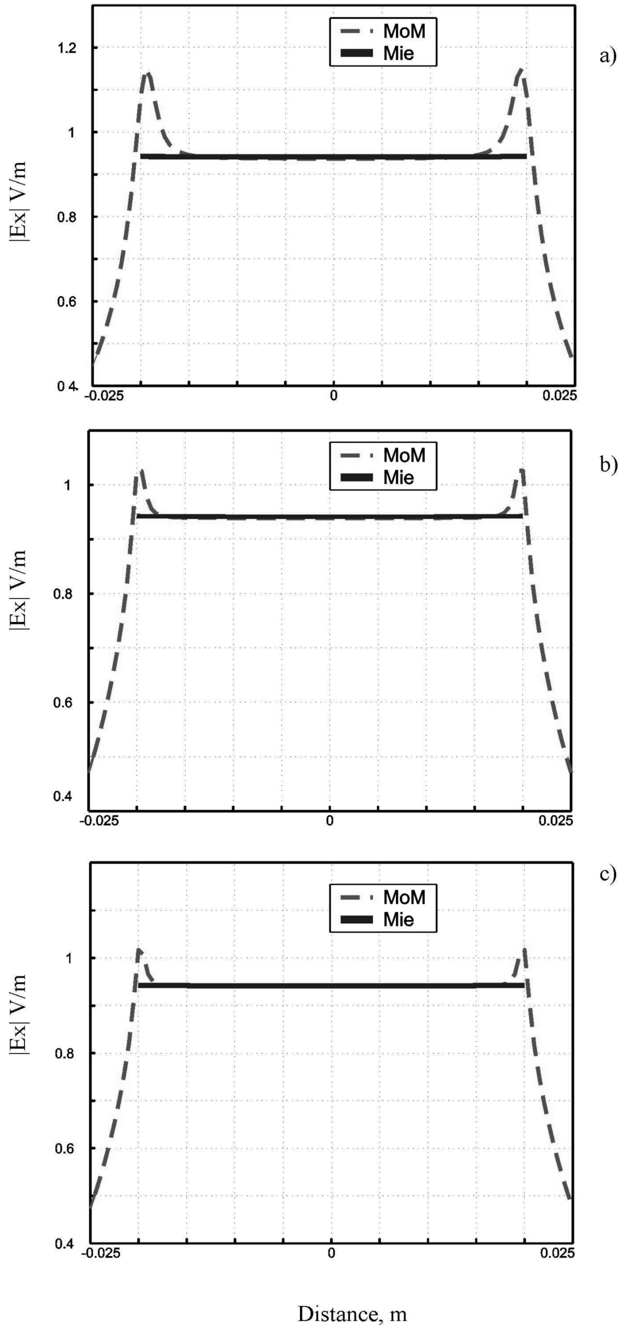


Fig. A.11 Dominant component of the scattered field $\left|E_x^s\right|$ within the dielectric sphere along the *z*-axis compared to the Mie solution (thick curve) at $k_0a = 0.4$ and $\varepsilon_r = 50$ (by using equation (A1)). Three groups of results are related to three different meshes a) 302 tetrahedra; b) 2348 tetrahedra; c) 4322 tetrahedra.

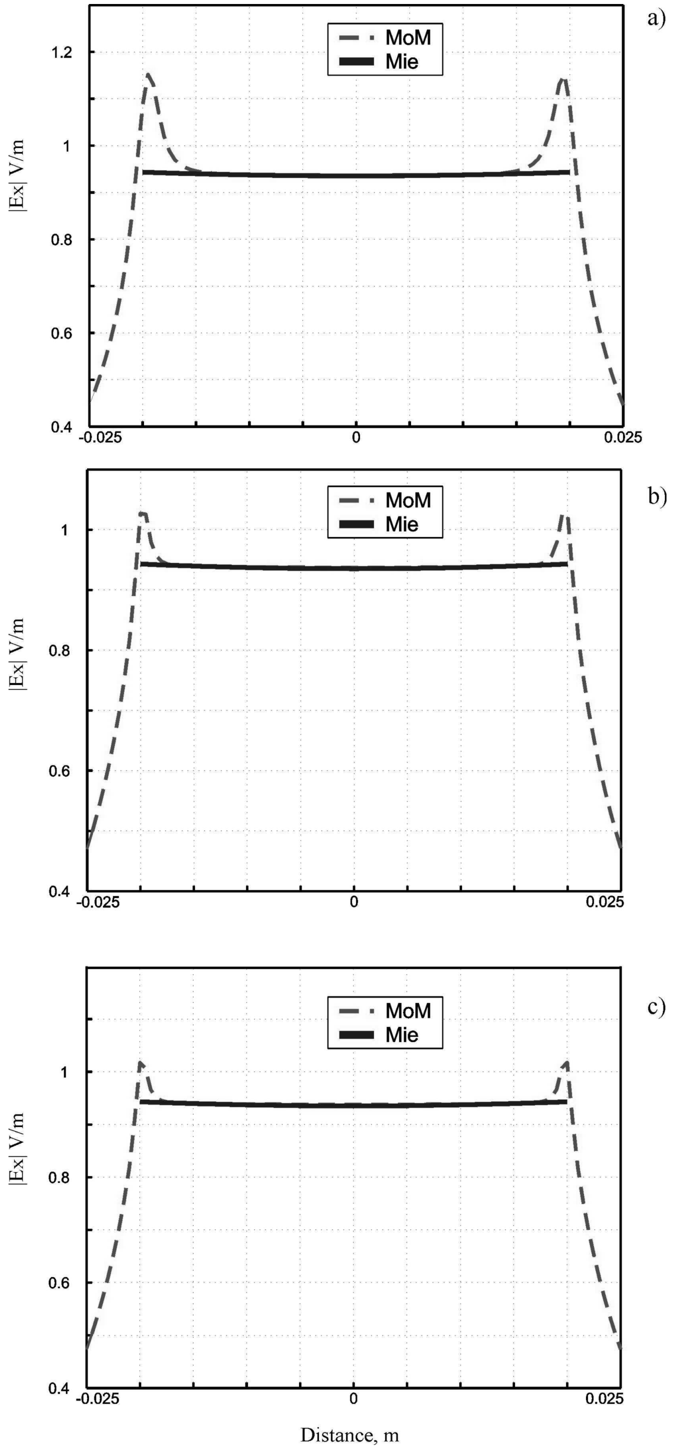


Fig. A.12 Dominant component of the scattered field $\left|E_x^s\right|$ within the dielectric sphere along the *z*-axis compared to the Mie solution (thick curve) at $k_0a = 0.5$ and $\varepsilon_r = 50$ (by using equation (A1)). Three groups of results are related to three different meshes a) 302 tetrahedra; b) 2348 tetrahedra; c) 4322 tetrahedra.

