

HYPERFRÉQUENCES

---

## **Document annexe**

# **Manuel d'utilisation d'OpenEMS**

---

*Auteur :*  
Thomas LEPOIX

MASTER 1 Systèmes Embarqués

**E.S.T.E.I.**  
École Supérieure des Technologies Électronique, Informatique, et Infographie  
Département Systèmes Embarqués

14 mai 2018

# Table des matières

<b>Table des matières</b>	<b>1</b>
<b>A Manuel d'utilisation d'OpenEMS</b>	<b>2</b>
A.1 Déclaration des variables de la simulation . . . . .	2
A.2 Paramétrage de la simulation . . . . .	3
A.3 Définition de l'objet géométrique . . . . .	4
A.4 Définition du maillage . . . . .	4
A.5 Définition des dumps . . . . .	6
A.6 Aperçu de la géométrie et lancement de la simulation . . . . .	7
A.7 Traitement post-simulation . . . . .	10
A.7.1 Déclarations de variables . . . . .	10
A.7.2 Tracé de l'impédance d'entrée . . . . .	11
A.7.3 Tracé du coefficient de réflexion en entrée S11 . . . . .	12
A.7.4 Tracé du rapport d'ondes stationnaires ROS . . . . .	12
A.7.5 Tracé des diagrammes de rayonnement . . . . .	13
A.7.6 Tracé en 3D des diagrammes de rayonnement . . . . .	16
A.7.7 Sauvegarde des diagrammes générés . . . . .	17
A.7.8 Export de données dans un fichier texte . . . . .	18
A.7.9 Export d'un fichier Touchstone.s1p . . . . .	18
A.7.10 Tracé d'un abaque de Smith via Qucs . . . . .	18
A.8 Visualisation de l'évolution des champs avec Paraview . . . . .	19

## Annexe A

# Manuel d'utilisation d'OpenEMS

OpenEMS est un logiciel libre de simulation électromagnétique utilisant la méthode FDTD.

Contrairement à la plupart des logiciels de simulation électromagnétique commerciaux comme AWR, OpenEMS s'utilise à travers une interface texte en langage Matlab / GNU Octave. Cela le rend particulièrement puissant du fait de la liberté offerte à l'utilisateur, en revanche son interface peut paraître austère et nécessite une certaine maîtrise de ce dont il est question.

Son utilisation repose sur plusieurs logiciels :

- GNU Octave ou Matlab : Pour lancer le script associé à la simulation.
- CSXCAD : Une librairie utilisée pour décrire des objets géométriques et un maillage.
- appCSXCAD : Logiciel permettant de visualiser l'objet 3D précédemment décrit. Ne permet aucune modification de l'objet.
- OpenEMS : Moteur de calcul FDTD.
- NF2FF : Near Field To Far Field, programme calculant les rayonnements électromagnétiques de l'antenne à partir des champs rayonnés par chaque cellule du maillage.
- Paraview : Logiciel de visualisation 3D utilisé pour observer certains résultats de simulations en 3D et évoluant dans le temps ou la fréquence, par exemple le champ électrique rayonné. Il s'agit des fichiers dump au format .vtk.

Sites officiels des différents outils utilisés :

- OpenEMS : <http://openems.de/start/index.php>
- GNU Octave : <https://www.gnu.org/software/octave/>
- Paraview : <https://www.paraview.org/>
- Qucs : <http://qucs.sourceforge.net/>

## A.1 Déclaration des variables de la simulation

Il s'agit de placer dans des variables au début du script toutes les valeurs numériques que l'on va utiliser, à savoir la fréquence centrale de l'antenne, les dimensions des éléments, les caractéristiques du substrat, etc.

```
%%% SYSTEM VAR
suffix = "channel1"
Sim_Path = 'tmp';
```

```

Sim_CSX = 'patch.xml';
[status, message, messageid] = rmdir( Sim_Path, 's' );
[status, message, messageid] = mkdir( Sim_Path );

%%% VAR
f0 = 2.4e9;
fc = 1e9;

unit = 1e-3;

substrate.tand = 0.02;
substrate.Er = 4.7;
substrate.K = substrate.tand * 2 * pi * f0 * EPS0 * substrate.Er;
substrate.h = 1.55;
substrate.L = 80;
substrate.W = 80;
substrate.cells = 4;

patch.L = 28.5151;
patch.W = 37.0218;

line.L = 16.621;
line.W = 2.8892;

ground.L = substrate.L;
ground.W = substrate.W;

copper.h = 0.035;

feed.W = line.W;
feed.h = substrate.h;
feed.R = 50;

```

## A.2 Paramétrage de la simulation

Il s'agit de définir le type d'excitation utilisé, la fréquence centrale, ainsi que les conditions aux limites de la zone de simulation.

Comme expliqué dans le paragraphe concernant la méthode FDTD, plusieurs comportements aux limites de la zone simulée existent :

- PEC : Conducteur électrique parfait.
- PMC : Conducteur magnétique parfait.
- MUR (ABC) : Condition absorbante simple.
- PML (ABC) : Plan parfaitement adapté.

