



David B. Davidson
Dept. E&E Engineering
University of Stellenbosch
Stellenbosch 7600, South Africa
Tel: +27 21 808 4458;
Fax: +27 21 808 4981
E-mail: davidson@sun.ac.za

Foreword by the Editor

From time to time, advances in both algorithms and computational capability result in the “re-discovery” of a method that was originally described many years back. The theory of characteristic modes (first published over forty years back) is a current case in point, with the topic receiving a lot of attention recently again in computational electromagnetics. This month’s contribution discusses a *MATLAB* implementation of the method, illustrated by application to some interesting antenna structures. As always, we thank the authors for their interesting contribution.

Implementation of the Theory of Characteristic Modes in *MATLAB*

Miloslav Capek, Pavel Hamouz, Pavel Hazdra, and Jan Eichler

Department of Electromagnetic Field
Czech Technical University in Prague
Czech Republic

E-mail: capekmi2@fel.cvut.cz; hamoupav@fel.cvut.cz; hazdrap@fel.cvut.cz; eichljan@fel.cvut.cz

Abstract

This paper describes the implementation of a complex *MATLAB* tool to calculate the characteristic modes and associated antenna parameters. The first code, written in *FORTRAN*, was presented in the early seventieths by Harrington and Mautz. Here, we utilize *MATLAB*, which is widely known and used in the antenna community these days. Because eigen-decomposition is time consuming, parallel and distributed computing is used. Thanks to the hundreds of built-in functions in *MATLAB*, computation of the surface currents from the eigenvectors obtained, as well as other important characteristics, are very easy and effective. The practical features are discussed with two examples.

Keywords: Computational electromagnetics; eigenvalues and eigenfunctions; *MATLAB*; electromagnetic theory

1. Introduction

The antenna tool presented in this paper is based on the well-known Method of Moments (MoM) [1] with the Rao-Wilton-Glisson (RWG) [2] basis functions. Used together with the modal decomposition formulated by Harrington [3], it constitutes a powerful tool for antenna analysis and synthesis. The input for the methods is an arbitrary three-dimensional surface “triangularized” by Delaunay triangularization [4]. The surface is assumed to be made of a perfect electric conductor (PEC) [5]; however, it is possible to include the effects of finite metal thickness and conductivity in post processing (see below). The metal surface must be placed in a vacuum, since the modal approach has been developed for a lossless surrounding medium, with relative permittivity $\epsilon_r = 1$.

The electric-field integral equation (EFIE) is formulated for the metal surface [6]. Two possible treatments of the electric-field integral equation were implemented in the tool: the direct solution by MoM, and the modal decomposition in terms of characteristic modes. Both methods work with the same discretization, and therefore with the same impedance matrix, \mathbf{Z} . As was proven in [7], if all modes on the structure are considered, both methods will give equal results. Former problems with the residual mode were solved recently in [8]. The direct MoM solution is therefore beneficial for checking the TCM (Theory of Characteristic Modes) solution.

In recent years, *MATLAB* [9] has also provided useful new and perspective features, like OOP (object-oriented programming, [10]), GPU computing (see, e.g., [11]), etc. All the presented algorithms were coded and successfully tested in *MATLAB Versions R2011b-R2012a*. All possible exceptions are caught correctly, so the presented version is stable. Some routines are included in the *LAPACK* package [12]. Those who are interested in the application presented here should see [13] for the compiled version, which is free.

2. An Overview of the Background Theory

All necessary theoretical issues are split into three parts:

- Calculation of the impedance matrix,
- Solution of the generalized eigenvalue problem, and
- Post processing.

Based on electric-field-integral-equation formulation [1], an impedance operator, \mathbf{Z} , is defined by

$$\begin{aligned} \mathbf{n}_0 \times \mathbf{E}_1^i(\mathbf{r}') &= -\mathbf{n}_0 \times \mathbf{E}_1^s(\mathbf{r}') \\ &= \mathbf{n}_0 \times \mathbf{Z}[\mathbf{J}(\mathbf{r}')], \end{aligned} \quad (1)$$

where $\mathbf{r}' \in \Omega$, and Ω is the selected structure to be analyzed.

For an arbitrary shape, the impedance operator in Equation (1) can be obtained only in a numerical way. The structure Ω is therefore discretized to M triangles that share N inner – RWG – edges. For details about the calculation of the RWG basis functions, we refer to the classic paper [2]. Our impedance matrix is constructed in accordance with [14].

The discretized impedance operator \mathbf{Z} can be separated as

$$\mathbf{Z} = \mathbf{R} + j\mathbf{X}, \quad (2)$$

where \mathbf{R} and \mathbf{X} have to be real and symmetrical matrices. The following expressions thus hold:

$$\mathbf{R} = \frac{1}{2}(\mathbf{Z} + \mathbf{Z}^*), \quad (3)$$

$$\mathbf{X} = \frac{1}{2j}(\mathbf{Z} - \mathbf{Z}^*), \quad (4)$$

where \mathbf{Z}^* means the complex conjugate of \mathbf{Z} . Matrix \mathbf{Z} is therefore non-Hermitian but symmetrical.

The generalized eigenvalue problem of the form

$$\mathbf{X}\mathbf{J}_n = \lambda_n \mathbf{R}\mathbf{J}_n \quad (5)$$

was derived in [3]. The above-stated decomposition constitutes the characteristic basis of eigencurrents $\{\mathbf{J}_n\}$, with associated eigenvalues $\{\lambda_n\}$. This formulation is known as the Theory of Characteristic Modes (TCM), which can also be formulated in terms of an energetic functional

$$\mathcal{F}(\mathbf{J}_n) = \frac{\langle \mathbf{J}_m, \mathbf{X}\mathbf{J}_n \rangle}{\langle \mathbf{J}_m, \mathbf{R}\mathbf{J}_n \rangle} = \frac{2\omega(W_m^n - W_e^n)}{P_r^n} = \lambda_n. \quad (6)$$

It is obvious that such a functional minimizes the total net power, and maximizes the radiated power.

Because Equation (4) forms an orthogonal system, we apply the following normalization immediately after decomposition:

$$\langle \mathbf{J}_m, \mathbf{R}\mathbf{J}_n \rangle = \delta_{mn}, \quad (7)$$

$$\langle \mathbf{J}_m, \mathbf{X}\mathbf{J}_n \rangle = \delta_{mn}\lambda_n, \quad (8)$$

$$\langle \mathbf{J}_m, \mathbf{Z}\mathbf{J}_n \rangle = (1 + j\lambda_n)\delta_{mn}, \quad (9)$$

to obtain the orthonormal basis with respect to the radiated power. Of course, the above-mentioned normalization not only has benefits, but also some disadvantages (such as extreme scaling of nonradiating modes, etc.). In the following, we denote the modal radiated power as P_r^n , and the total net stored power as $2\omega(W_m^n - W_e^n)$ for the n th mode.

3. MATLAB Implementation

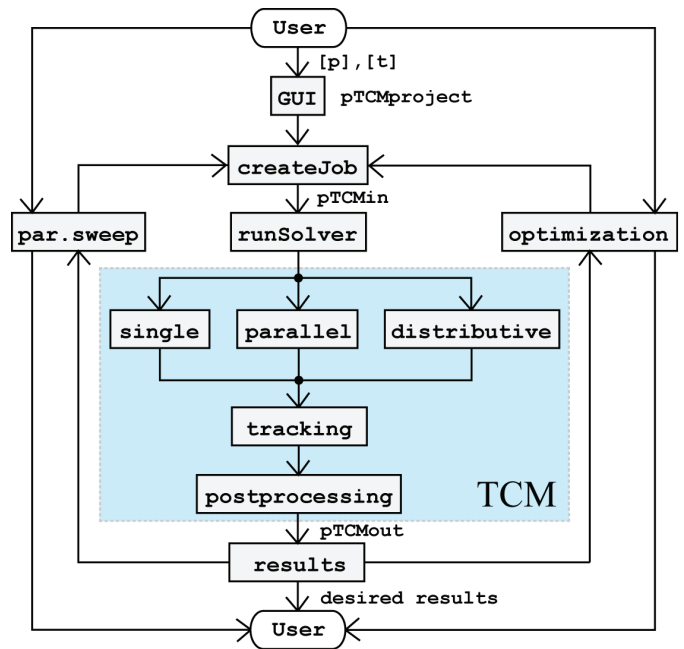
At this point, we start with a description of the schematic diagram depicted in Figure 1. For convenience, the setup of all necessary input data, the GUI, `preTCM`, of the software `TCMapp` was coded (see Figure 2). The `preTCM` routine stores the `pTCMproject` native format that contains all the necessary information to prepare the Theory of Characteristic Modes (TCM) task. For better understanding, we added a schematic code for the simplest Theory of Characteristic Modes decomposition of a thin-wire half-wavelength dipole in *MATLAB* (see Figure 3).