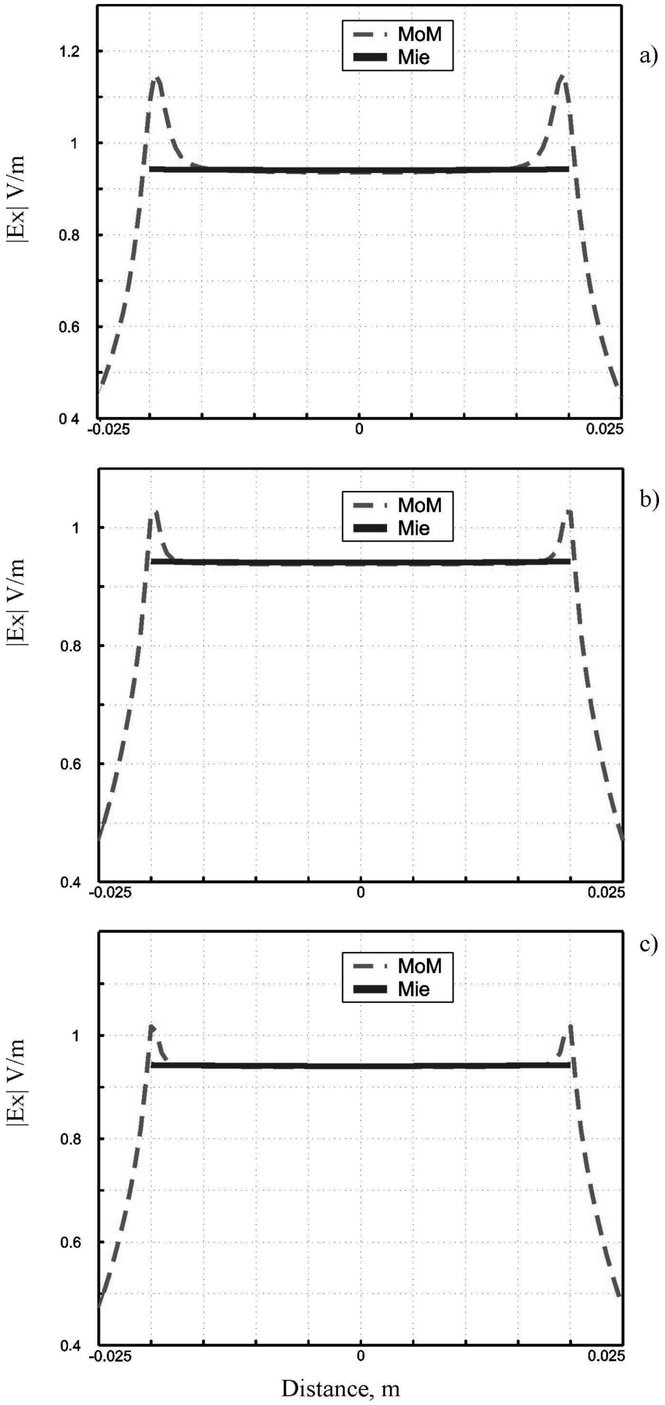


Fig. A.13 Dominant component of the scattered field $\left|E_x^s\right|$ within the dielectric sphere along the z-axis compared to the Mie solution (thick curve) at $k_0a = 0.6$ and $\varepsilon_r = 50$ (by using equation (A1)). Three groups of results are related to three different meshes a) 302 tetrahedra; b) 2348 tetrahedra; c) 4322 tetrahedra.

References to Appendix A

1 D. H. Schaubert, D. R. Wilton, and A. W. Glisson, “A tetrahedral modeling method for electromagnetic scattering by arbitrarily shaped inhomogeneous dielectric bodies,” *IEEE Trans. Antennas and Propagation*, vol. AP-32, no. 1, pp. 77-85, Jan. 1984.

2 J. A. Stratton, *Electromagnetic theory*, IEEE Press, McGraw-Hill, New York, 1941.

3 A. Ishimaru, *Electromagnetic Wave Propagation, Radiation, and Scattering*, Prentice Hall, Upper Saddle River, New Jersey, 1991.

4 A. A Kishk, G. Zhou, and A. W. Glisson, “Analysis of dielectric-resonator antenna with emphasis on hemispherical structures”, *IEEE Trans. Antennas and Propagation Magazine*, vol. 36, no. 2, pp. 20-31, April 1994.

Appendix B. Gaussian Formulae for integral calculation

a. Gaussian formulas for triangular patches [16, 17]

Degree of Precision=1 Number of Points=1

Points	Triangular Coordinates	Weights
a	1/3, 1/3, 1/3	1

Degree of Precision=2 Number of Points=3

Points	Triangular Coordinates	Weights
a	1/2, 1/2, 0	1/3
b	0, 1/2, 1/2	1/3
c	1/2, 0, 1/2	1/3

Degree of Precision=3 Number of Points=4

Points	Triangular Coordinates	Weights
a	1/3, 1/3, 1/3	-27/48

b	0.6, 0.2, 0.2	25/48
c	0.2, 0.6, 0.2	25/48
d	0.2, 0.2, 0.6	25/48

Degree of Precision=4 Number of Points=7

Points	Triangular Coordinates	Weights
a	1/3, 1/3, 1/3	0.2250000000
b	$\alpha_1, \beta_1, \beta_1$	0.1323941527
c	$\beta_1, \alpha_1, \beta_1$	0.1323941527
d	$\beta_1, \beta_1, \alpha_1$	0.1323941527
e	$\alpha_2, \beta_2, \beta_2$	0.1259391805
f	$\beta_2, \alpha_2, \beta_2$	0.1259391805
g	$\beta_2, \beta_2, \alpha_2$	0.1259391805

With

α_1=0.0597158717

β_1=0.4701420641

α_2=0.7974269853

β_2=0.1012865073

Degree of Precision=7 Number of Points=13

Points	Triangular Coordinates	Weights
a	1/3, 1/3, 1/3	-0.149570044467670
b	$\alpha_1, \beta_1, \beta_1$	0.175615257433204
c	$\beta_1, \alpha_1, \beta_1$	0.175615257433204
d	$\beta_1, \beta_1, \alpha_1$	0.175615257433204
e	$\alpha_2, \beta_2, \beta_2$	0.053347235608839
f	$\beta_2, \alpha_2, \beta_2$	0.053347235608839
g	$\beta_2, \beta_2, \alpha_2$	0.053347235608839
h	$\alpha_3, \beta_3, \gamma_3$	0.077113760890257
i	$\gamma_3, \alpha_3, \beta_3$	0.077113760890257
j	$\beta_3, \gamma_3, \alpha_3$	0.077113760890257
k	$\alpha_3, \gamma_3, \beta_3$	0.077113760890257
l	$\gamma_3, \beta_3, \alpha_3$	0.077113760890257
m	$\beta_3, \alpha_3, \gamma_3$	0.077113760890257

with

α_1=0.479308067841923

β_1=0.260345966079038

α_2=0.869739794195568

β_2=0.065130102902216

α_3=0.638444188569809

β_3=0.312865496004875

γ_3=0.04869031542531

b. Analytical formulas for self-integrals over tringlugar patches [18]

The integrals given below [5] are necessary to compute expressions (22) and (23) with the help of equations (24) and (25). Here,

$$a = (\vec{r}_3 - \vec{r}_1)\cdot(\vec{r}_3 - \vec{r}_1), b = (\vec{r}_3 - \vec{r}_1)\cdot(\vec{r}_3 - \vec{r}_2), c = (\vec{r}_3 - \vec{r}_2)\cdot(\vec{r}_3 - \vec{r}_2), \ \mathrm{det} = \sqrt{a - 2b + c} \qquad \text{(B1)}$$

and *A* is the area of the triangular patch. The remaining integrals are obtained using cyclic transformation.

$$\left(\frac{1}{4A^2}\right)\iint_A \iint_{A'} \frac{1}{|\vec{r} - \vec{r}'|}\, dada' = \left[\begin{array}{c} \dfrac{\log\left(\dfrac{\left(a - b + \sqrt{a}\,\mathrm{det}\right)\left(b + \sqrt{a}\sqrt{c}\right)}{\left(-b + \sqrt{a}\sqrt{c}\right)\left(-a + b + \sqrt{a}\,\mathrm{det}\right)}\right)}{6\sqrt{a}} + \\ \dfrac{\log\left(\dfrac{\left(b + \sqrt{a}\sqrt{c}\right)\left(-b + c + \sqrt{c}\,\mathrm{det}\right)}{\left(b - c + \sqrt{c}\,\mathrm{det}\right)\left(-b + \sqrt{a}\sqrt{c}\right)}\right)}{6\sqrt{c}} + \\ \dfrac{\log\left(\dfrac{\left(a - b + \sqrt{a}\,\mathrm{det}\right)\left(-b + c + \sqrt{c}\,\mathrm{det}\right)}{\left(b - c + \sqrt{c}\,\mathrm{det}\right)\left(-a + b + \sqrt{a}\,\mathrm{det}\right)}\right)}{6\,\mathrm{det}} \end{array}\right] \qquad \text{(B2)}$$