Notre antenne étant vouée à être en espace libre, on choisit un contours absorbant : MUR.

L'excitation gaussienne est une impulsion temporelle étendue en fréquence pour laquelle on définit une fréquence centrale et une fréquence de coupure. Elle convient dans notre cas.

```

tres = 300000;
FDTD = InitFDTD('NrTS', tres );
FDTD = SetGaussExcite( FDTD, f0, fc );
BC = {'MUR' 'MUR' 'MUR' 'MUR' 'MUR' 'MUR'};
FDTD = SetBoundaryCond( FDTD, BC );

```

Remarque : 300000 correspond au nombre d'étapes temporelles avant que la simulation ne se stoppe, terminée ou non.

### A.3 Définition de l'objet géométrique

Il s'agit d'entrer dans le logiciel les dimensions dans l'espace du substrat, du plan de masse et du patch. Pour cela on définit des matériaux CSX pour chaque objet, puis des objets CSX. On définit également le point d'excitation. L'antenne à structure basique servira d'exemple :

```
CSX = InitCSX();

%%% SHAPE
CSX = AddMetal(CSX, 'Ground' );
CSX = AddMetal(CSX, 'Patch');
CSX = AddMetal(CSX, 'Line');
CSX = AddMaterial(CSX, 'Substrate' );
CSX = SetMaterialProperty(CSX, 'Substrate', 'Epsilon', substrate.Er, 'Kappa', substrate.K);

CSX = AddBox(CSX, 'Substrate', 1, ...
    [-substrate.W/2, -substrate.L/2, -substrate.h], ...
    [substrate.W/2, substrate.L/2, 0]);
CSX = AddBox(CSX, 'Ground', 2, ...
    [-ground.W/2, -ground.L/2, -substrate.h], ...
    [ground.W/2, ground.L/2, -substrate.h-copper.h]);
CSX = AddBox(CSX, 'Line', 2, ...
    [-line.W/2, -patch.L/2, 0], ...
    [line.W/2, -patch.L/2-line.L, copper.h]);
CSX = AddBox(CSX, 'Patch', 2, ...
    [-patch.W/2, -patch.L/2, 0], ...
    [patch.W/2, patch.L/2, copper.h]);

%%% PORT
start = [-feed.W/2, -patch.L/2-line.L+0.1, 0];
stop = [feed.W/2, -patch.L/2-line.L-0.1, -substrate.h];
[CSX port] = AddLumpedPort(CSX, 5, 1, feed.R, start, stop, [0 0 1], true);
```

Remarque : Une note sur le calcul de champs lointain à partir des champs proches (NF2FF) dit qu'il est judicieux de placer le centre de l'élément rayonnant de l'antenne au point [0;0;0] du repère.

### A.4 Définition du maillage

Contrairement à beaucoup de logiciels, avec OpenEMS le maillage est manuel. Et c'est ici qu'est tout le challenge d'une simulation : c'est avant tout le maillage qui va déterminer la fiabilité d'une simulation.

La documentation du logiciel explique que pour épargner du temps de calcul, il est judicieux de définir un maillage inhomogène, c'est à dire avec une résolution plus précise *mres* pour les zones métalliques (le patch, la ligne d'alimentation, etc.) et une résolution plus grossière *sres* pour couvrir le substrat, où moins de champs ne circulent. On utilise généralement des divisions de la longueur d'onde comme résolution. Dans cet exemple :

- $sres = \lambda/30$
- $mres = \lambda/60$

Il est également précisé que pour une meilleure précision, il vaut mieux respecter la règle suivante : Sur un bord de piste, il faut définir une ligne de maillage à  $1/3mres$  à l'intérieur, coté piste et une à  $2/3mres$  à l'extérieur, coté substrat.