There are also other interfaces connected to the core of the application. These interfaces are fully automatic, and make it possible to perform optimization or the parametric sweep. Both `preTCM`, optimization, and the parametric sweep are directly connected to `createJob`. This part verifies all input data, finds computational resources, generates the `pTCMin` (see Figure 1) batch, and then sends the created job to the solver manager, `runSolver`. The `runSolver` part of the program is independent of low-level solvers (which can be placed, for example, on remote machines). It also makes it possible to add new solvers independently of each other (currently, the authors are working on a novel GPU solver in *MATLAB*).

The frequency samples for which the calculation will be performed have to be specified before the start of the calculation. It is possible to enter a fixed list of frequencies (see the top of Figure 4). This option is useful in all cases where the modal resonant frequencies are more or less known. Unfortunately, such frequencies are not usually known *a priori* for most of the optimization tasks, nor for complicated shapes. Therefore, the adaptive frequency sampler was developed (see the bottom of Figure 4). This powerful feature makes it possible to automatically refine all results near to modal resonances ($\lambda_n = 0$).

The low-level solvers can be directly chosen by the user (see Figure 5 for particular dialogs), or can be selected by the optimizer/parameter-sweep utility. The solution of Equation (4) may then be easily obtained using the `eig` routine in *MATLAB* (in fact, *MATLAB* calls some *LAPACK* functions: QR decomposition, reduction to Hessenberg matrix, etc.). After decomposition, all data are sent back to the `runSolver` service, which performs normalization, Equation (9), tracking, and sorting (in descending order).

The tracking routine [15] is somewhat delicate, because it can only be executed in a heuristic way (for details, see [15-16]). The available tracking methods are simple correlation (top of Figure 6), improved correlation (middle of Figure 6),



`pTCMin =`

```

meshStruct: [1x1 struct]
adaptFreq: [1x1 struct]
freqvs: [500000000 ... ]
mAnalyse: [1x1 struct]
sorting: [1x1 struct]
options: [1x1 struct]
acceleration: [1x1 struct]
postproc: [1x1 struct]
def: [1x1 struct]

```

`pTCMout =`

```

unsorted: [1x1 struct]
def: [1x1 struct]
results: [1x1 struct]
options: [1x1 struct]
freqList: [1x12 double]
adaptFreq: [1x1 struct]
meshStruct: [1x1 struct]
condNumbers: [1x1 struct]
postproc: [1x1 struct]
zmatrix: [1x1 struct]
history: [1x1 struct]
sorting: [1x1 struct]
mAnalyse: [1x1 struct]
sorted: [1x1 struct]

```

Figure 1. (top) A schematic diagram of the Theory of Characteristic Modes application, and (bottom) the structure of the `pTCMin` and `pTCMout` variables.

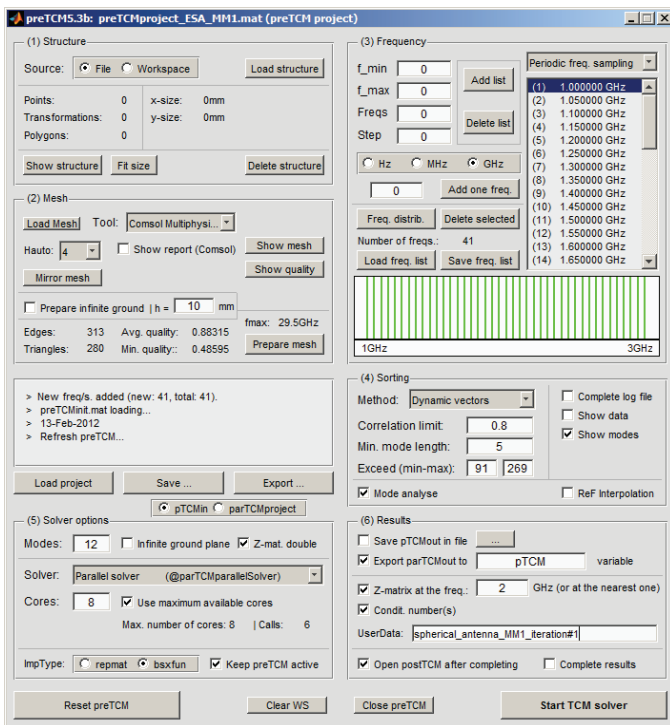


Figure 2. A screenshot of the preTCM tool in MATLAB.

```

1 L = 0.1; % dipole length [m]
2 a = 1e-6; % dipole radius [m]
3 N = 101; % number of segments
4 feed = 51; % feeding position
5 f = 1.5e9; % operational frequency
6 % -----
7 Zin = zeros(1,length(f)); % allocation
8 [Z,J,E] = dipole(f,L,a,N,feed); % MoM
9 ReZ = real(Z); % Re(Z)
10 ImZ = imag(Z); % Im(Z)
11 [Vec,Num] = eig(ImZ, ReZ); % naive TCM
12 Num = diag(Num); % (NxN)->(N,1)
13 [V0,N0] = quickSort(Vec,Num,0.8); % tracking
14 [V0,N0] = signifSort(V0,N0); % sort in ...
    desc. order
15 V0n = zeros(size(V0)); % allocation
16 gamma = zeros(N,1); % ...
17 for j = 1:N % "trivial" normalization
18     V0n(:,j) = V0(:,j)/...
19         sqrt(abs(V0(:,j)'*(ReZ*V0(:,j))));
20     gamma(j) = dot(V0n(:,j),E)/(1 + 1j*N0(j));
21 end
22 Jtot = zeros(size(J)); % allocation
23 for k = 1:N % Jdif = J - Jtot (check)
24     Jtot = Jtot + gamma(k)*V0n(:,k);
25 end
26 Jdif = imag(J) - imag(Jtot); % evan.mode?
27 % Results:
28 figure; plot(V0n); grid on; xlim([0 N]);
29 figure; plot(Jdif); grid on; xlim([0 N]);

```

Figure 3. Simple Theory of Characteristic Modes code for the thin-wire half-wavelength dipole.

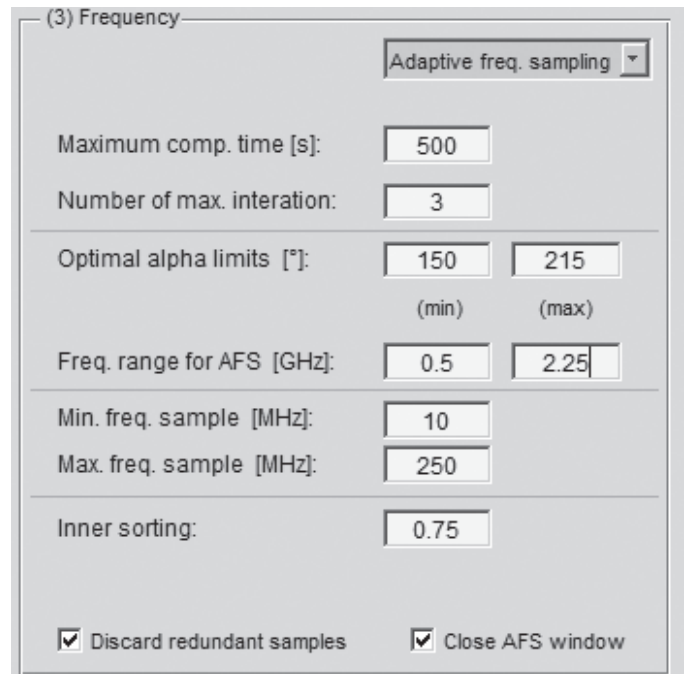
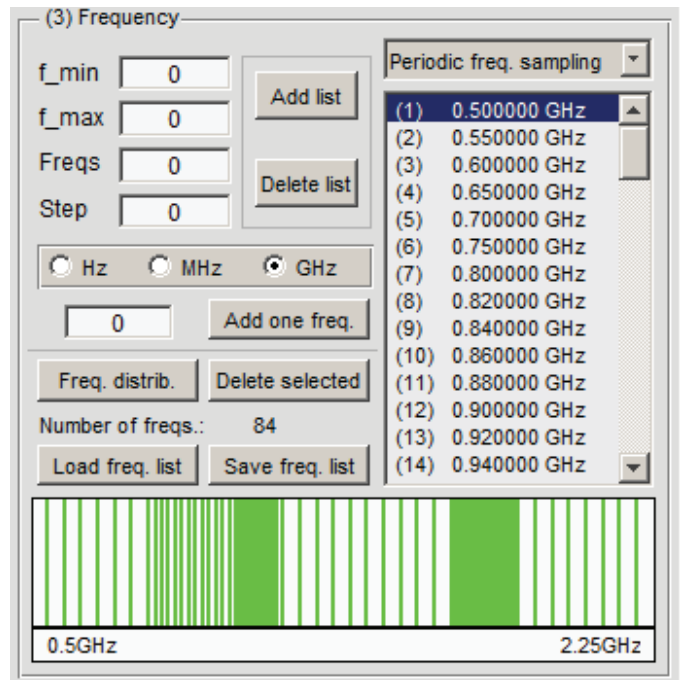


