MR_TETRA USERS MANUAL

SANDERS, A Lockheed Martin Co.
95 Canal Street, NCA1-6268
P.O. Box 868
Nashua, NH 03061-0868

Lars S. Andersen
John L. Volakis

October 1999

PROJECT INFORMATION

PROJECT TITLE: ANTENNA SIMULATIONS ON SHIPS FOR AMRFS APPLICATIONS

REPORT TITLE: Users Manual for Code MR_TETRA

U-M REPORT No.: 038954-1-T

CONTRACT
START DATE: October 1999
END DATE: March 2002

DATE: October 18, 1999

SPONSOR: Donald B. Spencer
SANDERS, INC, A Lockheed Martin Co.
MER 24-1583
PO Box 868
Nashua, NH 030601-0868
Phone: (603) 885-4093
Email: donald.b.spencer@lmco.com

SPONSOR
CONTRACT No.:

U-M PRINCIPAL
INVESTIGATOR: John L. Volakis
EECS Dept.
University of Michigan
1301 Beal Ave
Ann Arbor, MI 48109-2122
Phone: (313) 764-0500 FAX: (313) 747-2106
volakis@umich.edu
http://www-personal.engin.umich.edu/~volakis/

CONTRIBUTORS
TO THIS REPORT: Lars S. Andersen, John Volakis

User manual for MR_TETRA.f : A multi-resolution finite element / boundary integral (FE/BI) code with hierarchical tetrahedral elements

Lars S. Andersen and John L. Volakis
Radiation Laboratory
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109-2122, USA

Abstract

The following is a manual for a multi-resolution finite element / boundary integral code for analysis of printed antennas backed by material-filled (dielectric or ferrite) metallic cavities recessed in infinite metallic ground planes. The main characteristic of the code is several attractive options for expanding the electric field within the cavity using lowest order, higher order or any combination of lowest and higher order hierarchical mixed-order tangential vector finite elements for tetrahedra.

1 Introduction

The following is a manual for a multi-resolution finite element / boundary integral (FE/BI) FORTRAN code (referred to as MR_TETRA in the following) for analysis of printed metallic antennas backed by material-filled metallic cavities recessed in infinite metallic ground planes. It is based on a standard FE/BI formulation where the cavity volume is discretized into tetrahedral elements using a FE approach and the mesh is rigorously truncated at the aperture surface using a BI [1]. The electric field within each tetrahedral element is expanded using hierarchical mixed-order tangential vector finite elements (TVFEs) of order 0.5 or 1.5 (referred to as lowest and higher order TVFEs in the following) as presented by Andersen and Volakis [2]. The resulting linear equation system is solved using an iterative solver. The matrix-vector products within the iterative solver can be carried out either directly or via a two-dimensional discrete Fourier transform (DFT) in the spectral domain through application of the adaptive integral method (AIM) [3]. A quasi minimal residual (QMR) [4] or conjugate gradient squared (CGS) [4] solver can be used in case matrix-vector products are computed directly while a QMR or biconjugate gradient (BCG) [4] solver can be used in

case matrix-vector products are computed using a DFT. Upon solution of the resulting linear equation system, input impedance and, optionally, near and far fields (patterns, polarization characteristics) are computed.

The purpose of this manual is to familiarize the reader with the steps involved in running MR_TETRA. This is achieved via a general description as well as a set of examples. It is not the purpose of this manual to demonstrate the merits of the approach on which the code is based. For such a demonstration, see for instance [5].

This manual is organized as follows. Section 2 provides a general description of how to run MR_TETRA. All relevant steps needed by the user will be presented. Section 3 outlines the construction of the input file for MR_TETRA. Section 4 offers a few examples of how to run MR_TETRA. Surface meshes, input files as well as results are given in order to demonstrate various capabilities of the code. Section 5 summarizes the manual. In the appendix, certain additional features of MR_TETRA are described. These features cannot be taken advantage of by merely setting parameters in the input file but also require (minor) changes in the source code itself. The appendix is therefore intended for expert users familiar with FORTRAN in general and, to some extent, this code in particular. The appendix also discusses some of the CPU time bottlenecks of MR_TETRA and outlines possible courses of actions to eliminate those.

2 General description of how to run MR_TETRA

The root directory of MR_TETRA consists of two sub-directories. The sub-directory `src` contains all source files (`.f`), parameter files (`.h`) and a makefile `MMR_TETRA`. Upon compilation, it will also contain all object files (`.o`). The executable `MR_TETRA` will be in the sub-directory `bin` where the input file `MR_TETRA.in` and the universal file describing the mesh (see below) must also be placed. Upon execution, this sub-directory will also contain all output files. The sub-directory `bin` further contains a sub-directory `plot` with MATLAB codes for plotting various output files. These MATLAB codes are not documented in this manual.

To run the code, the following steps must be followed :

1. A mesh in universal file format must be provided in the sub-directory `bin`.

2. In the sub-directory `src`, several dimensioning parameters in the file `param1.h` must be initialized. They must be estimated conservatively, otherwise the code will either crash or produce erroneous results. After a first run for a given mesh, they can manually be set to their smallest possible values for that particular mesh. The parameters are `Nnomax` (max number of nodes), `Nsnomax` (max number of metallic nodes), `Nedmax` (max number of edges), `Nsedmax` (max number of metallic edges), `Nfamax` (max number of triangular faces), `Nsfamax` (max number of metallic triangular faces), `Nmgmax` (max number of material groups), `Npredmax` (max number of probe edges), `Nhovolmax` (max number of higher order sections), `Nbifamax` (max number of triangular BI faces), `Nbiedmax` (max number of BI edges), `IAIMmax` (max number of AIM grid points in the

x-direction), `JAIMmax` (max number of AIM grid points in the y-direction), `IJAIMmax` (max of `IAIMmax` and `JAIMmax`) and `Nnffamax` (max number of near field triangular AIM faces). Note that the file `param1.h` contains several additional dimensioning parameters. These have been set conservatively so the code will run for fully higher order cavities without AIM. This, however, is a tremendous overkill if lowest order TVFEs and AIM is applied. The user is highly advised to familiarize himself / herself with these parameters and set them optimally for each application. Specifically, `BWmax1` and `Nunkmax` can be lowered from 100 and `2*Nedmax+2*Nfamax` to 30 and `Nedmax` if only lowest order TVFEs are applied and `Nmatmax` can be lowered from `BWmax1*Nunkmax+Nbiedmax**2` to `BWmax1*Nunkmax` if AIM is used. This will lead to significant memory savings without altering the results.