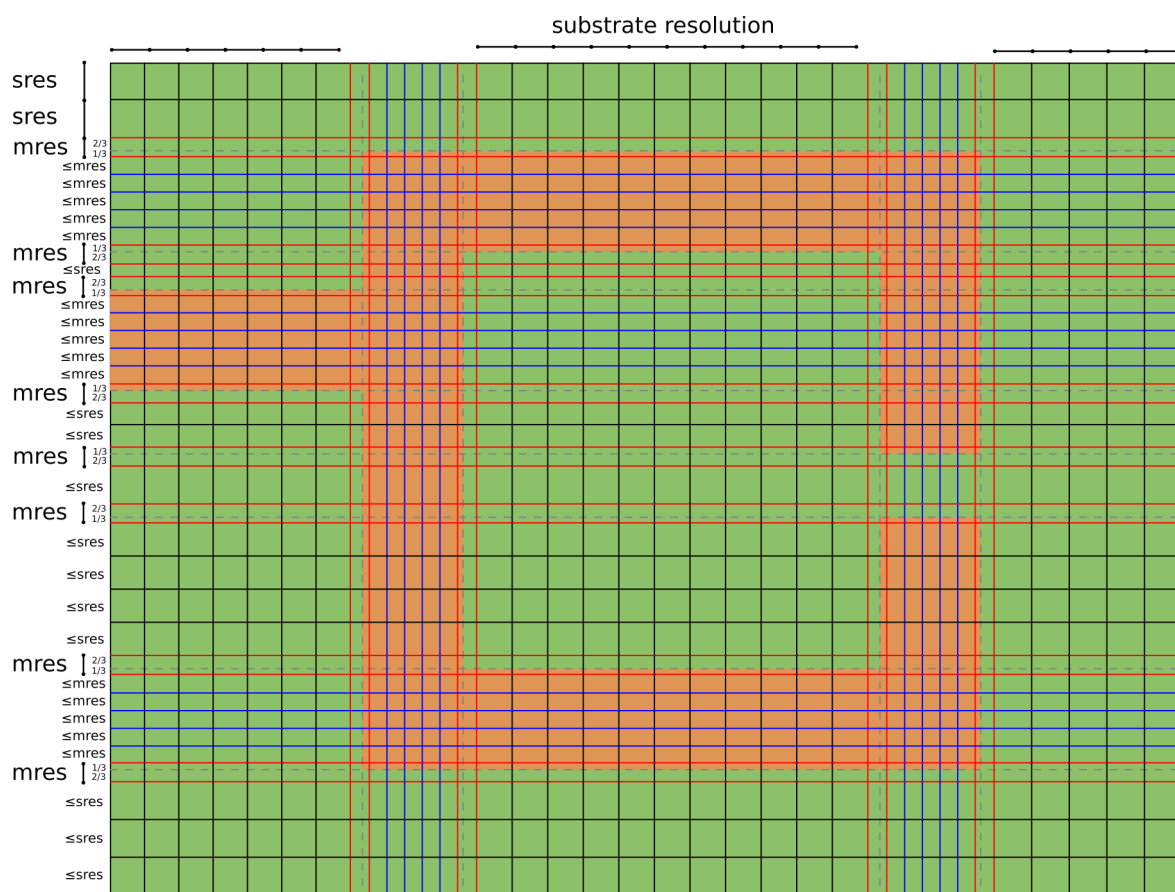


FIGURE A.1 – Représentation du maillage dans le plan du substrat avec en rouge les lignes  $1/3$ ;  $2/3$ , en bleu les lignes à la résolution  $mres$ , en noir les lignes à la résolution  $sres$

Remarque : Il semble être nécessaire que des lignes de maillage passent à travers la source d'excitation de l'antenne.

OpenEMS dispose de quelques fonctions permettant de générer un maillage adouci, sans variation brusque de la résolution.

Avec la fonction *SmoothMesh()*, la technique consiste à placer manuellement les lignes particulières aux environs des interfaces, aux bord du substrat ou à travers le port d'alimentation, puis à utiliser la fonction qui va générer un maillage adouci à la résolution spécifiée dans l'intervalle formé par les lignes manuelles les plus éloignées.

On répète l'opération une fois pour chaque résolution utilisée.

Déterminer les résolutions de maillage  $sres$  et  $mres$  qui conviennent peut s'avérer hasardeux et peut nécessiter plusieurs essais suivis de contrôles à l'aide de appCSXCAD. Il

arrive que les fonctions de maillage adoucissent produisent des résultats étranges, comme un très grand nombre de lignes dans un espace restreint ou des divisions par zéro lors de leurs calculs. Lorsque cela se produit, mieux vaut tenter une autre résolution.

Il faut également définir la zone prise en compte par le calcul des champs lointains (NF2FF), généralement plus petite que la zone de simulation.

```

%% VAR
SimBox = [120, 120, 80];
mdiv = 60;
sdiv = 30;
mres = c0 / (f0 + fc) / unit / mdiv;
sres = c0 / (f0 + fc) / unit / sdiv;

%% MESH
mesh.x = [ ...
    (-substrate.W/2) ...
    (substrate.W/2) ...
    (-feed.W/2)-(2*mres/3) ...
    (-feed.W/2)+(mres/3) ...
    (feed.W/2)+(2*mres/3) ...
    (feed.W/2)-(mres/3) ...
    (-patch.W/2)-(2*mres/3) ...
    (-patch.W/2)+(mres/3) ...
    (patch.W/2)+(2*mres/3) ...
    (patch.W/2)-(mres/3) ...
    (0) ...
];
mesh.y = [ ...
    (-substrate.L/2) ...
    (substrate.L/2) ...
    (-patch.L/2)-(2*mres/3) ...
    (-patch.L/2)+(mres/3) ...
    (patch.L/2)+(2*mres/3) ...
    (patch.L/2)-(mres/3) ...
    (-patch.L/2-line.L) ...
    (0) ...
];
mesh.z = linspace(-substrate.h-copper.h, copper.h, substrate.cells);
mesh = SmoothMesh(mesh, mres);

mesh.x = [mesh.x, -SimBox(1)/2, SimBox(1)/2];
mesh.y = [mesh.y, -SimBox(2)/2, SimBox(2)/2];
mesh.z = [mesh.z, -SimBox(3)/2, SimBox(3)/2];
mesh = SmoothMesh(mesh, sres);

CSX = DefineRectGrid( CSX, unit, mesh );

%% NF2FF
start = [mesh.x(10), mesh.y(10), mesh.z(10)];
stop = [mesh.x(end-9), mesh.y(end-9), mesh.z(end-9)];
[CSX nf2ff] = CreateNF2FFBox(CSX, 'nf2ff', start, stop);