Figure 4. Screenshots of the Theory of Characteristic Modes tool in MATLAB: (top) manual frequency sampler, and (bottom) adaptive frequency sampler.

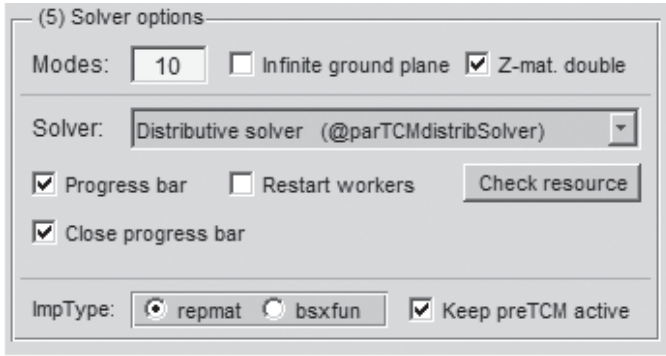
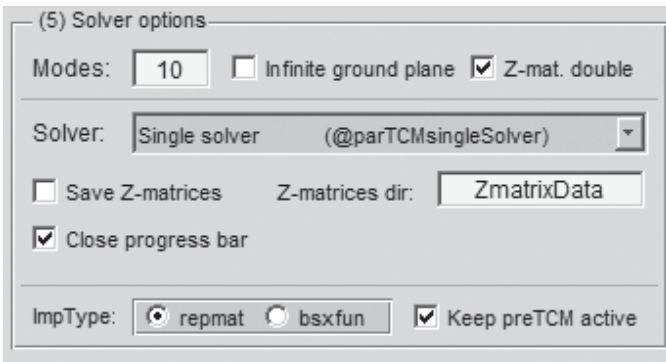


Figure 5. Screenshots of the Theory of Characteristic Modes tool in *MATLAB* showing the available solvers: (top) single, (middle) parallel, and (bottom) distributive.

and complex tracking with spline interpolation of missing data (bottom of Figure 6). The complex tracking method is schematically depicted in Figure 7 as a pseudocode. This part of the code is explicitly presented for the first time.

Finally, N modes have to be found in *MATLAB* and sent to the `postprocessing` part. However, due to numerical noise and the fact that the Theory of Characteristic Modes is ill-conditioned for nonradiating modes, we usually find far fewer modes than N . Fortunately, in practice, only five to 10 modes are needed for the correct analysis of antenna behavior in the selected frequency range. In the case of analysis of an electrically small antenna (ESA), only the first one or two modes are necessary.

The problems of time-consuming inversion and decomposition of the matrix \mathbf{Z} were treated by parallel and distributive computing in *MATLAB* [17]. In parallel mode, only up to eight (12 in *R2012b*) threads can be simultaneously used on a single machine. An example of a typical speedup of a mid-sized task (in terms of the size of the mesh and the number of frequency samples) is depicted in Figure 8. The distributive mode allows the use of a large number of cores (or nodes) to be connected through the LAN at a time, e.g., the initial part of the distributed solver is depicted in Figure 9. This piece of code can be generalized for any cluster calculations in *MATLAB*.

Note that the *MATLAB* cluster can easily be established thanks to the *MATLAB* AdminCenter footnote (however, of course, one must own the *MATLAB Distributed Computing*

Server and/or *Parallel Computing Toolbox*). To achieve the best possible speedup, the code also has to be properly modified in terms of Amdahl's law [19]. This means that the parallelized part of the software has to be maximized.

4. Results Processing

The tools for characteristic-modes computation of an arbitrary three-dimensional PEC surface antenna were described in the previous section. The next step is a post processing of these results. The eigenvalues and eigenvectors are regarded as primary results, since all other modal properties can be computed from them. Primary results (including eigenvalues, eigenvectors, the frequency list, the mesh structure, solver options, etc.) are stored in the `pTCMout` variable (see Figure 1), which can be saved to a file and then used for the following operation.

To give a short summary, we provided two tables (Table 1 and Table 2), containing the most important post-processing functions. In spite of the fact that we briefly describe them,

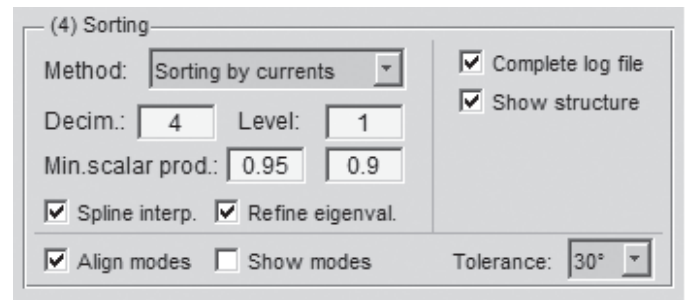
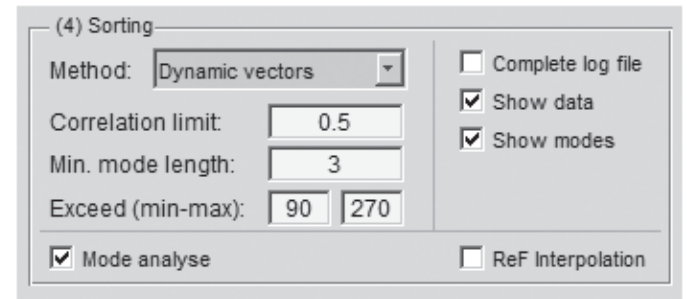
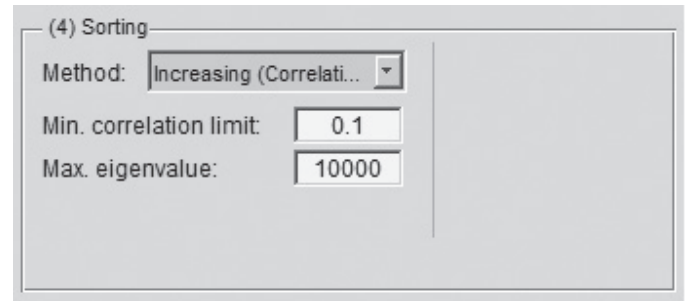