$$\left(\frac{1}{4A^2}\right)\iint_{A}\iint_{A'}\lambda_1{}'\lambda_1\frac{1}{|\vec{r}-\vec{r}'|}\,dada'=\left[\begin{array}{l}
\dfrac{\log\left(\dfrac{b+\sqrt{a}\sqrt{c}}{b-c+\sqrt{c}\,\det}\right)}{40\sqrt{c}}+\dfrac{\log\left(\dfrac{-b+c+\sqrt{c}\,\det}{-b+\sqrt{a}\sqrt{c}}\right)}{40\sqrt{c}}+\\
\dfrac{\left(\sqrt{a}\,\det-\sqrt{c}\,\det\right)}{120\det^3}+\dfrac{-\left(\sqrt{a}\sqrt{c}\right)+\sqrt{a}\,\det}{120a^{\frac{3}{2}}}+\\
\dfrac{(2a-5b+3c)\log\left(\dfrac{\left(a-b+\sqrt{a}\,\det\right)\left(c-b+\sqrt{c}\,\det\right)}{\left(b-a+\sqrt{a}\,\det\right)\left(b-c+\sqrt{c}\,\det\right)}\right)}{120\det^3}+\\
\dfrac{(2a+b)\log\left(\dfrac{\left(b+\sqrt{a}\sqrt{c}\right)\left(a-b+\sqrt{a}\,\det\right)}{\left(-b+\sqrt{a}\sqrt{c}\right)\left(-a+b+\sqrt{a}\,\det\right)}\right)}{120a^{\frac{3}{2}}}
\end{array}\right] \quad \text{(B3)}$$

$$\left(\frac{1}{4A^2}\right)\iint_{A}\iint_{A'}\lambda_2{}'\lambda_1\frac{1}{|\vec{r}-\vec{r}'|}\,dada'=\left[\begin{array}{l}
\dfrac{\log\left(\dfrac{b+\sqrt{a}\sqrt{c}}{b-c+\sqrt{c}\,\det}\right)}{120\sqrt{c}}+\dfrac{\log\left(\dfrac{a-b+\sqrt{a}\,\det}{-b+\sqrt{a}\sqrt{c}}\right)}{120\sqrt{a}}+\\
\dfrac{(2a-3b+c)\log\left(\dfrac{a-b+\sqrt{a}\,\det}{b-c+\sqrt{c}\,\det}\right)}{120\det^3}+\\
\dfrac{(a-3b+2c)\log\left(\dfrac{-b+c+\sqrt{c}\,\det}{-a+b+\sqrt{a}\,\det}\right)}{120\det^3}+\\
\dfrac{-3\sqrt{a}+3\det}{120c}+\dfrac{-3\sqrt{c}+3\det}{120a}\\
+\dfrac{(3b+2c)\log\left(\dfrac{-b+c+\sqrt{c}\,\det}{-b+\sqrt{a}\sqrt{c}}\right)}{120c^{\frac{3}{2}}}+\\
+\dfrac{(2a+3b)\log\left(\dfrac{b+\sqrt{a}\sqrt{c}}{-a+b+\sqrt{a}\,\det}\right)}{120a^{\frac{3}{2}}}
\end{array}\right] \quad \text{(B4)}$$

$$\left(\frac{1}{4A^{2}}\right)\iint_{A}\iint_{A'}\lambda_{1}'\,\frac{1}{\left|\vec{r}-\vec{r}'\right|}\,da\,da'=\left[\begin{array}{l}\dfrac{-\log\left(\dfrac{-b+\sqrt{a}\sqrt{c}}{a-b+\sqrt{a}\,\mathrm{det}}\right)}{24\sqrt{a}}+\dfrac{\log\left(\dfrac{b+\sqrt{a}\sqrt{c}}{b-c+\sqrt{c}\,\mathrm{det}}\right)}{24\sqrt{c}}+\\[2ex] \dfrac{-\left(\sqrt{a}\sqrt{c}\right)+\left(\sqrt{a}\,\mathrm{det}\right)}{24a^{\frac{3}{2}}}+\dfrac{(a+b)\log\left(\dfrac{b+\sqrt{a}\sqrt{c}}{-a+b+\sqrt{a}\,\mathrm{det}}\right)}{24a^{\frac{3}{2}}}+\\[2ex] \dfrac{\log\left(\dfrac{a-b+\sqrt{a}\,\mathrm{det}}{b-c+\sqrt{c}\,\mathrm{det}}\right)}{24\,\mathrm{det}}+\dfrac{\log\left(\dfrac{-b+\sqrt{a}\sqrt{c}}{-b+c+\sqrt{c}\,\mathrm{det}}\right)}{12\sqrt{c}}+\\[2ex] \dfrac{\left(\sqrt{a}\,\mathrm{det}\right)-\left(\sqrt{c}\,\mathrm{det}\right)}{24\,\mathrm{det}^{3}}+\dfrac{(a-3b+2c)\log\left(\dfrac{-b+c+\sqrt{c}\,\mathrm{det}}{-a+b+\sqrt{a}\,\mathrm{det}}\right)}{24\,\mathrm{det}^{3}}\end{array}\right] \tag{B5}$$

c. Gaussian formulas for tetrahedral elements [16, 17]

<u>Degree of Precision=1 Number of Points=1</u>

Points	Tetrahedral Coordinates	Weights
A	1/4, 1/4, 1/4, 1/4	1

<u>Degree of Precision=2 Number of Points=4</u>

Points	Tetrahedral Coordinates	Weights
a	$\alpha, \beta, \beta, \beta$	1/4
b	$\beta, \alpha, \beta, \beta$	1/4
c	$\beta, \beta, \alpha, \beta$	1/4
d	$\beta, \beta, \beta, \alpha$	1/4

With
$\alpha = 0.58541020$, $\beta = 0.13819660$

<u>Degree of Precision=3 Number of Points=5</u>

Points	Tetrahedral Coordinates	Weights
A	1/4, 1/4, 1/4, 1/4	-4/5
B	$\alpha, \beta, \beta, \beta$	9/20
C	$\beta, \alpha, \beta, \beta$	9/20
D	$\beta, \beta, \alpha, \beta$	9/20
E	$\beta, \beta, \beta, \alpha$	9/20

With
$\alpha = 1/2$, $\beta = 1/6$

Degree of Precision=4 Number of Points=11

Points	Tetrahedral Coordinates	Weights
a	1/4, 1/4, 1/4, 1/4	-0.131555555555555550e-1
b	$\alpha_1, \beta_1, \beta_1, \beta_1$	0.762222222222222222e-2
c	$\beta_1, \alpha_1, \beta_1, \beta_1$	0.762222222222222222e-2
d	$\beta_1, \beta_1, \alpha_1, \beta_1$	0.762222222222222222e-2
e	$\beta_1, \beta_1, \beta_1, \alpha_1$	0.762222222222222222e-2
f	$\alpha_2, \alpha_2, \beta_2, \beta_2$	0.248888888888888888e-1
g	$\beta_2, \alpha_2, \alpha_2, \beta_2$	0.248888888888888888e-1
h	$\beta_2, \beta_2, \alpha_2, \alpha_2$	0.248888888888888888e-1
i	$\alpha_2, \beta_2, \beta_2, \alpha_2,$	0.248888888888888888e-1
j	$\beta_2, \alpha_2, \beta_2, \alpha_2,$	0.248888888888888888e-1
k	$\alpha_2, \beta_2, \alpha_2, \beta_2$	0.248888888888888888e-1

With

$\alpha_1 = 0.714285714285714285$
$\beta_1 = 0.785714285714285714$
$\alpha_2 = 0.399403576166799219$
$\beta_2 = 0.100596423833200785$

d. Matlab scripts for Gaussian formulae

In this subsection we reproduce two Matlab scripts, which provide with the Gaussian formulae for the triangles (`tri.m`) and tetrahedra (`tet.m`). These scripts also give the formulae for the corresponding barycentric subdivision (of arbitrary order for triangles).

Appendix C. Flow chart and selected Matlab/C++/C codes

The MoM algorithm developed by the author has the format shown in Fig C.1.

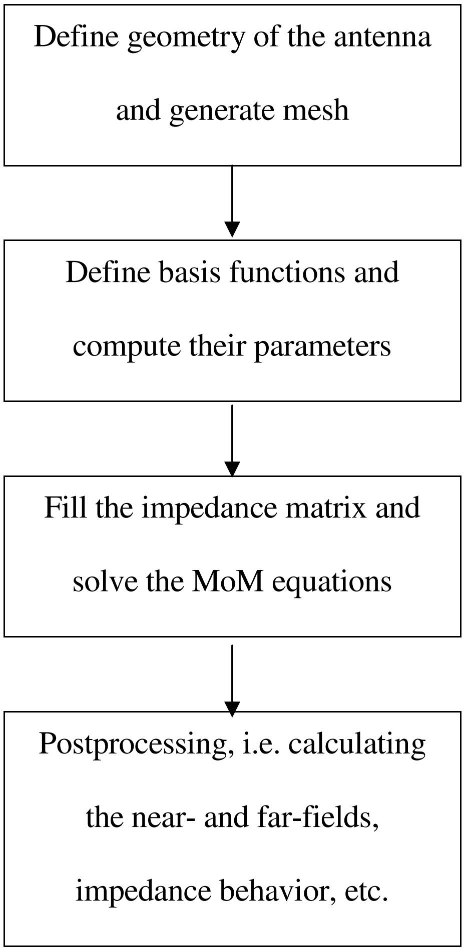

Define geometry of the antenna and generate mesh
Define basis functions and compute their parameters
Fill the impedance matrix and solve the MoM equations
Postprocessing, i.e. calculating the near- and far-fields, impedance behavior, etc.