3. Build the executable `MR_TETRA` (type `make -f MMR_TETRA` in the sub-directory `src` [1]).

4. Construct the input file `MR_TETRA.in` in the sub-directory `bin` (see below).

5. Run the code (type `MR_TETRA` in the sub-directory `bin`).

3 Construction of input file

The input file for MR_TETRA can be broken into 12 different sections. The format of each of these 12 sections will be described below. "`I`" denotes an integer, "`R`" denotes a double precision real number and "`C`" denotes a double precision complex number. "`Text`" denotes a text line that allows the user to describe the following input parameter(s). Information in all such lines is irrelevant to the code but it makes the input file easier to read and hereby easier to modify. We note that some parts seem unnecessary or redundant. This is partly due to the fact that the code has more options than those described in this manual and partly a result of convenient choices made while the code was under development. We also note that the following description is very general and that specific examples follow later in the report.

Section 1

This section describes the universal file that represents the mesh. Users unfamiliar with universal file formats are referred to SDRC I-DEAS manuals. A universal file section with the descriptor 2411 must give the node coordinates in cm (the ground plane and the metallic antenna must be in the plane $z = 0$ and the cavity must be in the half space $z < 0$). A universal file section with the descriptor 2412 must give the element connectivity and a material group for each element. Optionally, a universal file section with descriptor 2417 (used to describe groups of nodes) and name starting with "F" can be added. This section is used for describing the metallic triangular faces forming the printed antenna. Each node triplet describes a metallic face and hence the group will contain three times as many nodes as there are metallic faces. Note that this is the only way to uniquely describe the metallic antenna.

[1] We note that the compilation of source files was tested on a SUN ULTRA30 work station. Certain parts of the makefile `MMR_TETRA` as well as the timing commands in the code may have to be modified on other platforms.

For simple geometries, it can be done with nodes alone or with node doublets (edges) but in the general case node triplets (triangular faces) must be used. MR_TETRA allows easy specification of rectangular and circular patches via simple geometrical parameters (see Section 2). In these cases, a universal file section with descriptor 2417 is unnecessary. Note that the universal file can be generated using SDRC I-DEAS, the automatic mesher written at the University of Michigan or any other meshing package capable of producing a tetrahedral mesh described in universal file format. The format of this section is the following :

```
Text
"filename" : Universal file
```

Section 2

This section describes the metallic cavity and, optionally, the metallic patch. A cavity code is 1 / 2 for a rectangular / circular cavity. In the former case, $(x, y, z)_{start}$ and $(x, y, z)_{stop}$ for the rectangular cavity as well as $(x, y)_{start}$ and $(x, y)_{stop}$ for a rectangular patch in the plane $z = 0$ is given. In the latter case, the radius r_{cavity} and height h_{cavity} of a circular cavity centered at $(0, 0, -h_{cavity}/2)$ as well as the radius r_{patch} of a circular patch centered at $(0, 0, 0)$ is given. For cavities where the metallic faces are described in the universal file, all patch parameters should be set to zero. The format of this section is either of the following :

```
Text
1 : Cavity code (rectangular cavity)
Text
R : x_cavity_start
R : y_cavity_start
R : z_cavity_start
R : x_cavity_stop
R : y_cavity_stop
R : z_cavity_stop
R : x_patch_start
R : y_patch_start
R : x_patch_stop
R : y_patch_stop
```

```
Text
2 : Cavity code (circular cavity)
Text
R : r_cavity
R : h_cavity
R : r_patch
```

Section 3

This section describes the probe excitation. Each probe has unit magnitude and zero phase and must start and end at a node in the mesh. A probe code (must be 1) and the number

of probes is given along with $(x, y, z)_{start}$ and $(x, y, z)_{stop}$ for each probe. The format of this section is the following :

```
Text
1 : Probe code
Text
I : Number of probes
Text                    ⎫
R : x_probe_start       ⎪
R : y_probe_start       ⎪
R : z_probe_start       ⎬ Repeated for each probe
R : x_probe_stop        ⎪
R : y_probe_stop        ⎪
R : z_probe_stop        ⎭
```

Section 4

This section describes the higher order sections in the mesh. A higher order code of 0 / 1 / 2 / 3 indicates that no higher order TVFEs are used / that higher order TVFEs are contained within rectangular brick sections of the mesh (typically used for rectangular cavities) / that higher order TVFEs are contained within cylindrical shell sections of the mesh (typically used for circular cavities) / that higher order TVFEs are found adaptively [2] based on a solution with lowest order TVFEs only (used for arbitrarily shaped cavities). In case higher order TVFEs are pre-specified, the number of higher order sections is given along with a geometrical description of each higher order section. For rectangular brick sections, $(x, y, z)_{start}$ and $(x, y, z)_{stop}$ are given. The higher order section is then the brick having these points as opposite corners. For cylindrical shell sections, the inner and outer radii r_{start} and r_{stop} and the azimuthal angles ϕ_{start} and ϕ_{stop} are given. ϕ_{start} and ϕ_{stop} must be in degrees and be in the interval $[0°, 360°)$. The higher order section is then the shell from r_{start} to r_{stop} radially, from ϕ_{start} to ϕ_{stop} (counter-clockwise) azimuthally and from $-h_{cavity}$ to 0 in the z-direction. In case higher order TVFEs are found adaptively, the percentage of elements to be refined is given along with file names of output files containing geometrical parameters, input impedance information (the real and imaginary part of the input impedance for each frequency), convergence information (the number of iterations and the final relative residual for each frequency) and timing information (the time for the FEM part, BI part and solver for each frequency as well as time for pre-processing and "the rest of the code") for the adaptive solution. After the initial solution with lowest order TVFEs, the error in each element is indicated via an error indicator and a refined solution is computed utilizing higher order TVFEs in the specified percentage of the elements with the largest errors and lowest order TVFEs elsewhere. The format of this section is either of the following :

```
Text
0 : Higher order code (no higher order TVFEs)
```

[2] For a description of the various adaptive refinement options offered by MR_TETRA, see the appendix.

```
Text
1 : Higher order code (rectangular cavity)
Text
I : Number of higher order sections
Text                    ⎫
R : x_ho_start          ⎪
R : y_ho_start          ⎪
R : z_ho_start          ⎬ Repeated for each higher order section
R : x_ho_stop           ⎪
R : y_ho_stop           ⎪
R : z_ho_stop           ⎭

Text
2 : Higher order code (circular cavity)
Text
I : Number of higher order sections
Text                    ⎫
R : r_ho_start          ⎪
R : r_ho_stop           ⎬ Repeated for each higher order section
R : phi_ho_start        ⎪
R : phi_ho_stop         ⎭

Text
3 : Higher order code (adaptive refinement)
Text
R : Percentage of TVFEs to refine
Text
"filename" : Geometry file
Text
"filename" : Impedance file
Text
"filename" : Convergence file
Text
"filename" : Timing file
```