```

## A.5 Définition des dumps

Cette étape est intéressante. Il s'agit de définir une zone dans laquelle les champs calculés seront sauvegardés dans des fichiers .vtk.

Douze dumps différents sont disponibles :

- $\vec{E}_{(t)}$  : Champs électrique en fonction du temps.
- $\vec{H}_{(t)}$  : Champs magnétiques en fonction du temps.
- $\vec{J}_{(t)}$  : Courants électriques en fonction du temps.

- $\vec{rot}(\vec{H}_{(t)})$  : Densité de courant en fonction du temps.
- $\vec{E}_{(f)}$  : Champs électrique en fonction de la fréquence.
- $\vec{H}_{(f)}$  : Champs magnétiques en fonction de la fréquence.
- $\vec{J}_{(f)}$  : Courants électriques en fonction de la fréquence.
- $\vec{rot}(\vec{H}_{(f)})$  : Densité de courant en fonction de la fréquence.
- SAR local en fonction de la fréquence.
- Moyenne SAR 1g en fonction de la fréquence.
- Moyenne SAR 10g en fonction de la fréquence.
- Données brutes permettant de calculer le SAR (champs électriques, volume de cellule, conductivité et densité). Le Specific Absorption Rate, Débit d'Absorption Spécifique en français, indique une quantité d'énergie électromagnétique absorbée par un tissu organique. Son unité est le  $W/kg$ .

On choisit ici de conserver les quatre champs temporels sur le plan de la surface du patch, sur une zone de la taille du substrat.

```
start = [-substrate.W/2, -substrate.L/2, copper.h];
stop = [ substrate.W/2,  substrate.L/2, copper.h];

%%% DUMP ET
CSX = AddDump(CSX, 'Et');
CSX = AddBox(CSX, 'Et', 0, start, stop);

%%% DUMP HT
CSX = AddDump(CSX, 'Ht', 'DumpType', 1);
CSX = AddBox(CSX, 'Ht', 0, start, stop);

%%% DUMP CURRENT
CSX = AddDump(CSX, 'Jt', 'DumpType', 2, 'DumpMode', 0);
CSX = AddBox(CSX, 'Jt', 0, start, stop);

%%% DUMP CURRENT DENSITY
CSX = AddDump(CSX, 'Cdt', 'DumpType', 3);
CSX = AddBox(CSX, 'Cdt', 0, start, stop);
```

## A.6 Aperçu de la géométrie et lancement de la simulation

Il s'agit d'appeler le logiciel appCSXCAD pour visualiser l'ensemble des objets créés. Aucune modification n'est possible via ce logiciel.

Puis de lancer le calculateur FDTD OpenEMS.

```
%%% RUN SIMU
WriteOpenEMS([Sim_Path '/' Sim_CSX], FDTD, CSX);
CSXGeomPlot([Sim_Path '/' Sim_CSX]);
RunOpenEMS( Sim_Path, Sim_CSX);
```