Fig. C.1 Flow chart for the MoM algorithm

Each block corresponds to a specific group of codes. Below, the critical codes for the third and fourth block are presented for completeness.

```
* =============================================================
* zdd.cpp
*
* This is a MEX-file for MATLAB.
* AntennaLab/ECE/WPI 2003
* Authors: S. Kulkarni
* =============================================================
% zdd
% Computes the impedance matrix for pure dielectric structures.

% Syntax:
%    ZDD    = zdd(GEOM,const,omega_re,omega_im)
%
% Description
% The function zdd returns matrix ZDD (GEOM.Edgestotal by GEOM.Edgestotal)
% which correspond to the impedance matrix for pure dielectric. The
% inputs to the function are two structures GEOM and const and
% the real and imaginary components of the frequency of operation.
% The structure GEOM defines all the parameters of the dielectric
% triangles including self integrals which is obtained from rwgd
% while the structure const defines the electromagnetic constants
% such as mu, epsilon, speed of light etc.*/

#include "mex.h"

#ifdef __ICC

#include <mathimf.h>                              // Intel math library

#else

#include <math.h>

#endif

#ifndef M_PI

#define M_PI 3.1415926535898

#endif


void  ints(double  *PointsFA,int  IndexF,double  *WF,double  kr,double
ki,double *Int_r,double *Int_i);

void int01(double *PointsFA, int IndexT,double *WD,double *PointsFB,
       int IndexF, double *WF ,double kr,double ki,double *Int_r,double
*Int_i);

/* mexFunction is the gateway routine for the MEX-file. */

void

mexFunction( int nlhs, mxArray *plhs[],  int nrhs, const mxArray *prhs[])
{
// Inputs
const mxArray *GEOM, *const_struct;
double omega_re,omega_im, ratio;

// Check input arguments

if (nrhs != 5)
    mexErrMsgTxt("this  function  requires  5  inputs:  GEOM,const,omega_re,
omega_im, ratio");
```

```
// Get the input data
GEOM         = prhs[0];
const_struct = prhs[1];
omega_re    = mxGetScalar(prhs[2]);
omega_im    = mxGetScalar(prhs[3]);
ratio       = mxGetScalar(prhs[4]);

// Get values from structure GEOM

int TetrahedraTotal= (int)mxGetScalar(mxGetField(GEOM, 0, "
TetrahedraTotal "));
int FacesTotal= (int)mxGetScalar(mxGetField(GEOM, 0, "FacesTotal"));
int EdgesTotal = (int)mxGetScalar(mxGetField(GEOM, 0, "EdgesTotal"));
int IndexF             = (int)mxGetScalar(mxGetField(GEOM, 0, "IndexF"));
int IndexT             = (int)mxGetScalar(mxGetField(GEOM, 0, "IndexT"));
double *WD             = mxGetPr(mxGetField(GEOM, 0, "WD"));
double *WF             = mxGetPr(mxGetField(GEOM, 0, "WF"));
double *DiffContrast   = mxGetPr(mxGetField(GEOM, 0, "DiffContrast"));
double *PointsF        = mxGetPr(mxGetField(GEOM, 0, "PointsF"));
double *FF0            = mxGetPr(mxGetField(GEOM, 0, "FF0"));
double *EdgesFNI       = mxGetPr(mxGetField(GEOM, 0, "EdgesFNI"));
double *EdgesFN        = mxGetPr(mxGetField(GEOM, 0, "EdgesFN"));
double *EdgesTNI       = mxGetPr(mxGetField(GEOM, 0, "EdgesTNI"));
double *BasisTC        = mxGetPr(mxGetField(GEOM, 0, "BasisTC"));
double *VolumeT        = mxGetPr(mxGetField(GEOM, 0, "VolumeT"));
double *Contrast       = mxGetPr(mxGetField(GEOM, 0, "Contrast"));
double *PointsT        = mxGetPr(mxGetField(GEOM, 0, "PointsT"));
double *TT0            = mxGetPr(mxGetField(GEOM, 0, "TT0"));
double *EdgesTN        = mxGetPr(mxGetField(GEOM, 0, "EdgesTN"));
double *SizeT          = mxGetPr(mxGetField(GEOM, 0, "SizeT"));
double *SizeF          = mxGetPr(mxGetField(GEOM, 0, "SizeF"));
double *CenterT         = mxGetPr(mxGetField(GEOM, 0, "CenterT"));
double *CenterF         = mxGetPr(mxGetField(GEOM, 0, "CenterF"));

// Get values from structure const

double *epsilon        = mxGetPr(mxGetField(const_struct, 0, "epsilon"));
double *mu             = mxGetPr(mxGetField(const_struct, 0, "mu"));
double *c              = mxGetPr(mxGetField(const_struct, 0, "c"));
double *epsilon_r    = mxGetPr(mxGetField(const_struct, 0, "epsilon_r"));
double *Epsilon_r  = mxGetPr(mxGetField(const_struct, 0, "Epsilon_r"));


// Setup constants

double kr
double ki

double ConstFi=1/(4*M_PI*mxGetScalar(mxGetField(const_struct, 0,
"epsilon")));

double ConstA_r=-(omega_re*omega_re-omega_im* omega_im) * mxGetScalar
(mxGetField(const_struct, 0, "mu"))/(4*M_PI);

double ConstA_i = -(2*omega_im* omega_re) *mxGetScalar (mxGetField (
const_struct, 0, "mu"))/(4*M_PI);


// Outputs

double *ZDD;
plhs[0] = mxCreateDoubleMatrix(2*EdgesTotal*EdgesTotal, 1, mxREAL);
ZDD = (double *)mxGetPr(plhs[0]);
```

```
// Common variables

int c1, c2,c3,c4,m,n;// temp counters
int counter1, counter2;// temp integer variables
double d1,d2, arg1,arg2;
int FacesNontrivial,FullIndex;
double Int_r[1], Int_i[1],Intre,Intim, Ratio;
double one = 1.0;


// Dynamic Variables

int *Index       =   new int[FacesTotal];
double *PointsFA =   new double[3*IndexF];
double *PointsFB =   new double[3*IndexF];
double *PointsTA =   new double[3*IndexT];
double *PointsTB =   new double[3*IndexT];


//Preallocate memory

double*ZDDF_re=(double*)mxMalloc(sizeof(double) * EdgesTotal*EdgesTotal);
double*ZDDF_im=(double*)mxMalloc(sizeof(double) * EdgesTotal*EdgesTotal);
double*ZDDD = (double *)mxMalloc(sizeof(double) * EdgesTotal*EdgesTotal);
double*ZDDT_re=(double*)mxMalloc(sizeof(double) * EdgesTotal*EdgesTotal);
double*ZDDT_im=(double*)mxMalloc(sizeof(double) * EdgesTotal*EdgesTotal);


// zero all the memory locations.

for(c1=0;c1<(EdgesTotal*EdgesTotal);c1++){
    ZDDF_re[c1]= 0;
    ZDDF_im[c1]= 0;
    ZDDD   [c1]= 0;
    ZDDT_re[c1]= 0;
    ZDDT_im[c1]= 0;
}


FacesNontrivial =0;

for(c1=0; c1<FacesTotal;c1++){


d1=fabs((DiffContrast[0+3*c1]*DiffContrast[0+3*c1])+      (DiffContrast[1+
3*c1]*DiffContrast[1+  3*c1])+  (DiffContrast[2  +  3*c1]*DiffContrast[2  +
3*c1]));

    if(d1> 1e-8){