Figure 6. Screenshots of the Theory of Characteristic Modes tool in *MATLAB* showing tracking: (top) simple correlation, (middle) method #1, and (bottom) method #2.

```

1 corrTable = zeros (totNum,totNum,totFreq-1);
2 for thisFreq = 1:(totFreq-1)
3     for thisMode = 1:totNum
4         % calculation of correlation tables (corrTable)
5         end
6     end
7     %% ---INIT-----
8     for initMode = 1:totNum
9         % set up index2Modes, startOfModes & totalModes
10        end % ---INIT/END/-----
11    for sortFreq = 1:(totFreq-1)
12        % allocation (allocModesAtFr0, allocModesAtFr1)
13        %% ---GENERAL_CORR-----
14        for sortMode = 1:totNum
15            % primary function for modes that are ok
16            if nextCorrValue > minCorrLimit && ...
17                nextCorrValue >= checkCorrValue
18                if nextModePosit <= totNum && ...
19                    nextCorrValue >= ...
20                    checkCorrValue
21            end
22        end
23        end % ---GENERAL_CORR/END/-----
24    % find modes that are uncomplete and try ...
25    %% ---RESCUE_CORR-----
26    if numOFFinisModes > 0
27        for tryCompleteThisOne = ...
28            1:length(uncomplModes)
29            % for every unfinished mode
30            for thisRescueMode = ...
31                1:numOFFinisModes
32            % try to find finshed mode
33            end
34            if rescModeValue > minCorrLimit
35            % if correlation is OK and ...
36            if endOFModes(rescPosit) ~= 0
37            % ... original mode is closed: connect
38            end
39            end
40            end % ---RESCUE_CORR/END/-----
41    %% ---MODE_CLOSING-----
42    for thisCloseMode = 1:length(closeModesPos)
43        % close every mode that is unused
44        end % ---MODE_CLOSING/END/-----
45    %% ---MODE_OPENING-----
46    for thisOpenMode = 1:length(openModesPos)
47        % and, if necessary, open new ones
48        end % ---MODE_OPENING/END/-----
49    %% ---CHECK_NUMBER_OF_MODES_ACTUALLY_OPENED---
50    if totalModes(sortFreq+1) > totalModes_limit
51        end % ---CHECK_NUMBER_OF_MODES_ACT.../END/
52    end

```

Figure 7. Pseudo-code of the tracking procedure.

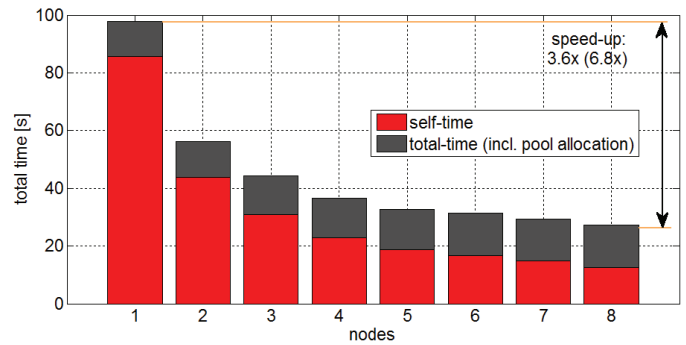


Figure 8. A typical parallel speedup of a medium-sized Theory of Characteristic Modes job.

```

1 % ... some distributive solver allocation
2 res = findResource();
3 if strcmp(res.Type,'local')
4     % RETURN
5 end
6 workers = res.ClusterSize;
7 j = createJob();
8 options.Zprec = opt.Zprecis; % single/double p.
9 if opt.infGround % Green's function for inf. GND
10    set(j,'FileDependencies',...
11        {'distBatch.m','distMeshStruct.mat',...
12         'private/eigFcn.m','private/impGFcn.m'});
13    options.gnd = true;
14 else
15    set(j,'FileDependencies',...
16        {'distBatch.m','distMeshStruct.mat',...
17         'private/eigFcn.m','private/impFcn.m'});
18    options.gnd = false;
19 end
20 delete('distMeshStruct.mat'); % del. temp file
21 if opt.restartWorkers % restart of all workers
22    set(j,'RestartWorker',true);
23 end
24 fiOpt.starTime = clock;
25 for k = 1:FreqSize % prepare all tasks
26    fiOpt.freqs = k;
27    createTask(j,@distBatch,8,...
28        {freqList(k),options},'FinishedFcn',...
29        @(scr,event)distBatchCompleted(fiOpt));
30 end
31 submit(j); % upload job to the cluster
32 waitForState(j,'finished');
33 results = getAllOutputArguments(j);
34 destroy(j); % destroy computed job
35 % results processing ...

```

Figure 9. The start of the distributed solver.

Table 1. Selected post-processing functions.

Function	Description
resCurDistrib	Calculates the modal current density
resCharge	Calculates the modal charge density
resCoeff	Calculates the modal coefficients
resFF	Calculates the modal radiated patterns
resVoltageGap	Connects the voltage gap
resPlaneWave	Generates the incident plane wave
resSumCur	Summarizes the selected modal currents
resSumCharge	Summarizes the selected modal charge density
resQeig	Estimates the modal radiation Q from a slope of λ_n
resQz	Calculates radiation Q factor from input impedance
resQM	Calculates radiation Q according to [18]

Table 2. Selected special display functions.

Function	Description
plotMesh	plots mesh, quality, tr./pt. numbers
plotEigCur	displays modal currents
plotEigCharge	displays charge densities
plotEigNum	displays eigenvalues
plotFF	displays modal radiated patterns

each function has complete help, which can be accessed via the *MATLAB* `help` command, if necessary.

Note also that many operations can be performed via standard built-in functions (such as `plot`, `surf`, etc.).

4.1 Selected Examples

Let us briefly discuss the application of `TCMapp` via two illustrative examples.

The first example is the Minkowski fractal in free space (see Figure 10). The `preTCM` accepts our in-house iterated function system (IFS) fractal [20] format, `FRC` [21], which represents any iterated function system in a compact form with base points, a list of affine transformations, and the iteration for fractal generation. For simplicity, we analyzed the above-mentioned fractal generated only for the first iteration. The initial size was 100 mm \times 60 mm (see Figure 11 for three various triangularizations).

The second example consists of a Franklin antenna [22], depicted in Figure 12. This structure was considered only for the post-processing calculation of modal efficiencies and radiation

Q factors. However, of course the Franklin antenna has to be analyzed first in the Theory of Characteristic Modes tool.

4.2 Geometry and Discretization

There are many possibilities for discretization of the selected structure, because only matrices with all vertex points `[p]` and with all triangles `[t]` are necessary for the computation of the characteristic modes. One can utilize the *MATLAB* PDE toolbox [23]. However, the PDE toolbox usually creates a mesh of poor quality. From the authors' point of view, the Comsol *Multiphysics* [24] option is better: it has an excellent mesh generator.

In order to improve the control of meshing even more, the authors are about to complete an in-house mesh generator based on the `distmesh` code [25]. We thus will be able to generate any required three-dimensional structure, including periodic

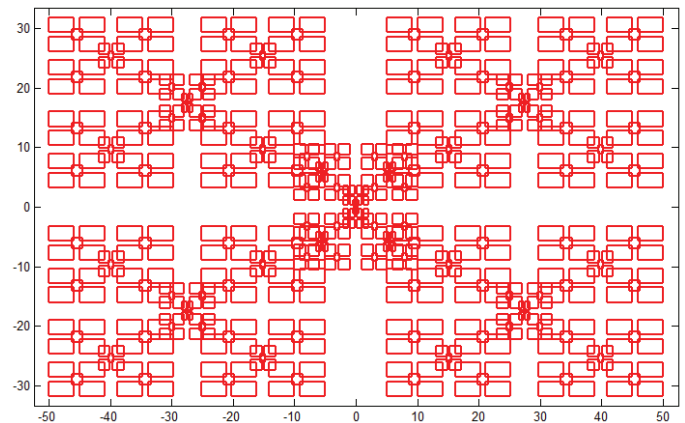


Figure 10. The Minkowski iterated function system fractal, fourth iteration.

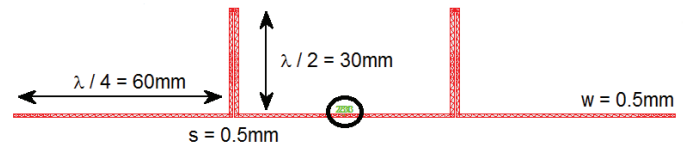


Figure 12. The Franklin antenna.

structures, fractals, and T-junctions, and control the details of the meshing process.

In terms of matrix inversion (direct MoM) as well as matrix decomposition (Theory of Characteristic Modes), it is necessary to control the quality of triangles, which can be defined as

$$\zeta_n = \frac{4\sqrt{3}A_n}{|\mathbf{P}_{n1} - \mathbf{P}_{n2}|^2 + |\mathbf{P}_{n1} - \mathbf{P}_{n3}|^2 + |\mathbf{P}_{n2} - \mathbf{P}_{n3}|^2} \quad (10)$$

The quality of some fractal motifs is depicted in Figure 11. Also, the number of triangles, N , is important, because the time complexity of inversion and decomposition is $\mathcal{O}[\text{inv}(\mathbf{Z})] \propto \mathcal{O}(N^2)$ and $\mathcal{O}[\text{eig}(\mathbf{X}, \mathbf{R})] \propto \mathcal{O}(N^3)$, respectively. Interestingly, for the preparation of larger meshes, the user-defined `repmat` function (utilizing the built-in routine `bsxf`) is faster than the original function `repmat` in Matlab. (`repmat` is exhaustively used to prepare the impedance matrix. To accelerate the calculations, the authors used vector operations everywhere possible.)