Section 5

This section describes the TVFEs used for field expansion. A TVFE code 1 / 3 indicates that lowest order TVFEs / a combination of lowest and higher order TVFEs are used. Note that for a TVFE code of 1, the information about higher order TVFEs in Section 4 is not used. This allows the user to toggle between lowest order TVFEs and a combination of lowest and higher order TVFEs by changing only the TVFE code. Note also that a TVFE code of 3 along with a higher order code of 3 inherently includes the case of a TVFE code of 1 for determination of the regions of refinement. The format of this section is the following :

```
Text
I : TVFE code
```

Section 6

This section describes the different material groups. The number of material groups is given. For each material group, the relative permittivity and permeability tensor is described. The former is described by the nine elements $\varepsilon_{xx}, \cdots, \varepsilon_{zz}$ and a conductivity σ in S/cm that gives rise to a frequency dependent loss. The latter is either the unit tensor or a Polder tensor for a gyrotropic material. It is described by a bias code (0 means a unit tensor while 1 / 2 / 3 means a Polder tensor with x- / y- / z-bias) and the value of the precession and saturation frequencies f_0 and f_m in Hz [6]. The format of this section is the following :

```
Text
I : Number of material groups
Text                    ⎫
C : epsilon_xx          ⎪
C : epsilon_xy          ⎪
C : epsilon_xz          ⎪
C : epsilon_yx          ⎪
C : epsilon_yy          ⎪
C : epsilon_yz          ⎬ Repeated for each material group
C : epsilon_zx          ⎪
C : epsilon_zy          ⎪
C : epsilon_zz          ⎪
R : sigma               ⎪
I : Bias axis           ⎪
R : f_0                 ⎪
R : f_m                 ⎭
```

Section 7

This section describes the orders of the Gauss-Legendre integrations for computing the FEM matrix elements, BI near zone elements, BI far zone elements and far fields. The first type of integration is a volume integration over a tetrahedron where an integer 1 / 2 / 3 / 4 denotes 1- / 4- / 5- / 11-point Gauss-Legendre integration. The last three types of integrations are surface integrations over triangles where an integer 1 / 2 / 3 / 4 denotes 1- / 3- / 4- / 7-point Gauss-Legendre integration. The integration order for FEM should be at least 2 for lowest order TVFEs and 4 for higher order TVFEs if exact results are desired. Since accurate evaluation of the near zone elements in the BI matrix is crucial for accurate analysis, the integration order for the BI near zone elements should be set to 4. The integration order for the far zone elements in the BI matrix is usually set to 1 but can be increased to give slightly more accurate results at the price of a significantly slower code. The integration order for far fields is usually set to 4. The format of this section is the following :

```
Text
I : Integration order for FEM
I : Integration order for BI_near
I : Integration order for BI_far
I : Integration order for far field
```

Section 8

This section describes the frequencies at which the geometry is analyzed. The start and stop frequency in Hz as well as the number of frequency points is given. The format of this section is the following :

```
Text
R : f_start
R : f_stop
Text
I : Number of frequency points
```

Section 9

This section describes the iterative solver used for solving the resulting matrix equation system as well as the tolerance imposed on the relative residual as a stopping criterion. A solver code of 1 indicates a QMR solver. A solver code of 2 indicates a CGS solver for the case where matrix-vector products are computed directly and a BCG solver for the case where AIM is applied and matrix-vector products are computed using a DFT. For the case of a higher order code of 3 (adaptive refinement), solver codes 3 / 4 are like solver codes 1 / 2 expect that the lowest order solution is used as the starting guess when computing the refined solution with lowest and higher order TVFEs. The format of this section is the following :

```
Text
I : Solver code
Text
R : Solver tolerance
```

Section 10

This section describes whether AIM is used as well as all AIM parameters. An AIM code 0 / 1 indicates that AIM is not / is used. Regardless of the AIM code, $(x, y)_{start}$ and $(x, y)_{stop}$ for the AIM grid are given along with the number of AIM grid points in the x- and y-direction and the number of near zone AIM grid points in both directions. This information is not used when AIM is not used. Reading the information anyway allows the user to toggle between direct and DFT computation of matrix-vector products by changing only the AIM code. The format of this section is the following :

```
Text
I : AIM code
Text
R : x_AIM_start
R : y_AIM_start
R : x_AIM_stop
R : y_AIM_stop
I : N_AIM_x
```

```
I : N_AIM_y
I : N_AIM_near
```

Section 11

This section describes whether near and far field distributions are computed. For near fields, a near field code of 0 / 1 indicates that they are not / are computed. For far fields, a far field code of 0 / 1 / 2 / 3 indicates that nothing / patterns only / polarization characteristics only / patterns and polarization characteristics are computed. For near fields, the cavity is sampled in 33×33 points in 4 equidistant planes from the bottom of the cavity to the top, as illustrated in Fig. 1. The file `xpyp.dat` has $33 \cdot 33 = 1089$ lines containing the values of (x, y) for the sample points in the plane $z = 0$. The file `FxFy.dat` has $33 \cdot 33 = 1089$ lines containing the corresponding values of the real part of the x- and y-directed electric field as well as the imaginary part of the x- and y-directed electric field. This surface electric field is the one that is being radiated (as an equivalent magnetic current) to give the far field. The file `xpypzp.dat` has $33 \cdot 33 \cdot 4 = 4356$ lines containing the values of (x, y, z) for the sample points in the 4 planes. The file `Fmgn.dat` has $33 \cdot 33 \cdot 4 = 4356$ lines containing the corresponding magnitudes $\sqrt{|E_x|^2 + |E_y|^2 + |E_z|^2}$ of the electric field. Note that an expert user can easily alter the routine `nearfielddist.f` to compute whatever near field quantity might be of interest or to change the sampling rate. For far fields, the half space above the antenna is sampled in $36 \times 16 = 576$ far field points (36 θ-values in $[0°, 90°]$ and 16 ϕ-values in $[0°, 360°]$ with θ and ϕ being the traditional polar and azimuthal angles of a spherical coordinate system related to the (x, y, z)-coordinate system). The file `ff.dat` has $36 \cdot 16$ lines containing the values of θ and ϕ along with a quantity proportional to $\sqrt{|E_\theta|^2 + |E_\phi|^2}$ in the far field. Note that an expert user can easily alter the routine `farfielddist.f` to compute whatever far field quantity might be of interest or to change the sampling rate. The file `pol.dat` contains information about the polarization of the far field normal to the plane of the antenna. It gives the axial ratio (in dB) of the polarization ellipse, the phase difference (in degrees) between the two components and the tilt angle (in degrees) of the polarization ellipse measured from the x-axis. The format of this section is the following :

```
Text
I : Near field code
Text
I : Far field code
```

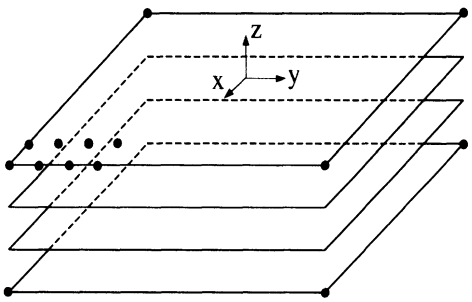

Figure 1: Illustration of cavity sampling.