        Index[FacesNontrivial] = c1;

        FacesNontrivial = FacesNontrivial +1;
    }
}




//-------------------------------------------------------------------------
--
```

```
// This loop calculates the self terms. Face with itself.

for(c1=0;c1<FacesNontrivial;c1++){

    FullIndex = Index[c1];

    for(c2=0;c2<(3*IndexF);c2++){

        PointsFA[c2]= PointsF[c2 + c1*IndexF*3];

    }

    *Int_r=0;

    *Int_i=0;

    ints(PointsFA,IndexF,WF,kr,ki,Int_r,Int_i);

    Intre = FF0[FullIndex]+(*Int_r);

    Intim = (*Int_i);

    for(c3=0;c3<(EdgesFNI[FullIndex]);c3++){

        for(c4=0;c4<(EdgesFNI[FullIndex]);c4++){

            m =(int)(EdgesFN[c3+3*FullIndex]-1); //base edge
            n = (int) (EdgesFN[c4 + 3*FullIndex]-1); //base edge


             ZDDF_re[m  +  n*EdgesTotal]  =  ZDDF_re[m+  n*EdgesTotal]  +
            (ConstFi * DiffContrast[c3 + 3*FullIndex]* DiffContrast[c4 +
            3*FullIndex]*Intre);


             ZDDF_im[m  +  n*EdgesTotal]  =  ZDDF_im[m+  n*EdgesTotal]  +
            (ConstFi * DiffContrast[c3 + 3*FullIndex]* DiffContrast[c4 +
            3*FullIndex]*Intim);
        }
    }
}


// This loop calculates the cross terms. Face to face.

for(c1=0;c1<FacesNontrivial;c1++){

    counter1 = Index[c1];

    for(c2=0;c2<c1;c2++){

        counter2 = Index[c2];

        *Int_r=0;

        *Int_i=0;

        d1= CenterF[0+c1*3]-CenterF[0+c2*3];

        Ratio= d1*d1;

        d1= CenterF[1+c1*3]-CenterF[1+c2*3];
        Ratio+= d1*d1;

        d1= CenterF[2+c1*3]-CenterF[2+c2*3];

        Ratio+= d1*d1;

        Ratio/= SizeF[c1]*SizeF[c2];


        if (Ratio < ratio)
        {

            for(c3=0;c3<(3*IndexF);c3++){

                PointsFA[c3]= PointsF[c3 + counter1*IndexF*3];

                PointsFB[c3]= PointsF[c3 + counter2*IndexF*3];
            }


            int01(PointsFA,IndexF,WF,PointsFB,IndexF,WF,kr,ki,Int_r,Int_)
        }

        else
        {
            for(c3=0;c3<3;c3++){

                PointsFA[c3]= CenterF[c3+c1*3];

                PointsFB[c3]= CenterF[c3+c2*3];
            }

            int01(PointsFA,1,&one,PointsFB,1,&one,kr,ki,Int_r,Int_i);
        }

        for(c3=0;c3<(EdgesFNI[counter1]);c3++){

            for(c4=0;c4<(EdgesFNI[counter2]);c4++){

d1=ConstFi*DiffContrast[c3+3*counter1]*DiffContrast[c4+3*counter2]* *Int_r);

d2=ConstFi*DiffContrast[c3+3*counter1]*DiffContrast[c4+3*counter2]* *Int_i);

                m = (int) (EdgesFN[c3 + 3*counter1]-1); //base edge

                n = (int) (EdgesFN[c4 + 3*counter2]-1); //base edge

                ZDDF_re[m + n*EdgesTotal] = ZDDF_re[m+ n*EdgesTotal] + d1;

                ZDDF_im[m + n*EdgesTotal] = ZDDF_im[m+ n*EdgesTotal] + d2;

                ZDDF_re[n + m*EdgesTotal] = ZDDF_re[n+ m*EdgesTotal] + d1;

                ZDDF_im[n + m*EdgesTotal] = ZDDF_im[n+ m*EdgesTotal] + d2;


            }
        }
    }
}
//  Loop over tetrahedra - symmetric - multiplied by the same epsilon

// This loop calculates D/epsilon term
```

```
for(c1=0;c1<TetrahedraTotal;c1++){

    for(c2=0;c2<(EdgesTNI[c1]);c2++){

        for(c3=0;c3<(EdgesTNI[c1]);c3++){

            d1 = BasisTC[0 +c2*3 + c1*6*3] * BasisTC[0 + c3*3 + c1*6*3] +

                 BasisTC[1 +c2*3 + c1*6*3] * BasisTC[1 + c3*3 + c1*6*3] +
                 BasisTC[2 + c2*3 + c1*6*3] * BasisTC[2 + c3*3 + c1*6*3];

            d2 = d1* VolumeT[c1];

            m = (int) (EdgesTN[c2 + 6*c1]-1); //base edge (global number)

            n = (int) (EdgesTN[c3 + 6*c1]-1); //base edge (global number)

            ZDDD[m  +  n*EdgesTotal]  =  ZDDD[m+  n*EdgesTotal]  +  ((d2  *
Contrast[c1])/Epsilon_r[c1]);
        }
    }
}

double realpart,imagpart;

// This loop calculates the self terms. Tetrahedra with itself.


for(c1=0;c1<TetrahedraTotal;c1++){

    for(c2=0;c2<(3*IndexT);c2++){

        PointsTA[c2]= PointsT[c2 + c1*IndexT*3];
    }

    *Int_r=0;

    *Int_i=0;

    ints(PointsTA,IndexT,WD,kr,ki,Int_r,Int_i);

    Intre = (*Int_r)* VolumeT[c1]* VolumeT[c1];

    Intim = (*Int_i)* VolumeT[c1]* VolumeT[c1];

    for(c2=0;c2<(EdgesTNI[c1]);c2++){

        for(c3=0;c3<(EdgesTNI[c1]);c3++){

            d1 = BasisTC[0 + c2*3 + c1*6*3] * BasisTC[0+c3*3 + c1*6*3] +

                 BasisTC[1 + c2*3 + c1*6*3] * BasisTC[1+c3*3 + c1*6*3] +

                 BasisTC[2 + c2*3 + c1*6*3] * BasisTC[2 + c3*3 + c1*6*3];

            arg1 = TT0[c2 + 6*c3 + 6*6*c1] + (d1 * Intre);

            arg2 = d1 * Intim;

            m = (int) (EdgesTN[c2 + 6*c1]-1); //base edge (global number)

            n = (int) (EdgesTN[c3 + 6*c1]-1); //base edge (global number)

            realpart = ConstA_r*arg1 - ConstA_i * arg2;
            imagpart = ConstA_r*arg2 + ConstA_i * arg1;

            ZDDT_re[m + n*EdgesTotal] = ZDDT_re[m+ n*EdgesTotal] +

                (Contrast[c1]* Contrast[c1]*realpart);

            ZDDT_im[m + n*EdgesTotal] = ZDDT_im[m+ n*EdgesTotal] +

                (Contrast[c1]* Contrast[c1]*imagpart);
        }
    }
}
// This loop calculates the cross terms. Tetrahedra to Tetrahedra.


for(c1=0;c1<TetrahedraTotal;c1++){

    for(c2=0;c2<c1;c2++){


        *Int_r=0;

        *Int_i=0;

        d1= CenterT[0+c1*3]-CenterT[0+c2*3];

        Ratio= d1*d1;

        d1= CenterT[1+c1*3]-CenterT[1+c2*3];

        Ratio+= d1*d1;

        d1= CenterT[2+c1*3]-CenterT[2+c2*3];

        Ratio+= d1*d1;

        Ratio                                     /= SizeT[c1]*SizeT[c2];

        if (Ratio < ratio)      // it seems that 225 (sqrt(225)=15) is ok
        {

            for(c3=0;c3<(3*IndexT);c3++){

                PointsTA[c3]= PointsT[c3 + c1*IndexT*3];

                PointsTB[c3]= PointsT[c3 + c2*IndexT*3];
            }

         int01(PointsTA,IndexT,WD,PointsTB,IndexT,WD,kr,ki,Int_r,Int_i);
        }

        else
        {
            for(c3=0;c3<3;c3++){