4.3 Eigenvalues

Eigenvalues are obtained directly from the decomposition of the impedance matrix and, with the exception of the tracking, they do not require any post processing. These values are continuous throughout the frequency range, they are equal to zero at resonance, and lie within the range of $(-\infty, +\infty)$. Eigenvalues define the character (capacitive/inductive) of the selected mode at a given frequency.

However, better representation of eigenvalues are the so-called eigenangles [26], defined as

$$\delta_n = 180 - \arctan(\lambda_u) \quad (11)$$

Each eigenangle is in the range of $(90^\circ, 270^\circ)$, and the mode resonates for $\delta_n = 180^\circ$. The eigenangles of the Minkowski fractal of the first iteration are depicted in Figure 13, and the adaptive frequency solver was used.

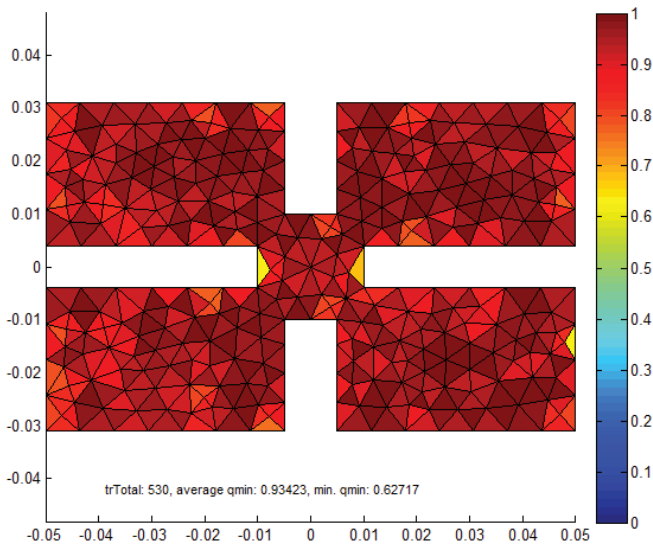
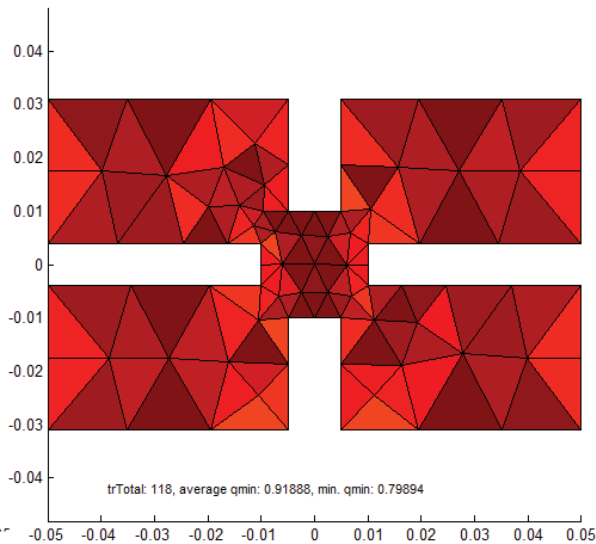
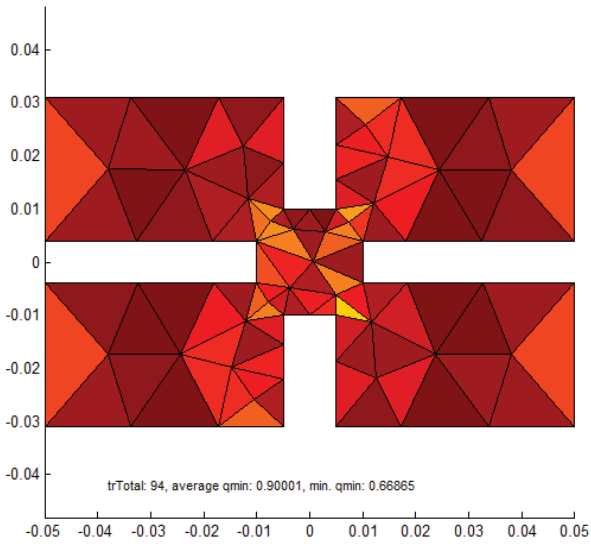


Figure 11. Fractal mesh generation (from coarse to fine, top to bottom). The quality of the triangles was computed according to Equation (10).

4.4 Surface Currents

The eigenvectors are not yet the actual currents flowing on the antenna's surface. The computation of the components of the surface-current density, $[J_x, J_y, J_z]$, over all triangles is performed by the function `resCurDistrib`. It uses the eigenvectors as MoM expansion coefficients, multiplied by the basis functions [2]: for examples, see Figure 14. Our software contains many post-processing routines, such as the calculation of the radiation pattern, the modal (and total) radiation Q factor, the modal (and total) radiation efficiency, and so on.

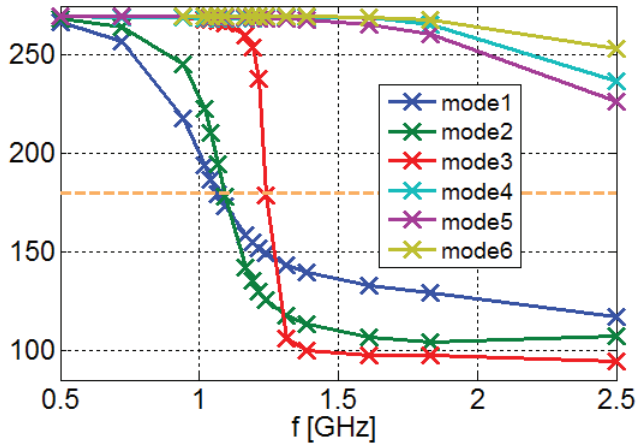
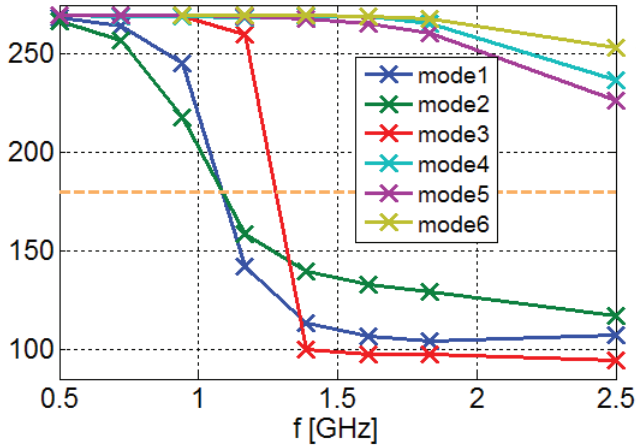
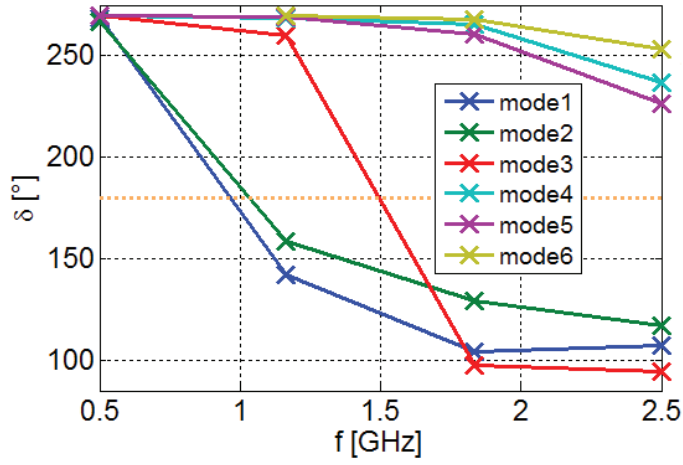


Figure 13. The adaptive frequency sampler: (top to bottom) first, second, and fourth iterations.