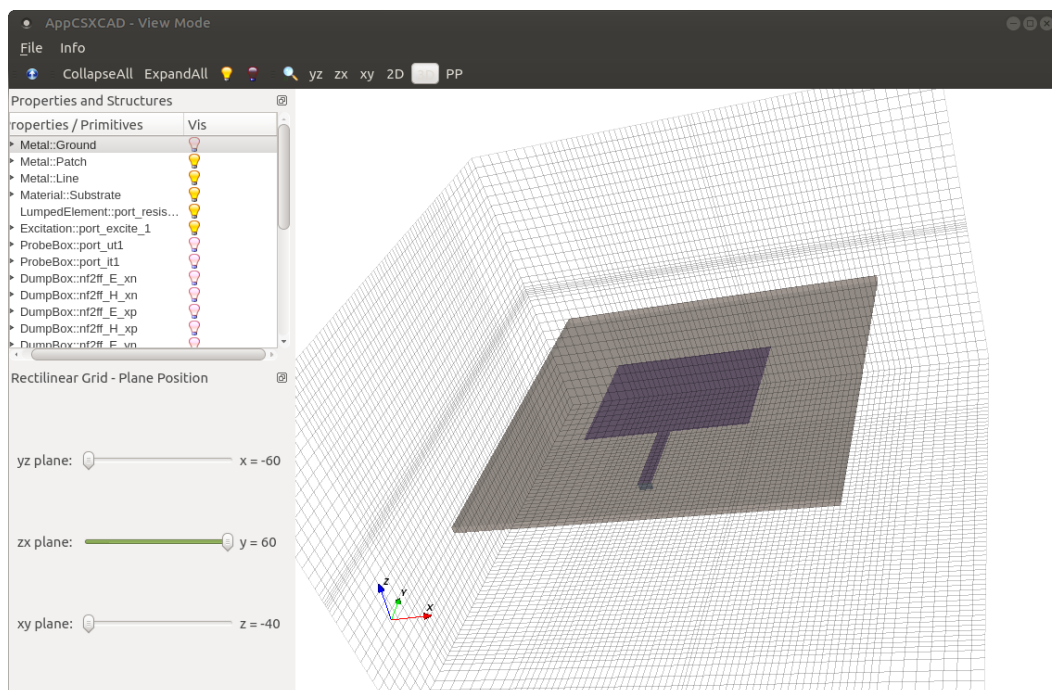


FIGURE A.2 – appCSXCAD : Visualisation de la géométrie décrite

Le logiciel dispose d'un menu permettant de sélectionner les objets que l'on affiche (en cliquant sur l'ampoule jaune/rose) et de changer leur couleur d'affichage (par un clic droit sur le nom de l'objet). On peut ainsi faire apparaître un plan de dump par exemple.

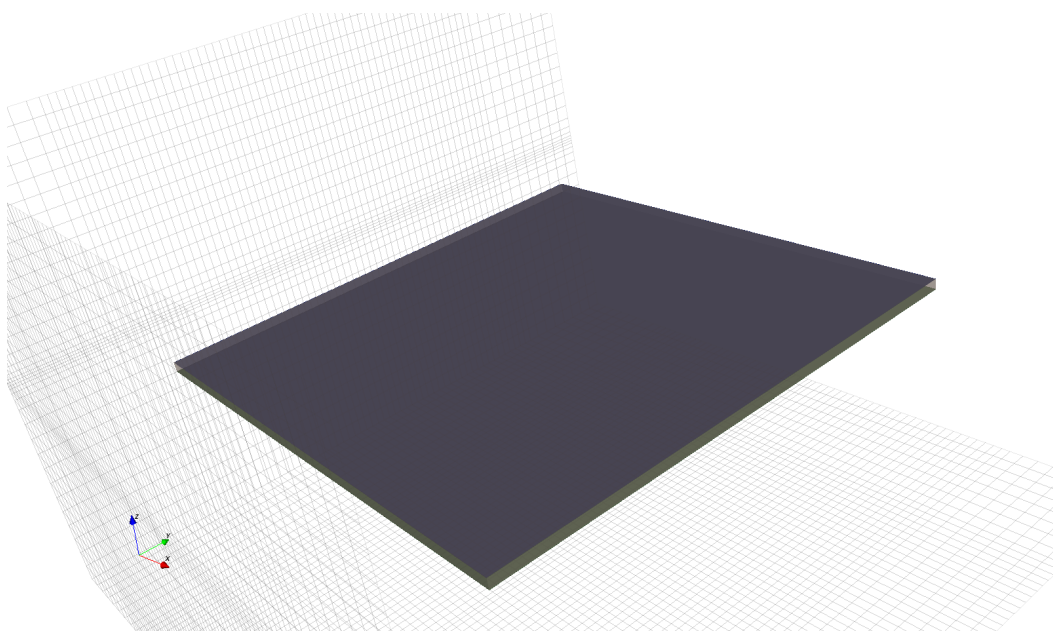


FIGURE A.3 – appCSXCAD : Visualisation du plan de dump, masquant le patch et sa ligne d'alimentation

Le logiciel dispose également d'un menu concernant le mode d'affichage, c'est à dire le nombre de dimensions ou le plan d'affichage. Afin d'observer le maillage, il est utile de passer en 2D pour supprimer la perspective.

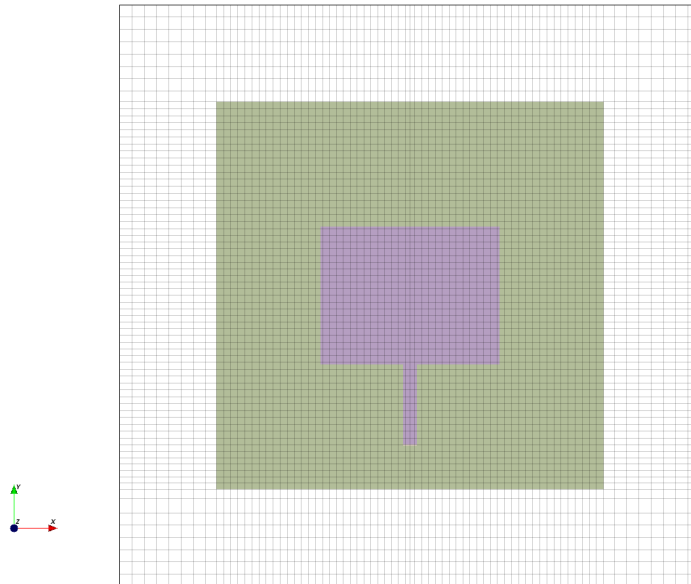


FIGURE A.4 – appCSXCAD : Visualisation en 2D du maillage inhomogène

On observe clairement deux résolutions différentes, une pour l'ensemble des objets, l'autre pour les espaces vides. Cela diffère légèrement de la règle préconisée mais permet tout de même de réduire les temps de simulation.