                PointsTA[c3]= CenterT[c3+c1*3];

                PointsTB[c3]= CenterT[c3+c2*3];
            }

            int01(PointsTA,1,&one,PointsTB,1,&one,kr,ki,Int_r,Int_i);
        }
        Intre=(*Int_r)*VolumeT[c1]*VolumeT[c2]*Contrast[c1]*Contrast[c2];
        Intim=(*Int_i)*VolumeT[c1]*VolumeT[c2]*Contrast[c1]*Contrast[c2];
        for(c3=0;c3<(EdgesTNI[c1]);c3++){
            for(c4=0;c4<(EdgesTNI[c2]);c4++){
                d1 = BasisTC[0+c3*3+c1*6*3]*BasisTC[0 +c4*3 + c2*6*3] +
                     BasisTC[1+c3*3+c1*6*3]*BasisTC[1 +c4*3+c2*6*3] +
                     BasisTC[2+c3*3+c1*6*3]*BasisTC[2 + c4*3 + c2*6*3];
                realpart = (ConstA_r*Intre - ConstA_i*Intim)*d1;
                imagpart = (ConstA_r*Intim + ConstA_i*Intre)*d1;
                m = (int)(EdgesTN[c3+6*c1]-1);//base edge (global number)
                n =(int) (EdgesTN[c4+6*c2]-1);//base edge (global number)
                ZDDT_re[m+n*EdgesTotal]=ZDDT_re[m+n*EdgesTotal]+ ealpart;
                ZDDT_im[m+n*EdgesTotal]=ZDDT_im[m+n*EdgesTotal]+ magpart;
                ZDDT_re[n+m*EdgesTotal]=ZDDT_re[n+m*EdgesTotal]+ ealpart;
                ZDDT_im[n+m*EdgesTotal]=ZDDT_im[n+m*EdgesTotal]+ magpart;
            }
        }
    }
}
for(c1=0;c1<(EdgesTotal*EdgesTotal);c1++){
    ZDD[2*c1] = ZDDF_re[c1] + ZDDD[c1] + ZDDT_re[c1];
    ZDD[2*c1 + 1]   = ZDDF_im[c1] + ZDDT_im[c1];
}
mxFree(ZDDF_re);
mxFree(ZDDF_im);
mxFree(ZDDD);
mxFree(ZDDT_re);
mxFree(ZDDT_im);
delete Index;
delete PointsFA;
delete PointsFB;
delete PointsTA;
delete PointsTB;

}// close main
```

```
* ==============================================================
* slv.cpp
*
* This is a MEX-file for MATLAB.
* AntennaLab/ECE/WPI 2003
* Authors: S. Kulkarni
* ==============================================================

/*=========================================================
/* Mex interface to LAPACK functions zsysv
*=======================================================*/
/*
% slv
% Computes the current vector.

% Syntax:
%  [I D] = slv(ZMM,ZDD,ZMD,VM,VD,omega_re, omega_im);

% Description
% The function slv returns the current vector I (1 by geom.EdgesTotal)
% and D (1 by GEOM.EdgesTotal) in the metal and dielectric respectivley.
% The inputs to the function are the three impedance matrices
% ZMM(geom.EdgesTotal by geom.EdgesTotal)
% ZDD (GEOM.EdgesTotal by GEOM.EdgesTotal) and
% ZMD (geom.EdgesTotal by GEOM.EdgesTotal) obtained from zmm, zdd
% and zmd respectivley. In addition there are the two volatage vectors
% VM (1 by geom.EdgesTotal) and VD (1 by GEOM.EdgesTotal) obtained from
% voltagem and voltaged and the real and imaginary part of the frequency.
% The function uses LAPACK function zsysv (zsysv_ for LINUX) for solving
% equations of the form [Z][ I] = [V] where Z is a complex symmetric
matrix.
%
% See also:
% LAPACK and BLAS functions in MATLAB Help.
*/

#include "mex.h"

#include <math.h>

void mexFunction(int nlhs,mxArray*plhs[],int nrhs,const mxArray *prhs[])
{

char *uplo = "U", msg[101];
int   c1,c2,nrhsB, ldA, ldB, lwork, info, *ipiv;
int arg1, arg2;
int nmetal,ndielectric, dim;
double *ZMM,*ZDD,*ZMD,*VM_re,*VM_im,*VD_re,*VD_im, omega_re, omega_im;
double *I_re, *I_im,*D_re,*D_im;
double *A,*B, *work, *work1;
double epsilon = 8.854187e-12;

if ((nlhs > 2) || (nrhs != 7)) {
 mexErrMsgTxt("Expect 7 input arguments and returns 2 output arguments");
}

// Input Parameters

ZMM= mxGetPr(prhs[0]);
ZDD= mxGetPr(prhs[1]);
ZMD= mxGetPr(prhs[2]);
VM_re= mxGetPr(prhs[3]);
VM_im= mxGetPi(prhs[3]);
VD_re= mxGetPr(prhs[4]);
```

```
VD_im= mxGetPi(prhs[4]);
omega_re= mxGetScalar(prhs[5]);
omega_im= mxGetScalar(prhs[6]);
nmetal= (int)sqrt((double)mxGetM(prhs[0])/2.0);
ndielectric= (int)sqrt((double)mxGetM(prhs[1])/2.0);
dim = nmetal + ndielectric;// dimension of the Z matrix


// Output Parameter

plhs[0] = mxCreateDoubleMatrix(nmetal,1,mxCOMPLEX);
I_re= mxGetPr(plhs[0]);
I_im= mxGetPi(plhs[0]);
plhs[1] = mxCreateDoubleMatrix(ndielectric,1,mxCOMPLEX);
D_re= mxGetPr(plhs[1]);
D_im= mxGetPi(plhs[1]);


//Preallocate memory.

A = (double *)mxMalloc(sizeof(double) * 2*dim*dim);
B = (double *)mxMalloc(sizeof(double) * 2*dim);


// Filling the Z matrix Z = [ZMM ZMD; ZDM ZDD];
// However we only care about the upper Triangular matrix and
//hence ZDM is ignored.Filling ZMM and ZDM.

for(c1=0;c1<nmetal;c1++)
{
    for(c2=0;c2<(2*nmetal);c2++)
    {
        A[c2 + c1*2*dim]= ZMM[c2 + c1*2*nmetal];
    }
    for(c2=0;c2<(2*ndielectric);c2++){
        A[c2+ 2*nmetal + c1*2*dim]= 0;
    }
}


// Filling ZMD and ZDD
for(c1=0;c1<ndielectric;c1++)
{
    for(c2=0;c2<(2*nmetal);c2++)
    {
        A[c2 + c1*2*dim+ 2*dim*nmetal] = ZMD[c2 + c1*2*nmetal];
    }
    for(c2=0;c2<ndielectric;c2++)
    {
      A[2*c2+2*nmetal+c1*2*dim+2*dim*nmetal]=(-ZDD[2*c2+2*c1* ndielectric
      ]* omega_im) - (ZDD[2*c2 +1 + 2*c1*ndielectric]* omega_re);

      A[2*c2+1+2*nmetal+c1*2*dim+2*dim*nmetal]=(ZDD[2*c2+2*c1*
      ndielectric]*omega_re)-(ZDD[2*c2+1 +2*c1*ndielectric]* omega_im);

    }
}

// Filling the V vector V = [VM ; VD];

for(c1=0;c1<nmetal;c1++){
    B[2*c1]= VM_re[c1];
    B[2*c1+ 1]= VM_im[c1];
}
```

```
for(c1=0;c1<ndielectric;c1++){

    B[2*c1+ 2*nmetal]= (-VD_re[c1]*omega_im)-(VD_im[c1]*omega_re);

    B[2*c1+ 1 + 2*nmetal]= (VD_re[c1]*omega_re) -(VD_im[c1]*omega_im);

}
ldA    = dim;
ipiv   = (int *)mxCalloc(dim,sizeof(int));
ldB    = dim;
nrhsB  =  1;
lwork  = -1;
info   =  0;
work   = (double *)mxCalloc(2,sizeof(double));

zsysv(uplo, &dim, &nrhsB, A, &ldA, ipiv, B, &ldB, work, &lwork, &info);

lwork = (int)(work[0]);
work1 = (double *)mxCalloc(2*lwork,sizeof(double));

zsysv(uplo, &dim, &nrhsB, A, &ldA, ipiv, B, &ldB, work1, &lwork, &info);


for (c1 = 0 ; c1 < nmetal ; c1++) {
    I_re[c1] = B[2*c1];
    I_im[c1] = B[2*c1+1];
}