Section 12

This section describes file names of output files containing geometrical parameters, input impedance information (the real and imaginary part of the input impedance for each frequency), convergence information (the number of iterations and the final relative residual for each frequency) and timing information (the time for the FEM part, BI part, solver and (possibly) determination of regions of refinement for each frequency as well as time for pre-processing and "the rest of the code"). The format of this section is the following :

```
Text
"filename" : Geometry file
Text
"filename" : Impedance file
Text
"filename" : Convergence file
Text
"filename" : Timing file
```

4 Examples

Consider a square metallic patch antenna backed by a rectangular cavity recessed in an infinite metallic ground plane, as illustrated in Fig. 2 (side view) and Fig. 3 (top view). The cavity-backed patch antenna is situated in free space characterized by the permittivity ε_0 and the permeability μ_0. The cavity is of dimensions 1.85 cm $\times$ 1.85 cm $\times$ 0.15 cm and filled with a dielectric material of permittivity $10\,\varepsilon_0$ and conductivity 0.0003 S/cm. The patch is of side length 0.925 cm and centered in the cavity aperture. It is fed by a vertical coaxial line whose outer conductor is attached to the ground plane and whose inner conductor is attached to the patch at the mid point of an edge, as illustrated in Fig. 2 and Fig. 3. The coaxial feed will be modeled as a vertical probe of constant current.

A coarse surface mesh is given in Fig. 4. For mixed-order TVFEs of order 0.5 and 1.5, the particular mesh is too coarse to yield the exact resonant frequency of 4.43GHz as obtained by Schuster and Luebbers [7] and confirmed by Andersen and Volakis for finer meshes [5]. Nevertheless, the mesh is very useful for illustrating the capabilities of MR_TETRA.

Let us analyze the above antenna at 21 frequency points in the interval [3.5GHz,4.5GHz]. We will solve resulting linear equation system using a QMR solver with tolerance 10^{-3}. AIM will not be used. The input file is given below.

```
Name of universal file
  "mesh.unv"
PEC code
  1
(x,y,z)_start, (x,y,z)_stop, (x_pa,y_pa)_start, (x_pa,y_pa)_stop
  -0.4625d0
  -0.4625d0
```

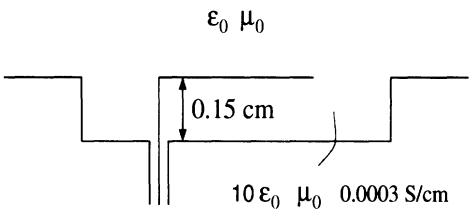

Figure 2: Side view of square metallic patch antenna backed by a dielectric-filled rectangular cavity recessed in an infinite metallic ground plane.

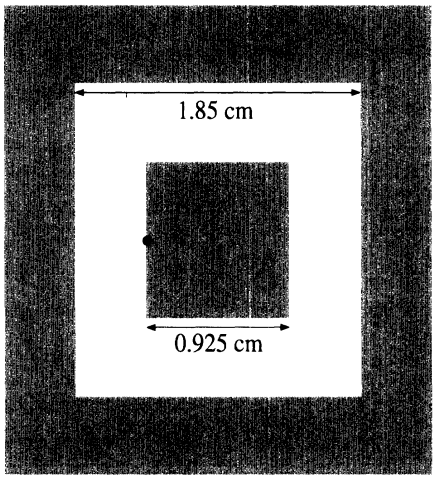

Figure 3: Top view of square metallic patch antenna backed by a dielectric-filled rectangular cavity recessed in an infinite metallic ground plane.

```
  -0.15d0
  1.3875d0
  1.3875d0
  0d0
  0d0
  0d0
  0.925d0
  0.925d0
Probe code
  1
Number of probes
  1
(x_pr,y_pr,z_pr)_start, (x_pr,y_pr,z_pr)_stop for first probe
  0d0
  0.4625d0
  -0.15d0
  0d0
  0.4625d0
  0d0
HO code
  0
```

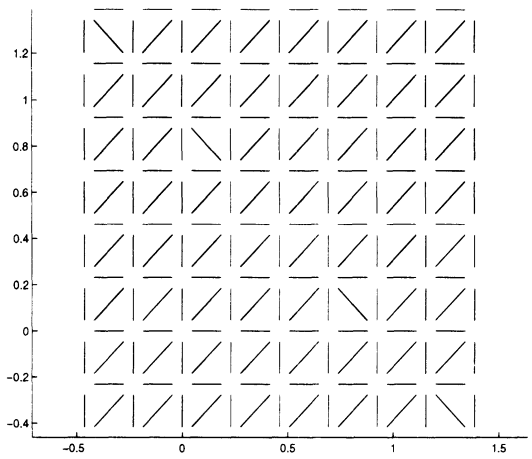

Figure 4: Top view of coarse surface mesh for square metallic patch antenna backed by a dielectric-filled rectangular cavity recessed in an infinite metallic ground plane.

```
TVFE code
  1
Number of material groups
  1
Parameters for the first group
  (10d0,-0d0)
  (0d0,-0d0)
  (0d0,-0d0)
  (0d0,-0d0)
  (10d0,-0d0)
  (0d0,-0d0)
  (0d0,-0d0)
  (0d0,-0d0)
  (10d0,-0d0)
  0.0003d0
  0
  0d0
  0d0
Integration order : FEM, BI_near, BI_far and farfield
  4
  4
  1
  4
Start and stop frequencies
  3.50d9
  4.50d9
Number of frequency points
  21
Solvercode
  1
```

```
Tolerance
  1d-3
AIM code
  0
AIM parameters (x_start, y_start, x_stop, y_stop, I, J, K)
  0d0
  0d0
  0d0
  0d0
  0
  0
  0
Near field distribution code
  0
Far field distribution code
  0
Name of file with geometrical data
  "MR_TETRA.out1"
Name of file with input impedances
  "MR_TETRA.imp1"
Name of file with convergence data
  "MR_TETRA.con1"
Name of file with timing data
  "MR_TETRA.tim1"
```