On observe également un resserrement du maillage au niveau de la ligne d'alimentation et parfois au niveau des interfaces.

En zoomant davantage on peut observer le respect de la règle  $1/3 ; 2/3$  sur le contour de l'antenne.

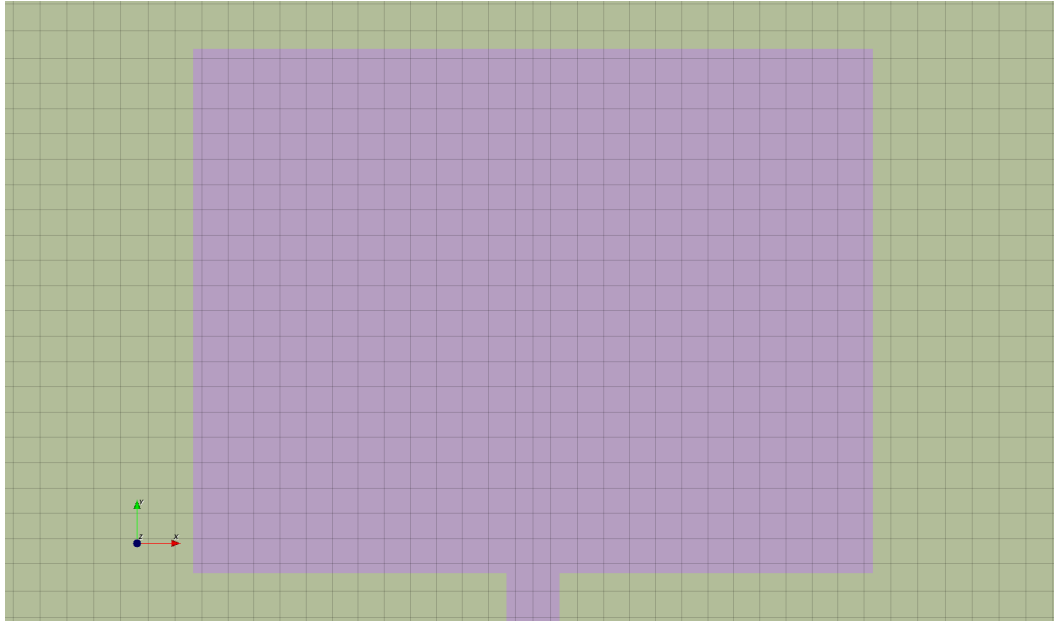


FIGURE A.5 – appCSXCAD : Visualisation en 2D du maillage selon la règle 1/3;2/3

## A.7 Traitement post-simulation

Il s'agit là de traiter les données calculées. Pour cela l'interface Matlab / GNU Octave entre en jeu et demeure appréciable pour sa puissance.

### A.7.1 Déclarations de variables

```
%%% VAR
freq = linspace( max([1e9,f0-fc]), f0+fc, 501 );
port = calcPort(port, Sim_Path, freq);
Zin = port.uf.tot ./ port.if.tot;
s11 = port.uf.ref ./ port.uf.inc;
P_in = 0.5 * port.uf.inc .* conj( port.if.inc );
Pin_f0 = interp1(freq, port.P_acc, f0);

f_res_ind = find(s11==min(s11));
f_res = freq(f_res_ind);
```

### A.7.2 Tracé de l'impédance d'entrée

On peut tracer trois courbes de l'impédance d'entrée :

- Son module.
- Sa partie réelle, c'est à dire sa résistance.
- Sa partie imaginaire, c'est à dire sa réactance.