for (c1 = 0 ; c1 < ndielectric ; c1++) {

    D_re[c1] = B[2*c1 + 2*nmetal];
    D_im[c1] = B[2*c1+1 + 2*nmetal];

}

mxFree(work);
mxFree(work1);
mxFree(ipiv);
mxFree(A);
mxFree(B);

}
```

```
/*
* ================================================================
* fieldd.cpp
*
* This is a MEX-file for MATLAB.
* AntennaLab/ECE/WPI 2003
* Authors: S. Kulkarni
* ================================================================
*/
% fieldd
%  Computes radiated/scattered electric and magnetic field and the
%   Poynting vector for radiated/scattered field in a dilectric structure.

% Syntax:
% [Poynting,E,H]=fieldd(const,GEOM,DF,frequency,Points,option)

%
% Description:
% The function field returns vectors calculated in the dilectric volume
% E (3 by Points where the field is calculated)
% Radiated/scattered electric field (complex vector at a point, V/m)
% H (3 by Points where the field is calculated)
% Radiated/scattered magnetic field (complex vector at a point, A/m)
% Poynting  (3 by Points where the field is calculated)
% Poynting vector (W/m^2) for radiated/scattered field
% The inputs to the function are two structures GEOM. and const .The
structureGEOM. defines all the parameters of the metal triangles
including self integrals which is obtained from rwgd while the structure
const defines the electromagnetic constants such as mu, epsilon, speed of
light etc. In addition to these, the other input parameters include the
current in the dielectric DF which is obtained using slv,frequency of
operation, observation point where the field is to be measured and an
addition parameter called option. If option =0 then the scattered filed
is calculated while when the option=1 the total field is calculated.

================================================================
*/



#include "mex.h"
#ifdef __ICC
#include <mathimf.h>// Intel math library
#else
#include <math.h>
#endif
#ifndef M_PI
#define M_PI 3.1415926535898
#endif


/* mexFunction is the gateway routine for the MEX-file. */

void mexFunction(int nlhs,mxArray *plhs[],int nrhs,const mxArray*prhs[] )

{
// Inputs

const mxArray *const_struct;
const mxArray *GEOM;
const mxArray *input;
double *DF_r, *DF_i;
double frequency;
double *Points;
bool option;

if (nrhs != 7)

mexErrMsgTxt("this function requires 7 inputs: const, GEOM, input, D,
frequency, Points , option");

// Outputs

double *P_r;
double *E_r, *E_i;
double *H_r, *H_i;

// Get the input data
const_struct= prhs[0];
GEOM= prhs[1];
input= prhs[2];
DF_r= mxGetPr(prhs[3]);
DF_i= mxGetPi(prhs[3]);
frequency= mxGetScalar(prhs[4]);
Points= mxGetPr(prhs[5]);
int sizeP  = (mxGetDimensions(prhs[5]))[1];
option= (int)mxGetScalar(prhs[6]) != 0;


// Get values from structure GEOM

int FacesTotal     = (int)mxGetScalar(mxGetField(GEOM, 0, "FacesTotal"));
int TetrahedraTotal= (int)mxGetScalar(mxGetField(GEOM, 0, "TetrahedraTotal"));
int IndexF= (int)mxGetScalar(mxGetField(GEOM, 0, "IndexF"));
int IndexT   (int)mxGetScalar(mxGetField(GEOM, 0, "IndexT"));
double *PointsT= mxGetPr(mxGetField(GEOM, 0, "PointsT"));
double *WD= mxGetPr(mxGetField(GEOM, 0, "WD"));
double *EdgesTNI= mxGetPr(mxGetField(GEOM, 0, "EdgesTNI"));
double *BasisTC= mxGetPr(mxGetField(GEOM, 0, "BasisTC"));
double *EdgesTN= mxGetPr(mxGetField(GEOM, 0, "EdgesTN"));
double *VolumeT= mxGetPr(mxGetField(GEOM, 0, "VolumeT"));
double *Contrast= mxGetPr(mxGetField(GEOM, 0, "Contrast"));
double *PointsF= mxGetPr(mxGetField(GEOM, 0, "PointsF"));
double *WF= mxGetPr(mxGetField(GEOM, 0, "WF"));
double *EdgesFNI= mxGetPr(mxGetField(GEOM, 0, "EdgesFNI"));
double *EdgesFN= mxGetPr(mxGetField(GEOM, 0, "EdgesFN"));
double *DiffContrast= mxGetPr(mxGetField(GEOM, 0, "DiffContrast"));
double *CenterT= mxGetPr(mxGetField(GEOM, 0, "CenterT"));
double *CenterF= mxGetPr(mxGetField(GEOM, 0, "CenterF"));
double *SizeT= mxGetPr(mxGetField(GEOM, 0, "SizeT"));
double *SizeF= mxGetPr(mxGetField(GEOM, 0, "SizeF"));


// Get values from structure const

double c_  = mxGetScalar(mxGetField(const_struct, 0, "c"));
double epsilon  = mxGetScalar(mxGetField(const_struct, 0, "epsilon"));
double mu   = mxGetScalar(mxGetField(const_struct, 0, "mu"));
double eta = mxGetScalar(mxGetField(const_struct, 0, "eta"));


// Setup the output data
plhs[0] = mxCreateDoubleMatrix(3, sizeP, mxREAL);
plhs[1] = mxCreateDoubleMatrix(3, sizeP, mxCOMPLEX);
plhs[2] = mxCreateDoubleMatrix(3, sizeP, mxCOMPLEX);
```

```
P_r = mxGetPr(plhs[0]);
E_r = mxGetPr(plhs[1]);
E_i = mxGetPi(plhs[1]);
H_r = mxGetPr(plhs[2]);
H_i = mxGetPi(plhs[2]);


// common variables

int m, n;
int c1, c2, c3, c4;
double d1,d2,d3,d4, d5,d6,d7, e1, e2, e3, G_r, G_i;
double dummy_re, dummy_im;
double INT_rho_re[3], INT_rho_im[3],INT_Grho_re[3], INT_Grho_im[3];
double G_grad_r[3], G_grad_i[3], int_Grho_re[3], int_Grho_im[3],
int_grad_re[3] , int_grad_im[3];


double omega       = 2 * M_PI * frequency;
double k           = omega / c_;
double ConstantA   = mu/(4.0 * M_PI);
double ConstantAE  = ConstantA*omega*omega;
double ConstantAH  =(ConstantA*omega)/mu;
double ConstantF   = -1.0/( epsilon *4.0 *M_PI);// REAL EPSILON
double real[3];



// main loop
double eps = 1.0e-64;

for (c1 = 0 ; c1 < sizeP ; c1++) {

     for (c2 = 0 ; c2 < 3 ; c2++) {

           E_r[c2+ c1*3] = 0;
           E_i[c2+ c1*3] = 0;
           H_r[c2+ c1*3] = 0;
           H_i[c2+ c1*3] = 0;
           P_r[c2+ c1*3] = 0;

     }

      for (c2 = 0 ; c2 < TetrahedraTotal ; c2++) {

        dummy_re = 0;

        dummy_im = 0;

        for(c4=0;c4<3;c4++){

        int_Grho_re[c4]= 0;
        int_Grho_im[c4]= 0;

    }

    for(c3=0; c3< IndexT; c3++){

        d1 = Points[0+ c1*3] - PointsT[3*c3+ 3*IndexT*c2 +0];
        d2 = Points[1+ c1*3] - PointsT[3*c3+ 3*IndexT*c2 +1];
        d3 = Points[2+ c1*3] - PointsT[3*c3+ 3*IndexT*c2 +2];
        d4 = d1 * d1 + d2 * d2 + d3 * d3;