The output file containing the input impedance for each frequency is given below and also plotted in Fig 5. The antenna is seen to experience the expected resonant behavior. Also, the resonant frequency is seen to be around 3.97GHz which is much smaller than the true resonant frequency 4.43GHz. This is due to the coarse mesh and the fact that lowest order TVFEs have been used.

```
    3500000000.0000    2.5588456293470    41.628451744725
    3550000000.0000    3.2429206914211    45.778072533465
    3600000000.0000    4.2286657863228    50.904979316993
    3650000000.0000    5.7163554782377    57.440189012613
    3700000000.0000    8.1003200176110    66.098467821245
    3750000000.0000    12.236883071922    78.132689480470
    3800000000.0000    20.294341985730    95.967863359927
    3850000000.0000    38.929881217035    124.44216366128
    3900000000.0000    94.338002704657    169.57056063719
    3950000000.0000    278.78777970206    152.06654344528
    4000000000.0000    248.05473459466   -138.59626984899
    4050000000.0000    87.641136274513   -133.27080997795
    4100000000.0000    39.664558360104   -92.517092077889
    4150000000.0000    22.234424951724   -66.678035812629
    4200000000.0000    14.241706452950   -50.108946578041
```

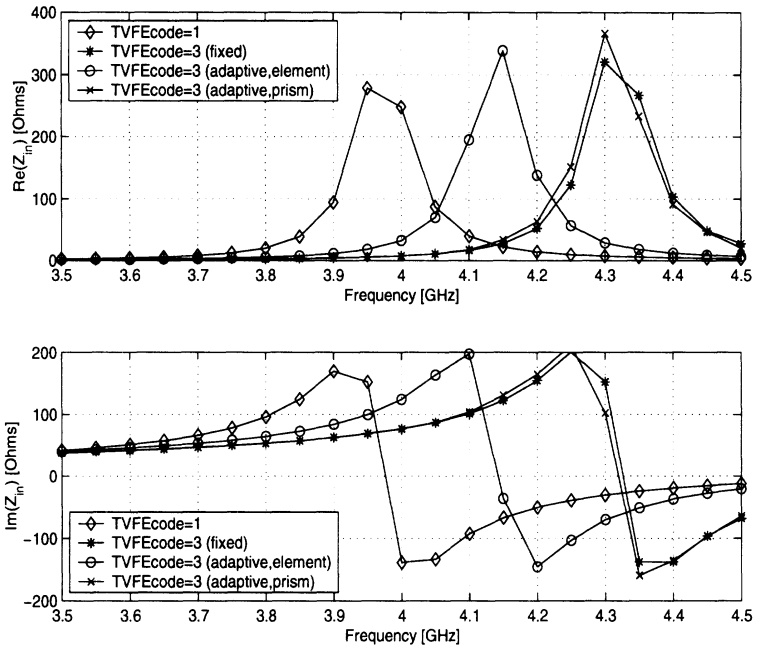

Figure 5: Real and imaginary part of the input impedance of the antenna in Fig. 2 and Fig. 3.

```
4250000000.0000    9.9540096442860    -38.749056086777
4300000000.0000    7.3944826938964    -30.493855847262
4350000000.0000    5.7400309760047    -24.197511427906
4400000000.0000    4.6063185864701    -19.211509191283
4450000000.0000    3.8071120800757    -15.235725559360
4500000000.0000    3.2156270709780    -11.848352913871
```

At the resonant frequency 3.97GHz, let us now look at the polarization of the far field normal to the antenna as well as the far field patterns. Hence, we change Section 8 and 11 of the input file to read

```
Start and stop frequencies
  3.97d9
  3.97d9
Number of frequency points
  1

Near field distribution code
  0
Far field distribution code
  3
```

The polarization of the far field normal to the antenna is described in the file `pol.dat` :

```
  3970000000.0000
  26.122202028302    -85.365890875233    0.23007593744868
```

This output implies that the polarization ellipse is tilted 0.23° from the x-axis and has an axial ratio of 26.12dB. That is, we essentially have linear polarization along the x-axis which is what we expect for this patch. The E-plane ($y = 0$) and H-plane ($x = 0$) patterns at 3.97GHz are given in Fig. 6 (polar angles in $[0°, 90°]$ on the plot corresponds to ϕ-values in $[0°, 180°]$ while polar angles in $[90°, 180°]$ on the plot corresponds to ϕ-values in $[180°, 360°]$; also, the polar angle 90° on the plot corresponds to $\theta = 0°$ and the polar angles 0° and 180° correspond to $\theta = 90°$). These are consistent with the polarization of the far field normal to the antenna and exactly what we expect for this patch.

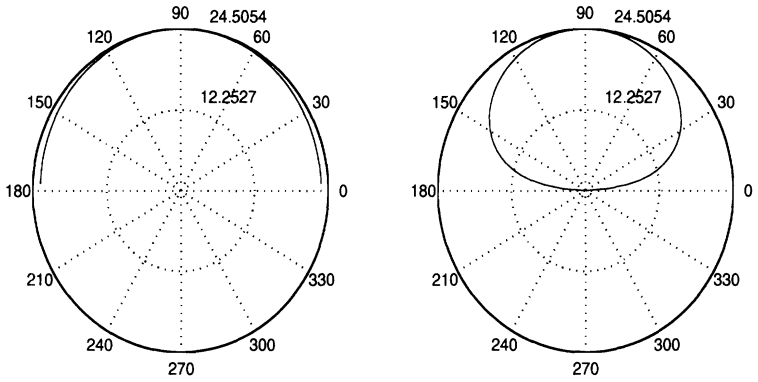

Figure 6: E-plane ($y = 0$) and H-plane ($x = 0$) patterns at 3.97GHz for the antenna in Fig. 2 and Fig. 3.

To demonstrate the use of higher order TVFEs for improved field modeling, we apply higher order TVFEs around the radiating edges of the patch where the field is known to be the strongest and experience the most variation. Hence, we change Section 4 and 5 of the original input file to read

```
HO code
  1
Number of higher order volumes
  2
(x,y,z)_start, (x,y,z)_stop for first higher order volume
  -0.23125d0
  -0.23125d0
  -0.15d0
  0.23125d0
  1.15625d0
  0d0
(x,y,z)_start, (x,y,z)_stop for second higher order volume
  0.69375d0
  -0.23125d0
  -0.15d0
  1.15625d0
  1.15625d0
  0d0
```

```
TVFE code
  3
```

Let us consider the application of AIM and use a DFT to compute the matrix-vector products in the iterative solver. This is terribly inefficient for small problems like the one considered here and only serves the purpose of demonstrating how AIM is used. To invoke AIM, we change Section 10 of the original input file to read

```
AIM code
  1
AIM parameters (x_start, y_start, x_stop, y_stop, I, J, K)
  -0.668055555555d0
  -0.668055555555d0
  1.5930555555555d0
  1.5930555555555d0
  23
  23
  8
```