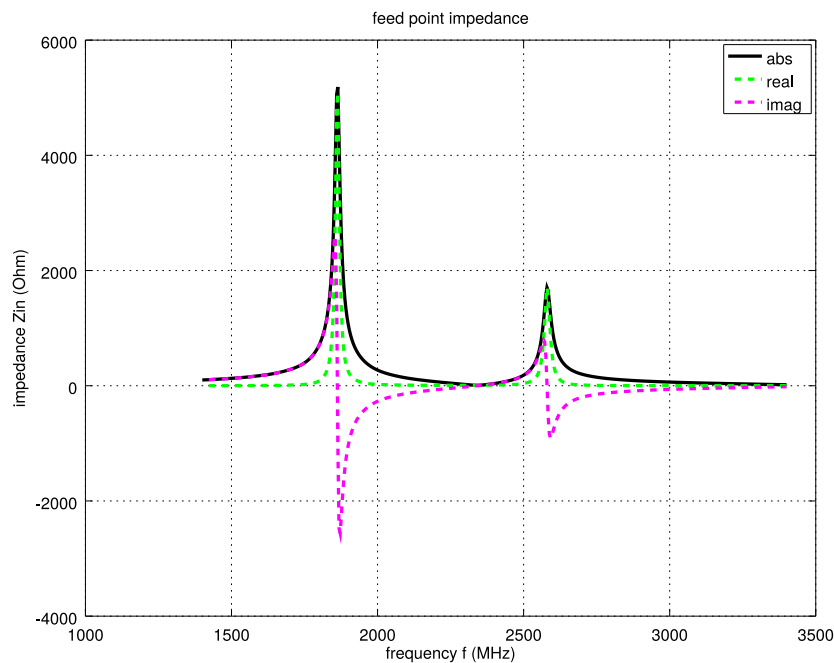
On peut également afficher dans le terminal la valeur de l'impédance pour la fréquence centrale de l'antenne.

```

%%% ZIN
figure
plot( freq/1e6, abs(Zin), 'k-', 'Linewidth', 2 );
hold on
grid on
plot( freq/1e6, real(Zin), 'g--', 'Linewidth', 2 );
hold on
plot( freq/1e6, imag(Zin), 'm--', 'Linewidth', 2 );
title( 'feed point impedance' );
xlabel( 'frequency f (MHz)' );
ylabel( 'impedance Zin (Ohm)' );
legend( 'abs', 'real', 'imag', );

disp( ['->   Zin @ ', num2str(f_res/1e9), 'GHz = ', num2str(Zin(f_res_ind)), ] );
disp( ['->   |Zin| @ ', num2str(f_res/1e9), 'GHz = ', num2str(abs(Zin(f_res_ind))), ' Ohm' ] );

```



```

->   Zin @ 2.36GHz = 9.2486+10.491i
->   |Zin| @ 2.36GHz = 13.986 Ohm

```

FIGURE A.6 – Impédance d'entrée de l'antenne

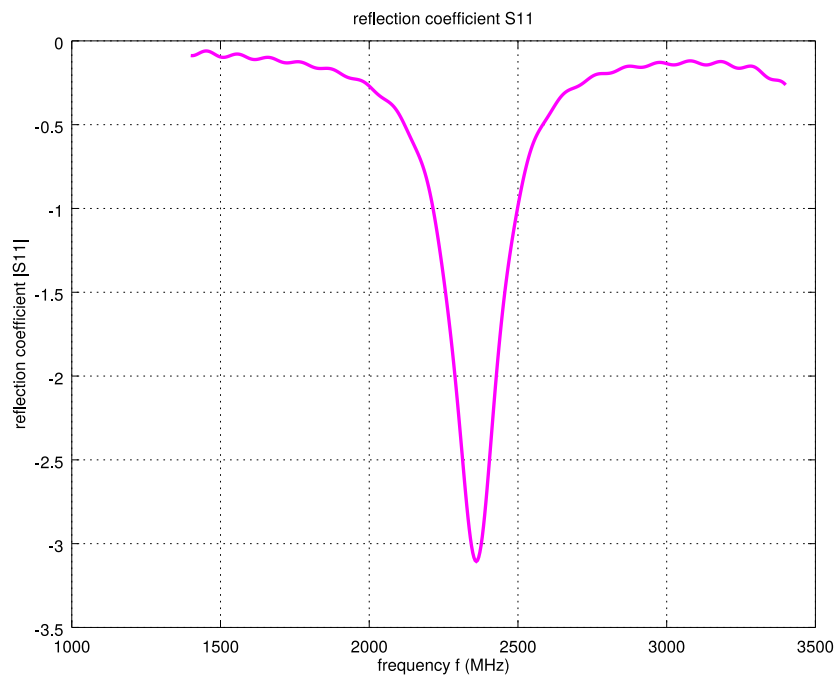
### A.7.3 Tracé du coefficient de réflexion en entrée $S_{11}$

```

%%% S11
figure
plot( freq/1e6, 20*log10(abs(s11)), 'm-', 'Linewidth', 2 );
grid on
title( 'reflection coefficient S11' );
xlabel( 'frequency f (MHz)' );
ylabel( 'reflection coefficient |S11|' );
drawnow

disp( ['->   S11 @ ', num2str(f_res/1e9), 'GHz = ', num2str(20*log10(abs(min(s11)))), ' dB'] );