        e1 = d4 / (d4 + eps);
        d4 = d4 +eps;
        d5 = sqrt(d4);

        // Green's function multiplied by weights

        d6 = (e1 * WD[c3]) / d5;
        d7 = k * d5;
        G_r = (+d6 * cos(d7));
        G_i = (-d6 * sin(d7));
        dummy_re += G_r;
        dummy_im += G_i;

        // Green's function gradient

        e1 = - G_r / d4 + k * G_i / d5;//to real part
        e2 = - k * G_r / d5  - G_i / d4;//to imaginary part
        G_grad_r[0] = d1 * e1;
        G_grad_i[0] = d1 * e2;
        G_grad_r[1] = d2 * e1;
        G_grad_i[1] = d2 * e2;
        G_grad_r[2] = d3 * e1;
        G_grad_i[2] = d3 * e2;

        for(c4=0;c4<3;c4++){

            int_Grho_re[c4]= int_Grho_re[c4] + G_grad_r[c4];
            int_Grho_im[c4]= int_Grho_im[c4] + G_grad_i[c4];

        }

    }

    m = (int)EdgesTNI[c2];

    for(c3=0 ; c3< m ; c3++){


      INT_Grho_re[0]=-(BasisTC[2+3*c3+c2*3*6]*int_Grho_re[1])+
      (BasisTC[1+ 3*c3 + c2*3*6]*int_Grho_re[2]);

      INT_Grho_re[1]=(BasisTC[2+3*c3+c2*3*6]*int_Grho_re[0])-
       (BasisTC[0+ 3*c3 + c2*3*6]*int_Grho_re[2]);

      INT_Grho_re[2]=-(BasisTC[1+3*c3+c2*3*6]*int_Grho_re[0])+
      (BasisTC[0+ 3*c3 + c2*3*6]*int_Grho_re[1]);

      INT_Grho_im[0]=-(BasisTC[2+3*c3+c2*3*6]*int_Grho_im[1])+
      (BasisTC[1+ 3*c3 + c2*3*6]*int_Grho_im[2]);

      INT_Grho_im[1]=+(BasisTC[2+3*c3+c2*3*6]*int_Grho_im[0])-
      (BasisTC[0+ 3*c3 + c2*3*6]*int_Grho_im[2]);

      INT_Grho_im[2]=-(BasisTC[1+3*c3+c2*3*6]*int_Grho_im[0])+
      (BasisTC[0+ 3*c3 + c2*3*6]*int_Grho_im[1]);

        for (c4=0 ; c4< 3 ;c4++){

            INT_rho_re[c4]= dummy_re * BasisTC[c4 + 3*c3 + c2*6*3];
            INT_rho_im[c4]= dummy_im * BasisTC[c4 + 3*c3 + c2*6*3];

        }

        n = (int) EdgesTN[c3 + 6*c2]-1;
```

```
        for (c4=0 ; c4< 3 ;c4++){

            E_r[c4 + c1*3] = E_r[c4+ c1*3] + ((DF_r[n]* INT_rho_re[c4]
            -  DF_i[n]*  INT_rho_im[c4])  *  VolumeT[c2]*  Contrast[c2]*
            ConstantAE);

            E_i[c4 + c1*3] = E_i[c4 + c1*3] + ((DF_r[n]* INT_rho_im[c4]
            +  DF_i[n]*  INT_rho_re[c4])  *  VolumeT[c2]*  Contrast[c2]*
            ConstantAE);

            H_i[c4 + c1*3] = H_i[c4 + c1*3] + (-(DF_r[n]* INT_Grho_re[c4]
            -DF_i[n]*   INT_Grho_im[c4])*   VolumeT[c2]*   Contrast[c2]*
            ConstantAH);

            H_r[c4 + c1*3] = H_r[c4 + c1*3] + ((DF_r[n]* INT_Grho_im[c4]
            +DF_i[n]*   INT_Grho_re[c4])*   VolumeT[c2]*   Contrast[c2]*
            ConstantAH);

        }

    }

}

for (c2=0 ; c2<FacesTotal ; c2++){

    for(c4=0;c4<3;c4++){

        int_grad_re[c4]= 0;
        int_grad_im[c4]= 0;

    }

    for(c3=0 ; c3< IndexF ; c3++){

        d1 = Points[0+ c1*3] - PointsF[3*c3+ 3*IndexF*c2 +0];
        d2 = Points[1+ c1*3] - PointsF[3*c3+ 3*IndexF*c2 +1];
        d3 = Points[2+ c1*3] - PointsF[3*c3+ 3*IndexF*c2 +2];
        d4 = d1 * d1 + d2 * d2 + d3 * d3;
        e1 = d4 / (d4 + eps);
        d4 = d4 +eps;
        d5 = sqrt(d4);

        // Green's function multiplied by weights

        d6 = (e1 * WF[c3]) / d5;
        d7 = k * d5;
        G_r = (+d6 * cos(d7));
        G_i = (-d6 * sin(d7));

        // Green's function gradient

        e1 = - G_r / d4 + k * G_i / d5; //to real part
        e2 = - k * G_r / d5  - G_i / d4;//to imaginary part

        G_grad_r[0] = d1 * e1;
        G_grad_i[0] = d1 * e2;
        G_grad_r[1] = d2 * e1;
        G_grad_i[1] = d2 * e2;
        G_grad_r[2] = d3 * e1;
        G_grad_i[2] = d3 * e2;
        for(c4=0;c4<3;c4++){

            int_grad_re[c4] = int_grad_re[c4] + G_grad_r[c4];
            int_grad_im[c4] = int_grad_im[c4] + G_grad_i[c4];

        }
    }


    m = (int)EdgesFNI[c2];

    for(c3=0 ; c3< m ; c3++){

        n = (int) EdgesFN[c3 + 3*c2]-1;

        for (c4=0 ; c4< 3 ;c4++){

            E_r[c4+ c1*3] = E_r[c4 + c1*3] + ((DF_r[n]* int_grad_re[c4]
            -  DF_i[n]*  int_grad_im[c4])  *  DiffContrast[c3  +  3*c2]*
            ConstantF);

            E_i[c4+ c1*3] = E_i[c4 + c1*3] + ((DF_r[n]* int_grad_im[c4]
             +  DF_i[n]*  int_grad_re[c4])  *  DiffContrast[c3  +  3*c2]*
            ConstantF);

        }

    }

}

// Find the Poynting vector.

real[0] = (E_r[1+ c1*3]*H_r[2+ c1*3] + E_i[1+ c1*3]*H_i[2+ c1*3])
    - (E_r[2+ c1*3]*H_r[1+ c1*3] + E_i[2+ c1*3]*H_i[1+ c1*3]);

real[1] = -(E_r[0+ c1*3]*H_r[2+ c1*3] + E_i[0+ c1*3]*H_i[2+ c1*3])
    + (E_r[2+ c1*3]*H_r[0+ c1*3] + E_i[2+ c1*3]*H_i[0+ c1*3]);

real[2] = (E_r[0+ c1*3]*H_r[1+ c1*3] + E_i[0+ c1*3]*H_i[1+ c1*3])
    - (E_r[1+ c1*3]*H_r[0+ c1*3] + E_i[1+ c1*3]*H_i[0+ c1*3]);

for(c2=0;c2<3;c2++){

    P_r[c2+ 3*c1] = 0.5* real[c2];

}

}


// find the total field (scattered plus incident)

double dot;

int i1, i2;

double EI_r[3], EI_i[3];

if(option == 1) {

double *dir        = mxGetPr(mxGetField(input, 0, "dir"));
double *pol        = mxGetPr(mxGetField(input, 0, "pol"));
```

```
for (c1 = 0 ; c1 < sizeP ; c1++) {

    dot = 0;
    for (c2 = 0 ; c2 < 3 ; c2++) {
        dot += dir[c2] * Points[c2 + c1*3];

    }


    d2 = k * dot;
    d3 = +cos(d2);
    d4 = -sin(d2);

    for (c2 = 0 ; c2 < 3 ; c2++) {

        d1 = pol[c2] * d3;
        d2 = pol[c2] * d4;
        E_r[c2 + c1*3] += d1;
        E_i[c2 + c1*3] += d2;
        EI_r[c2] = d1;
        EI_i[c2] = d2;
    }


    for (c2 = 0 ; c2 < 3 ; c2++) {

        i1 = (c2+1)%3;
        i2 = (c2+2)%3;
        H_r[c2 + c1*3] += (dir[i1] * EI_r[i2] -dir[i2] * EI_r[i1]) / eta;
        H_i[c2 + c1*3] += (dir[i1] *EI_i[i2] - dir[i2] * EI_i[i1]) / eta;

    }

    for (c2 = 0 ; c2 < 3 ; c2++) {

        i1 = (c2+1)%3 + c1*3;

        i2 = (c2+2)%3 + c1*3;

        P_r[c2 + c1*3] = 0.5 *(E_r[i1] * H_r[i2] + E_i[i1] * H_i[i2])
            -(E_r[i2] * H_r[i1] + E_i[i2] * H_i[i1]));// + because of conjugate

    }

}

}

}
```