The output file containing the input impedance for each frequency is given below and also plotted in Fig 5. The antenna is again seen to experience the expected resonant behavior. The resonant frequency is seen to be 4.32GHz which is much closer to the true resonant frequency 4.43GHz than the 3.97GHz found when lowest order TVFEs were used throughout the cavity.

```
3500000000.0000    1.0325245571199    37.824886278503
3550000000.0000    1.1787602915226    39.744351943887
3600000000.0000    1.3630059719665    41.855292671137
3650000000.0000    1.5908198292434    44.211407702366
3700000000.0000    1.8827809279585    46.855809984373
3750000000.0000    2.2511366594008    49.890077408460
3800000000.0000    2.7452949855043    53.404271011424
3850000000.0000    3.4128718362669    57.554844531757
3900000000.0000    4.3376897449660    62.542584004342
3950000000.0000    5.6748167344092    68.707629601590
4000000000.0000    7.6934532336221    76.555049192618
4050000000.0000    10.930495769666    86.945891370685
4100000000.0000    16.607146734682   101.292500931489
4150000000.0000    27.615471055343    122.03417410905
4200000000.0000    52.159382628536    154.29204802524
4250000000.0000    122.49555861383    200.49906174576
4300000000.0000    320.78659000805    152.29837557693
4350000000.0000    267.24188214807   -137.93155070714
4400000000.0000   103.160187785883   -137.62354804365
4450000000.0000    48.490909355324   -96.045352685244
4500000000.0000    27.621340718090   -67.895949125991
```

To demonstrate alternative use of higher order TVFEs for improved field modeling, we apply higher order TVFEs in regions found adaptively. We choose to let 40% of the computational domain be modeled with higher order TVFEs. We will not invoke AIM. Hence, we change Section 4 and 5 of the original input file to read

```
HO code
  3
Percentage of elements to refine
  40d0
Name of file with geometrical data
  "MR_TETRA.out2"
Name of file with input impedances
  "MR_TETRA.imp2"
Name of file with convergence data
  "MR_TETRA.con2"
Name of file with timing data
  "MR_TETRA.tim2"

TVFE code
  3
```

The output file containing the input impedance for each frequency is given below and also plotted in Fig 5. The antenna is again seen to experience the expected resonant behavior. The resonant frequency is seen to be 4.14GHz which is closer to the true resonant frequency 4.43GHz than the 3.97GHz found when lowest order TVFEs were used throughout the cavity but not nearly as close as the 4.32GHz found when higher order TVFEs were placed around the radiating edges of the patch. The reason is that the default refinement approach implemented in MR_TETRA is ineffective for this particular application. This can be remedied by changing the refinement approach, see the appendix.

```
3500000000.0000    1.4535888891966    39.870528744146
3550000000.0000    1.7319599432206    42.538940084124
3600000000.0000    2.1147703938756    45.640297087769
3650000000.0000    2.5877336352995    49.060616280213
3700000000.0000    3.2526423300235    53.209266699692
3750000000.0000    4.1648340860772    58.098521638998
3800000000.0000    5.5496995575525    64.439641743519
3850000000.0000    7.7331209807985    72.828669968818
3900000000.0000    11.450484756409    83.840784528818
3950000000.0000    18.206265103302    99.585388431779
4000000000.0000    32.486024075791    124.44579938528
4050000000.0000    70.211726544462    163.46028209891
4100000000.0000    195.11936736138    197.26892802321
4150000000.0000    338.43135458204   -35.875074055130
4200000000.0000    137.98624942065   -145.55090880899
4250000000.0000    56.699910811376   -103.030465231416
```

```
4300000000.0000    28.805986318971   -69.822218925078
4350000000.0000    18.113550353720   -50.952151626392
4400000000.0000    12.105891667284   -36.744367930337
4450000000.0000    9.0384604765585   -27.693481840863
4500000000.0000    6.9833075220781   -20.593012919138
```

5 Summary

This manual describes how to run MR_TETRA, a multi-resolution FE/BI code for analysis of printed antennas backed by material-filled (dielectric or ferrite) metallic cavities recessed in infinite metallic ground planes. Both a general description as well as examples are given. In the appendix, certain additional features of MR_TETRA are described.

Appendix

In this appendix, certain additional features of MR_TETRA are described. These features cannot be taken advantage of by merely setting parameters in the input file but also require (minor) changes in the source code itself. This appendix is therefore intended for expert users familiar with FORTRAN in general and, to some extent, this code in particular. The appendix also discusses some of the CPU time bottlenecks of MR_TETRA and outlines possible courses of actions to eliminate those.

Adaptive refinement

A very simple refinement strategy is implemented in MR_TETRA. After the initial solution with lowest order TVFEs, the error in each element is indicated via an error indicator and a refined solution is computed utilizing higher order TVFEs in a certain pre-specified percentage of the elements with the largest errors and lowest order TVFEs elsewhere.

Three different error indicators can be used. With the error indicator EI_1, the magnitude of the electric flux density in the center of an element indicates the error within the element. This error indicator is the default option in MR_TETRA. With the error indicators EI_2 or EI_3, the center or average discontinuity in the electric flux density across faces maximized over all faces of an element indicates the error within the element. These error indicators can be used simply by commenting the lines in `refinement.f` corresponding to EI_1 and uncommenting the lines corresponding to EI_2 or EI_3.

The default option in MR_TETRA is to carry out the refinement on an element by element basis. That is, a certain pre-specified percentage of the elements with the highest error are refined. When the mesh is grown from a triangular surface mesh by extruding it into prisms and breaking each of these into tetrahedra, the alternative option of carrying out the refinement on a sub-domain by sub-domain basis is offered. That is, elements are grouped into sub-domains (all elements in the prisms extruded from a given surface triangle form a sub-domain) and the refinement is done based on these instead of the individual elements. This requires uncommenting the lines in `refinement.f` computing the errors in