4.5 Far-Field Computation

The modal currents are the sources of modal radiated fields and the corresponding modal radiation patterns. Two methods were implemented to compute these radiated fields: a direct method, and the dipole method [27]. The direct method is a numerical integration of analytical expressions for the radiation vector $\mathbf{F}(\theta, \phi)$ [5]. The input is a surface-current density over all triangles, which first has to be computed by the `resEichCur` function. Since the triangulated surface made of the PEC is infinitesimally thin and the current density is assumed to be constant on the triangle, the volume integration is reduced to a surface summation. The contribution to the radiation vector from one triangle is

$$\mathbf{F}_{\theta i}(\theta, \phi) = \begin{bmatrix} \cos(\phi)\cos(\theta)J_{xi} + \sin(\phi)\cos(\theta)J_{yi} \\ -\sin(\theta)J_{zi} \end{bmatrix} e^{jk\mathbf{r}\cdot\mathbf{r}'} A_i \quad (12)$$

and

$$\mathbf{F}_{\phi i}(\theta, \phi) = \begin{bmatrix} -\sin(\phi)J_{xi} + \cos(\phi)J_{yi} \end{bmatrix} e^{jk\mathbf{r}\cdot\mathbf{r}'} A_i, \quad (13)$$

where A_i is the area of the i th triangle; J_{xi} , J_{yi} , and J_{zi} are the components of the current density \mathbf{J}_i on the triangle; and

$$\mathbf{r}\cdot\mathbf{r}' = [x_i \cos(\phi) + y_i \sin(\phi)] \sin(\theta) + z_i \cos(\theta). \quad (14)$$

The center of the i th triangle is specified by the Cartesian coordinates x_i , y_i , and z_i . The total radiation vector is next computed as a sum of contributions from all triangles:

$$\mathbf{F}_{\theta}(\theta, \phi) = \sum_i \mathbf{F}_{\theta i}(\theta, \phi), \quad (15)$$

$$\mathbf{F}_{\phi}(\theta, \phi) = \sum_i \mathbf{F}_{\phi i}(\theta, \phi). \quad (16)$$

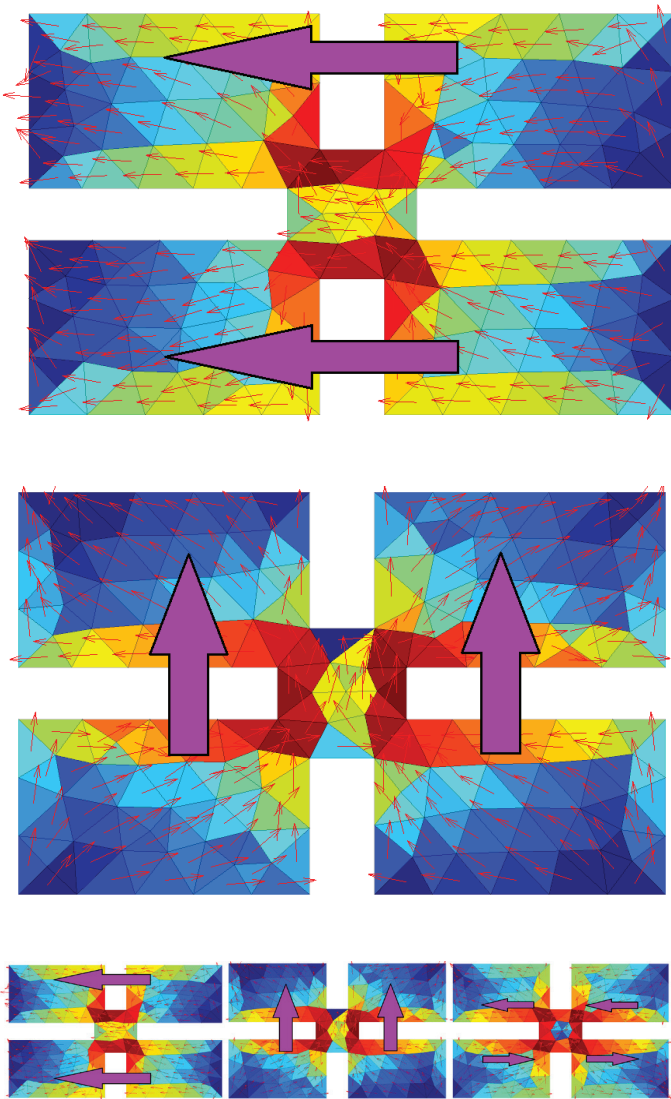


Figure 14. The modal currents (top to bottom: the first, second, and third modes at the resonance frequency).

Table 3. Far-field computations for the Minkowski fractal with different angle steps.

Angle Step	Angle Points	Max. Directivity	Time	Rad. Power
45°	45	2.6521	0.0082	0.9088
20°	190	2.3089	0.0121	0.9836
12°	496	2.2604	0.0194	0.9946
7.2°	1326	2.2433	0.0406	0.9986
4.5°	3321	2.2374	0.0915	1.0000
3°	7381	2.2354	0.1931	1.0004
1.5°	29161	2.2341	0.7478	1.0006
1°	65341	2.2339	1.6610	1.0008
0.5°	260281	2.2338	6.5763	1.0008

From the values of the radiation vector, the electric field in the far-field region, as well as the radiation intensity, \mathbf{U} , can be computed. To estimate the modal directivity pattern, the radiated power has to be known. It is obtained by numerically integrating \mathbf{U} over a sphere. As all the modal currents are normalized and the structure is supposed to be lossless, the integration should give a unit radiated power. The algorithm is very fast, even with high angular resolution (see Table 3).

The second approach is the dipole method. This method regards the current flowing along a mesh edge as an infinitesimal dipole. This current is a direct result of the MoM, and in the Theory of Characteristic Modes, these currents can be associated with characteristic vectors. The advantage of the dipole method is that analytical expressions for radiated fields \mathbf{E} and \mathbf{H} exist. These expressions were derived without any far-field approximation: they are therefore valid for an arbitrary distance larger than approximately the length of the dipole (edge) [27]. These expressions are thus suitable for the near-field calculations. The total fields at an arbitrary observation point are computed easily as the sum of contributions to \mathbf{E} and \mathbf{H} from all edges (i.e., dipoles).

The dipole method is more general, because it computes \mathbf{E} and \mathbf{H} fields at an arbitrary distance. On the other hand, the evaluation is approximately 15-20 times slower than the direct far-field computation.

Let us next consider a visualization of the computed fields. The modal radiation patterns are plotted by a *MATLAB* surface plot. Note that spherical coordinates should be mapped to appropriate Cartesian coordinates before the visualization.

The comparisons of computation times and computed values for different angle steps are in Table 3. Similar convergence was observed for different modes and different structures. It could be seen that both the maximum directivity and the radiated power converged with a smaller angle step. A 3° step was a reasonable compromise between accuracy and computation time. Examples of three-dimensional directivity plots and cuts are in Figures 15 and 16.

4.6 Near Field

As was mentioned in the previous section, the dipole method is valid for arbitrary distances (greater than the edge length). Near-field plots can therefore be created by computing fields at certain points lying on the surface of virtually any shape: for example, see the magnitude of the electric field on a plane in Figure 17.

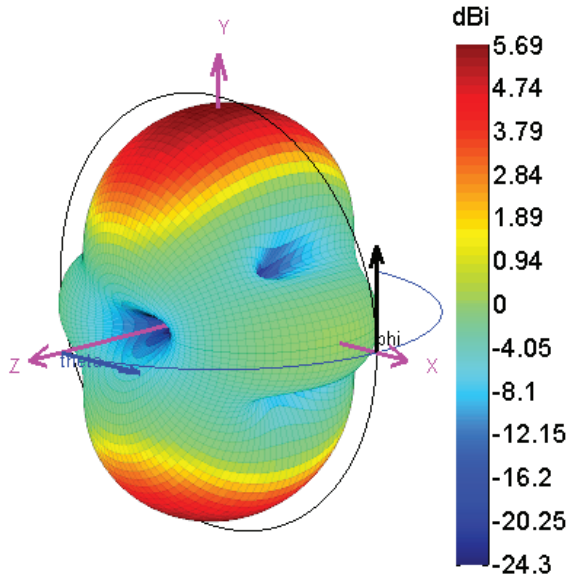
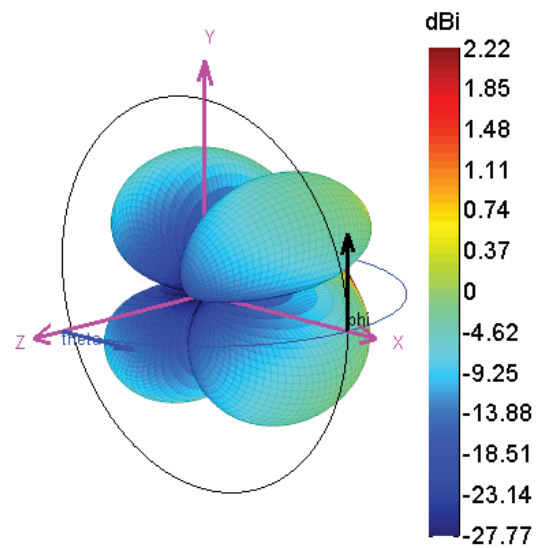
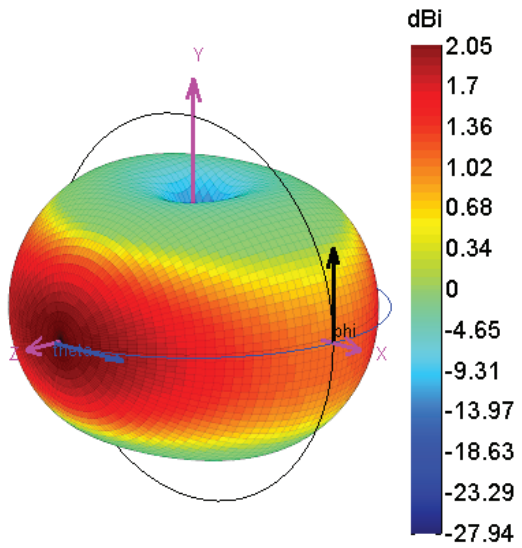
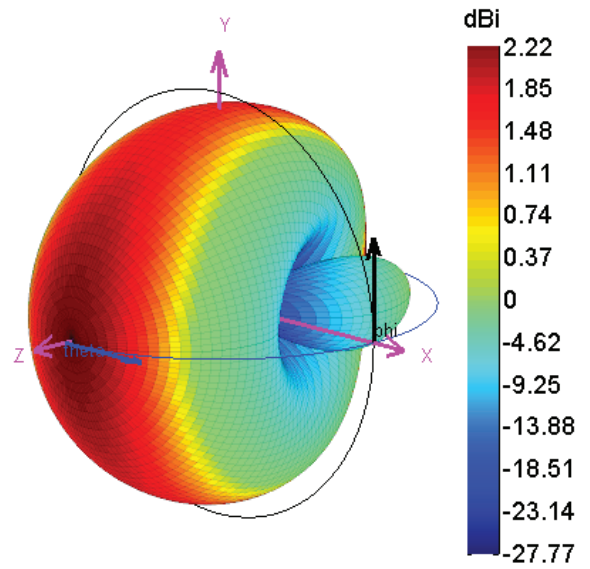
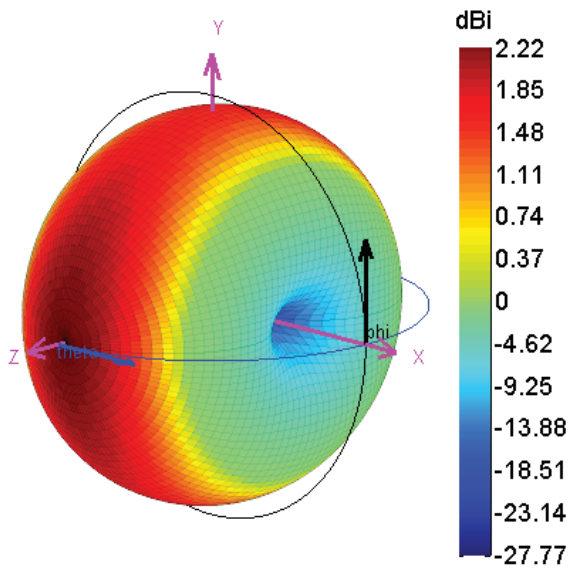


Figure 16. The radiation patterns for the first mode of the Minkowski fractal motif in free space: (top) the Ludwig3 horizontal component, (bottom) the Ludwig3 vertical component.

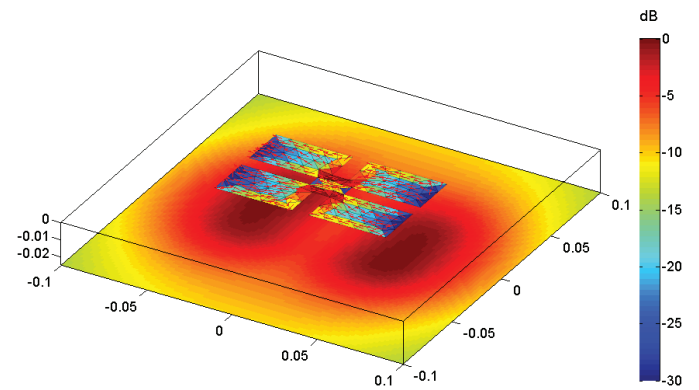


Figure 17. The normalized magnitude of the electric field for mode 1 on a plane $z = -25$ mm of the Minkowski fractal, shown on a logarithmic scale.

Figure 15. Radiation patterns for the first three modes for the Minkowski fractal motif in free space.

5. Closing Comments

The presented software can be used for educational purposes, as well as for effective antenna design. Nowadays, the major disadvantage is that the modal decomposition is time consuming. This issue can be treated by using a large number of high-speed processors. For this reason, the authors predict a growing interest in modal methods.

To demonstrate the usefulness of the Theory of Characteristic Modes, we concluded with two complex structures that were designed and manufactured with the help of the TCMapp software. Both structures were fractal shapes, depicted in Figure 20. While both of them were simulated in professional CST *MWS* software [30], our software gave the first (and very important) insight into their physical behavior. Antenna design is more effective based on this modal information. For details, we refer the reader to [31] and [21].

Together with the authors' other activities (in the fields of optimization, Delaunay triangularization, and electrically

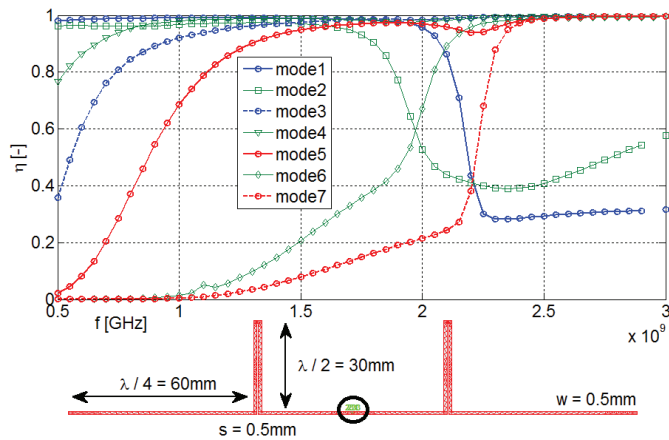


Figure 18. The modal radiation efficiency for the Franklin antenna: $\sigma = 5.85 \times 10^7 \text{ Sm}^{-1}$, $t = 50 \mu\text{m}$.

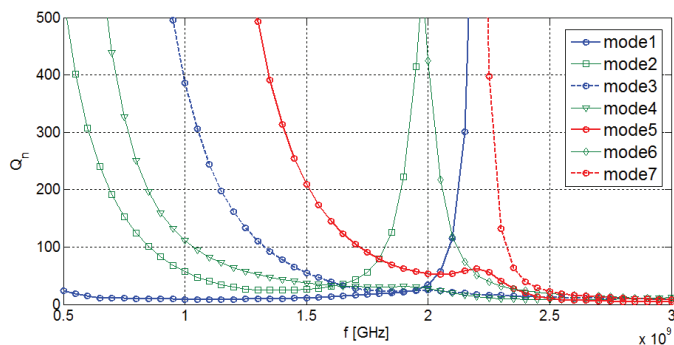


Figure 19. The modal radiation Q factor for the Franklin antenna.

4.7 Radiation Efficiency

While we consider the PEC for the Theory of Characteristic Modes decomposition, the modal radiation efficiency [5] can be estimated for good conductors (such as copper, aluminum, etc.). We demonstrated the results for the case of the Franklin antenna in the frequency range 0.5 GHz to 3 GHz (see Figure 18). Modal efficiencies were computed for the conductivity of copper, $\sigma = 5.85 \times 10^7 \text{ Sm}^{-1}$, and a metallization thickness of $t = 50 \mu\text{m}$.