sub-domains (prisms), commenting the lines in `refinement.f` defining error on an element by element basis and uncommenting the lines in `refinement.f` defining error on a sub-domain by sub-domain (prism by prism) basis. Since the sub-domain by sub-domain refinement approach ensures that groups of elements are clustered together for refinement, this approach often leads to better results than element by element refinement. Specifically, the input impedance of the antenna in Fig. 2 and Fig. 3 as predicted with the sub-domain by sub-domain refinement approach, see the data below, is very close to that when higher order TVFEs are placed around the radiating edges of the patch, see Fig. 5. However, since the sub-domain by sub-domain refinement approach imposes regularity constraints on the volume mesh, the more general element by element refinement approach was chosen as the default option in MR_TETRA. A general sub-domain by sub-domain refinement approach with no volume mesh constraints can easily be implemented but this was not done.

```
3500000000.0000    1.4535888891966    39.870528744146
3550000000.0000    1.7319599432206    42.538940084124
3600000000.0000    2.1147703938756    45.640297087769
3650000000.0000    2.5877336352995    49.060616280213
3700000000.0000    3.2526423300235    53.209266699692
3750000000.0000    4.1648340860772    58.098521638998
3800000000.0000    5.5496995575525    64.439641743519
3850000000.0000    7.7331209807985    72.828669968818
3900000000.0000    11.450484756409    83.840784528818
3950000000.0000    18.206265103302    99.585388431779
4000000000.0000    32.486024075791    124.44579938528
4050000000.0000    70.211726544462    163.46028209891
4100000000.0000    195.11936736138    197.26892802321
4150000000.0000    338.43135458204   -35.875074055130
4200000000.0000    137.98624942065   -145.55090880899
4250000000.0000    56.699910811376   -103.030465231416
4300000000.0000    28.805986318971    -69.822218925078
4350000000.0000    18.113550353720    -50.952151626392
4400000000.0000    12.105891667284    -36.744367930337
4450000000.0000    9.0384604765585    -27.693481840863
4500000000.0000    6.9833075220781    -20.593012919138
```

The regions of refinement can easily be illustrated for the case where the mesh is grown from a triangular surface mesh by extruding it into prisms and breaking each of these into tetrahedra. The files `triangle.dat`, `boundary.dat` and `refinement.dat` are provided for this purpose. In `triangle.dat`, (x, y) coordinates for the three nodes of each surface triangle are given. In `boundary.dat`, (x, y) coordinates for the two nodes of each metallic edge are given. In `refinement.dat`, the number of frequencies is given along with the frequency (in Hz) and surface triangle numbers for each element being refined for each frequency. The MATLAB code `displayref.m` in the sub-directory `bin/plot` in takes these files as input and illustrates the regions of refinement.

Printed antennas within a cavity

The FE/BI formulation on which MR_TETRA is based carries no assumptions regarding the interior of the metallic cavity. Within the framework of MR_TETRA, any antenna composed of metallic surfaces and dielectric or magnetic materials as detailed in this manual can be analyzed. "If you can dream it, you can do it" as Walt Disney said. As a special case, we can analyze printed antennas within a metallic cavity whose side and bottom surfaces are covered by an artificial absorber to simulate an antenna situated in free space. The following is a brief description of how to easily generate meshes suitable for such analysis and changes to be made in MR_TETRA for certain adaptive refinement features to be valid when such analysis is carried out. The example antenna used in the following is a linearly tapered slot antenna but the methodology is not restricted to such antennas.

As mentioned above, we aim to analyze printed antennas in a metallic cavity whose side and bottom surfaces are covered by an artificial absorber to simulate an antenna situated in free space. The tetrahedral volume mesh for the cavity (not just the antenna) can be grown from a triangular surface mesh in the plane of the antenna by extruding it into a number of prism layers in the direction normal to the antenna and breaking each prism into three tetrahedra. To properly specify the various materials in the cavity, we assign a material code from 1 to N_{surmat} to each surface triangle, extrude the triangular surface mesh into N_{lay} prism layers and let each of the tetrahedra that a given prism is broken into have a material code from 1 to $N_{surmat}N_{lay}$ in accordance with the material codes for the triangular surface mesh. That is, a tetrahedron in the nth prism layer extruded from a surface triangle with material code i has the material code $(n-1)N_{surmat}+i$. Provided the metallic antenna is situated at a boundary between two prism layers, it can easily be specified if the corresponding triangles in the surface mesh are marked.

To exemplify this, a triangular surface mesh suitable for generating a valid tetrahedral volume mesh is given in Fig. 7. By assigning material code 1 to triangles in the white or dark gray shaded region and material code 2 to triangles in the light gray shaded region and extruding the surface mesh into N_{lay} prism layers, prisms with tetrahedra having material codes from 1 to $2N_{lay}$ are generated. By properly specifying the physical or artificial material corresponding to each material code as detailed in this manual, the artificial absorbers covering the metallic walls and any pertinent dielectrics are easily specified. The material codes 1 and $2N_{lay}-1$ and all even material codes represent the artificial absorber while the remaining material codes represent free space and, if applicable, dielectrics inside the cavity. The specification of metallic triangles is also illustrated in Fig. 7 where triangles in the dark gray shaded region are metallic while the remaining triangles are non-metallic.

The triangular surface mesh can be grown in I-DEAS with the lower left corner of the surface mesh being $(x,y)=(0,0)$. Material groups are assigned to each surface triangle as detailed above and metallic triangles are marked and stored in a group with descriptor 2430 and a name starting with "M". The geometry is written to an I-DEAS universal file [3]. Note that the triangular surface mesh can also be generated using alternative meshing packages

[3] In certain versions of I-DEAS, groups are written to a universal file with a descriptor of 2432 instead of 2430. If this is the case, 2432 must manually be changed to 2430 after writing the universal file.

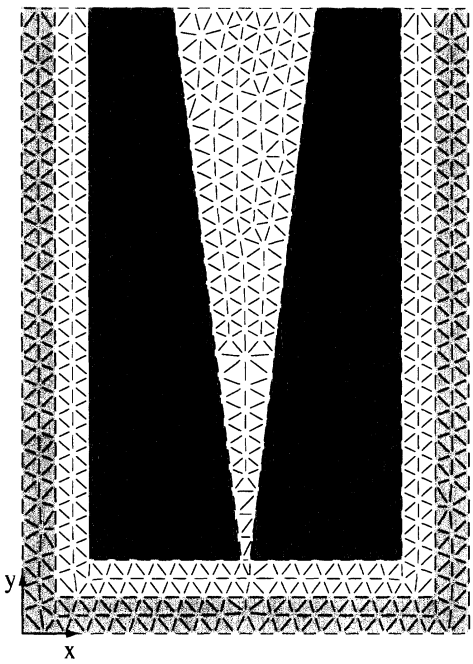

Figure 7: Illustration of surface mesh for tapered slot antenna.

as long as the mesh is eventually described in universal file format.