4.8 Radiation Q Factor

Based on [28] and [29], the modal radiation Q factors can also be rigorously calculated [18]. The results for the first seven modes are depicted in Figure 19. The Franklin antenna from the previous section was used.

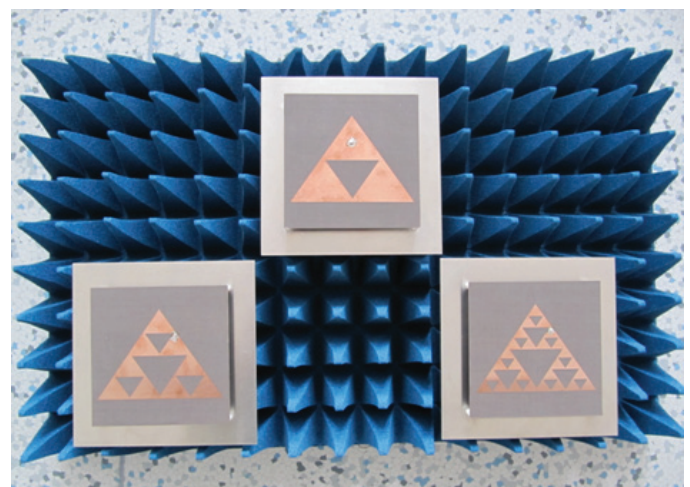
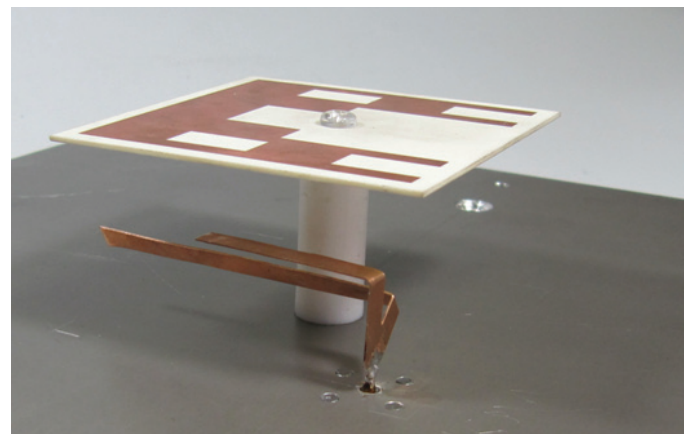


Figure 20. The manufactured fractal antennas that were analyzed with the Theory of Characteristic Modes software.

small antenna design), the *MATLAB* toolbox elements consisting of the new RWG MoM, the Theory of Characteristic Modes simulator with comprehensive post processing, the particle-swarm optimizer, the modal radiation Q factor, and the radiation-efficiency packages, are all scheduled towards the end of 2013. The MoM and the Theory of Characteristic Modes can be further generalized towards multilayered or dielectric structures, which can be the next challenge to interested readers and scientists in the field.

6. Acknowledgement

This work was supported by a project of the Czech Science Foundation, grant No. P102/12/2223, and by the Grant Agency of the Czech Technical University in Prague, grant No. SGS12/142/OHK3/2T/13. The authors also would like to thank N. Bell for his comments.

7. References

1. R. F. Harrington, *Field Computation by Moment Methods*, New York, Wiley-IEEE Press, 1993.

2. S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic Scattering by Surfaces of Arbitrary Shape," *IEEE Transactions on Antennas and Propagation*, **AP-30**, 3, May 1982, pp. 409-418.

3. R. F. Harrington and J. R. Mautz, "Theory of Characteristic Modes for Conducting Bodies," *IEEE Transactions on Antennas and Propagation*, **AP-19**, 5, September 1971, pp. 622-628.

4. M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry, Third Edition*, Berlin, Springer, 2008.

5. C. A. Balanis, *Advanced Engineering Electromagnetics*, New York, Wiley, 1989.

6. W. C. Gibson, *The Method of Moments in Electromagnetics*, Boca Raton, FL, Chapman and Hall/CRC, 2007.

7. P. Hazdra and P. Hamouz, "On the Modal Superposition Lying under the MoM Matrix Equations," *Radioengineering*, **17**, 3, September 2008, pp. 42-46.

8. M. Capek, P. Hazdra, and J. Eichler, "Complex Power-Ratio Functional for Radiating Structures with Applications to the Characteristic Mode Theory," submitted to *IEEE Transactions on Antennas and Propagation*.

9. The *MATLAB*, available at <http://www.mathworks.com>.

10. The MathWorks, *Object-Oriented Programming*, The MathWorks, Inc., 2012.

11. The Jacket, available at <http://www.accelereyes.com/>.

12. E. Anderson, et al., *LAPACK Users' Guide*, Philadelphia, PA, Society for Industrial and Applied Mathematics (SIAM), 1999.

13. *MATLAB* TCM application, available at www.old.elmag.org/doku.php/wiki:user:capek:tcmap.

14. S. N. Makarov, *Antenna and EM Modeling with MATLAB*, New York, John Wiley, 2002.

15. M. Capek, P. Hazdra, P. Hamouz, and J. Eichler, "A Method for Tracking Characteristic Numbers and Vectors," *Progress In Electromagnetics Research B*, **33**, 2011, pp. 115-134.

16. B. D. Raines and R. G. Rojas, "Wideband Characteristic Mode Tracking," *IEEE Transactions on Antennas and Propagation*, **AP-60**, 7, July 2012, pp. 3537-3541.

17. The MathWorks, *MATLAB Distributed Computing Server*, The MathWorks, Inc., 2011.

18. M. Capek, P. Hazdra, and J. Eichler, "A Method for the Evaluation of Radiation Q Based on Modal Approach," *IEEE Transactions on Antennas and Propagation*, **AP-60**, 10, October 2012, pp. 4556-4567.

19. J. L. Gustafson, "Reevaluating Amdahl's Law," *Communications of the ACM*, May 1988, pp. 532-533.

20. K. Falconer, *Fractal Geometry: Mathematical Foundations and Applications*, New York, John Wiley, 2003.

21. M. Capek, P. Hazdra, P. Hamouz, and M. Mazanek, "Software Tools for Efficient Generation, Modeling and Optimisation of Fractal Radiating Structures," *IET Microwaves, Antennas and Propagation*, **5**, 8, June 2011, pp. 1002-1007.

22. P. Hamouz, P. Hazdra, M. Polivka, M. Capek, and M. Mazanek, "Radiation Efficiency and Q Factor Study of Franklin Antenna Using the Theory of Characteristic Modes," Proceedings of the EuCAP 2011, Rome, Italy.

23. The MathWorks, *Partial Differential Equation Toolbox*, The MathWorks, Inc., 2012.

24. The Comsol Multiphysics, available: <http://www.comsol.com/>.

25. P.-O. Persson, *Mesh Generation for Implicit Geometries*, PhD Thesis, Cambridge, MA, MIT, 2005.

26. E. H. Newman, "Small Antenna Location Synthesis Using Characteristic Modes," *IEEE Transactions on Antennas and Propagation*, **AP-27**, 4, July 1979, pp. 530-531.

27. S. Makarov, "MoM Antenna Simulations with *MATLAB*: RWG Basis Functions," *IEEE Antennas and Propagation Magazine*, **43**, 5, October 2001, pp. 100-107.
28. G. A. E. Vandenbosch, "Reactive Energies, Impedance, and Q Factor of Radiating Structures," *IEEE Transactions on Antennas and Propagation*, **AP-58**, 4, April 2010, pp. 1112-1127.
29. P. Hazdra, M. Capek, and J. Eichler, "Radiation *Q*-Factors of Thin-Wire Dipole Arrangements," *IEEE Antennas and Wireless Propagation Letters*, **10**, May 2011, pp. 556-560.
30. Computer Simulation Technology, available at <http://www.cst.com/>.
31. J. Eichler, P. Hazdra, M. Capek, T. Korinek, and P. Hamouz, "Design of a Dual-Band Orthogonally Polarized L-Probe-Fed Fractal Patch Antenna Using Modal Methods," *IEEE Antennas and Wireless Propagation Letters*, **10**, May 2011, pp. 1389-1392. 