Based on this triangular surface mesh in universal file format, the automatic mesher developed at the University of Michigan [8, 9] can be utilized to extrude the surface mesh in the z-direction into the desired number of prism layers with the desired height and to generate a tetrahedral volume mesh by breaking each prism into three tetrahedra. In the sub-directory `bin/newmesher`, the universal file describing the triangular surface mesh must be named `surfmesh.unv` and the number of prism layers and the height of each prism layer must be specified in `meshlsasurfunv.in`. Compilation of `meshlsasurfunv.f` via `f77 meshlsasurfunv.f -o meshlsasurfunv` and execution via `meshlsasurfunv` then generates the tetrahedral volume mesh in `mesh.unv`. This tetrahedral volume mesh, however, does not have the proper material codes defined, it does not have any information about the metallic antenna and it is not in a coordinate system compatible with MR_TETRA. To remedy this, copy the files `surfmesh.unv` and `mesh.unv` to `TSAsurf.unv` and `TSAvol_nometal.unv` in the sub-directory `bin/newmesher/TSA`, respectively. In the sub-directory `bin/newmesher/TSA`, compile `interface1.f`, `interface2.f`, `interface3.f` and `interface4.f` via `make -f Minterface1`, `make -f Minterface2`, `make -f Minterface3` and `make -f Minterface4` and execute `interface` (executes `interface1`, `interface2`, `interface3` and `interface4`). This properly defines material codes, flips coordinates and specifies metallic triangles as node-triplets so the resulting file `TSAvol.unv` in the sub-directory `bin/newmesher/TSA` can be directly used by MR_TETRA. For this mesh, the x-direction is identical to the x-direction in the original surface and volume mesh, the z-direction is identical to the y-direction in the original surface and volume mesh and the y-direction is identical to the z-direction in the original volume mesh. The mesh is centered in x and y with the top surface at $z = 0$ and the metallic antenna centered in the plane $y = 0$ (assumes N_{lay} even).

A few changes must be made in MR_TETRA for certain adaptive refinement features to be valid when such analysis is carried out. These are all related to the fact that prisms extend in a transverse cavity direction (y) as opposed to the normal cavity direction (z). They are not all necessary in all cases but all adaptive refinement features described in this manual are valid provided the following changes are made. In `edgefacelo.f`, uncomment and comment proper lines used for construction of `upper`, `Nupper`, `elup` and `upel`. In `refinement.f`, comment and uncomment proper lines where information about the refinement is written to files. Further, uncomment proper lines in the computation of error via EI_1 if the refinement is to be constrained to certain prism layers close to the antenna. Finally, uncomment proper lines if the refinement is to be carried out on a prism by prism basis instead of an element by element basis.

Bottlenecks

MR_TETRA is far from an efficient FE/BI code. It was written solely with modularity and ease of debugging in mind in order to validate and evaluate the merits of multi-resolution FE/BI modeling for electromagnetic analysis. In the process hereof, a large degree of CPU time efficiency was traded. An inefficient but simple code was deemed of more value than an efficient but more complex one. For use of the code for analysis, however, the latter is clearly preferred over the former. Therefore, this section briefly describe some of the CPU time bottlenecks of MR_TETRA and outlines possible courses of actions to eliminate those.

For large computational problems, the pre-processing of MR_TETRA in `edgefacelo.f` is inefficient. The construction of the tables `surrel2`, `Nsurr2`, `surrel3` and `Nsurr3` is based on a double loop over the number of elements. Although done only once, this is extremely time consuming for large problems. For meshes where the element numbers fulfil certain rules (for instance a mesh where the N_{el} elements are organized in N_{lay} layers and the first N_{el}/N_{lay} elements are in the first layer, the next N_{el}/N_{lay} elements are in the next layer and so on), time saving modifications are relatively easy to implement. For completely arbitrary meshes this is not immediate and more substantial changes will have to be made, either in the construction of the tables `surrel2`, `Nsurr2`, `surrel3` and `Nsurr3` or by changing the pre-processing strategy so these tables are not used.

When AIM is not invoked, BI matrix elements are explicitly computed and stored. This requires a large amount of integrations over surface triangles to be carried out. These are currently carried out in an inefficient manner as numerous computations are repeated several times. Changing the routine `BIcompandass.f` could provide a CPU time saving at no sacrifice in accuracy.

When AIM is invoked, BI matrix elements are not explicitly computed and stored. In this case, the dominant CPU time is spent carrying out matrix-vector products within the iterative solver. For each matrix-vector product, the majority of the CPU time is spent performing the six FFTs required with AIM. Speeding up the computation of FFTs is therefore crucial to overall CPU time speed-ups. For an $I \times J$ point AIM grid, $(2I-1) \times (2J-1)$-point two-dimensional FFTs must be computed. In MR_TETRA, each of these is found via a series of one-dimensional FFTs (I J-point FFTs and J I-point FFTs). For the case

of $J = I$, the $(2I-1) \times (2I-1)$-point two-dimensional FFT is directly carried out via a two-dimensional FFT routine. This routine performs best when $2I-1$ is a product of small primes ($I = 23 \Rightarrow 2I-1 = 45 = 3 \cdot 3 \cdot 5$, $I = 41 \Rightarrow 2I-1 = 81 = 3 \cdot 3 \cdot 3 \cdot 3$ and so on). Padding zeroes to allow even point FFTs could make the computations more efficient as the small prime 2 could then be used repeatedly. Also, a different and optimized two-dimensional FFT routine might provide a CPU time speed-up. Further, note that more aggressive AIM settings (coarser AIM grid and smaller near zone threshold) can also provide a speed-up but at the expense of accuracy.

MR_TETRA is written in double precision. Changing MR_TETRA to be single precision would provide a memory and a CPU time reduction at the expense of accuracy. We note that the two-dimensional FFT routine used for computing DFTs within the framework of AIM for $J = I$ is a single precision routine. When applying AIM in this manner, MR_TETRA is therefore only accurate to single precision and hence changing the rest of the code to be single precision is not an unreasonable modification in this case.

References

[1] J.L. Volakis, A. Chatterjee, and L.C. Kempel. *Finite element method for electromagnetics.* IEEE Press, USA, 1998.

[2] L.S. Andersen and J.L. Volakis. 'Hierarchical tangential vector finite elements for tetrahedra'. *IEEE Microwave and Guided Wave Letters*, vol. 8, pp. 127-129, March 1998.

[3] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz. 'AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems'. *Radio Science*, vol. 31, pp. 1225-1251, September-October 1996.

[4] Y. Saad. *Iterative methods for sparse linear systems.* PWS Publishing Company, USA, 1996.

[5] L.S. Andersen and J.L. Volakis. 'Accurate and efficient simulation of antennas using hierarchical mixed-order tangential vector finite elements for tetrahedra'. *IEEE Transactions on Antennas and Propagation*, vol. AP-47, pp. 1240-1243, August 1999.

[6] D.M. Pozar. *Microwave engineering.* Addison-Wesley Publishing Company, Inc., 1990.

[7] J.W. Schuster and R. Luebbers. Private communication.

[8] Z. Li, T. Özdemir, and J.L. Volakis. 'Structured 2-D and 3-D meshing of antenna surfaces and volumes : Users manual'. Radiation Laboratory, University of Michigan, December 1997.

[9] Z. Li and J.L. Volakis. 'Structured 2-D and 3-D meshing of antenna surfaces and volumes : Users manual version 2'. Radiation Laboratory, University of Michigan, July 1999.