```



```

->   S11 @ 2.36GHz = -3.1061 dB

```

FIGURE A.7 – Coefficient de réflexion en entrée  $S_{11}$

### A.7.4 Tracé du rapport d'ondes stationnaires ROS

Le ROS est une grandeur que l'on peut calculer de plusieurs façon : à partir d'une tension, d'un courant, d'une puissance, etc. La notation  $VSWR$ ,  $VROS$  en français, indique que le ROS est calculé à partir d'une tension.

```

%%% ROS
sigmaf = abs(s11);
sigmap = 1+(sigmaf);
sigmam = 1-(sigmaf);

figure
plot( freq/1e6, sigmap./sigmam, 'g-', 'Linewidth', 2 );
grid on
title( 'VSWR' );
xlabel( 'frequency f (MHz)' );
ylabel( 'VSWR' );
drawnow

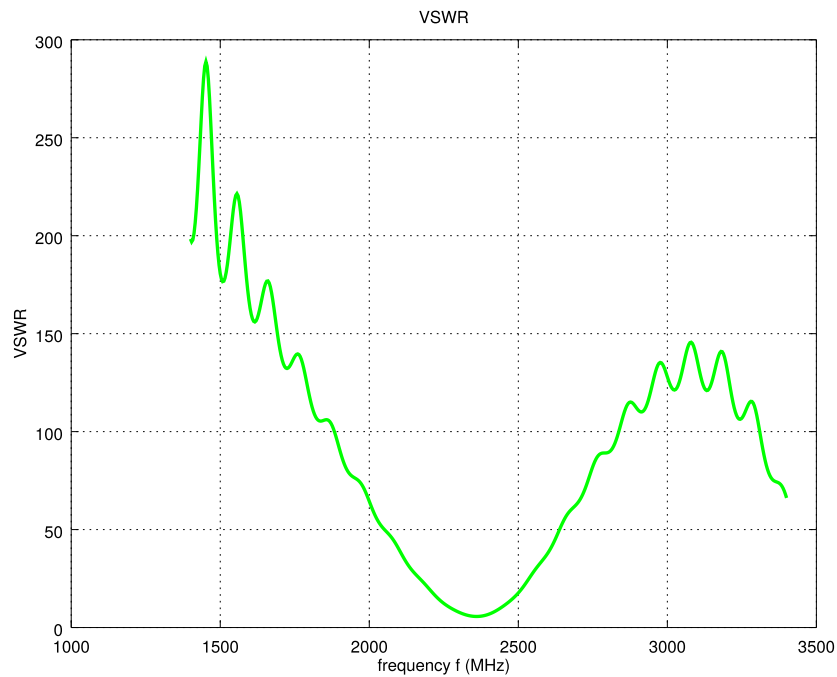
```

```

sigmaf = abs(s11(f_res_ind));
sigmap = 1+(sigmaf);
sigmam = 1-(sigmaf);

disp( ['->    VSWR @ ', num2str(f_res/1e9), 'GHz = ', num2str(sigmap./sigmam) ]);

```



```

->    VSWR @ 2.36GHz = 5.6523

```

FIGURE A.8 – Rapport d'ondes stationnaires

### A.7.5 Tracé d'un abaque de Smith

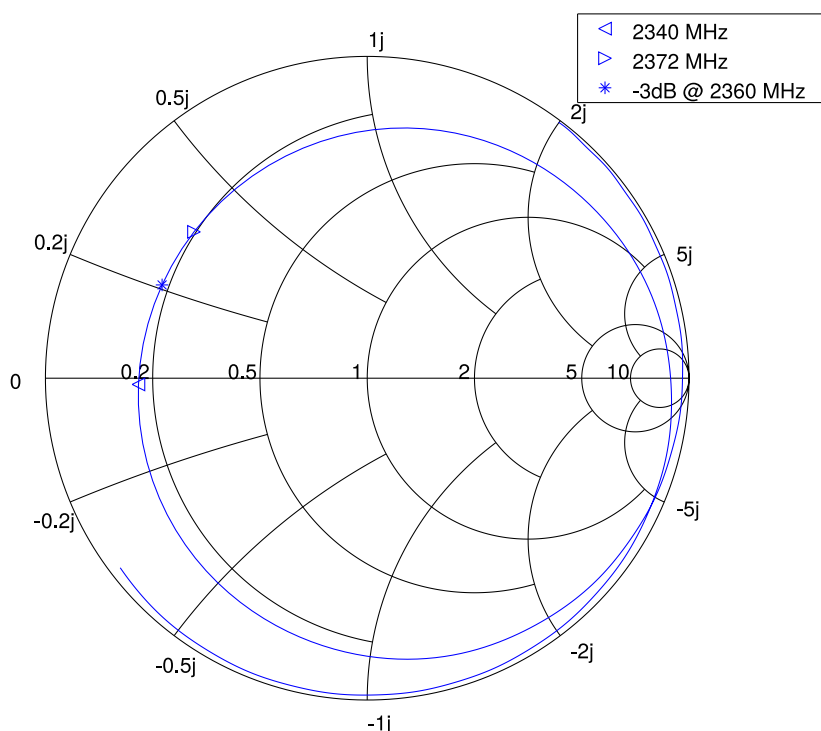
OpenEMS dispose d'une fonction permettant de tracer le coefficient de réflexion en entrée  $S_{11}$  sur un abaque de Smith rudimentaire. Cet abaque a le défaut d'être imprécis sur la valeur du coefficient de réflexion (au  $dB$  près) et de ne pas afficher d'impédances sur les marqueurs.

Le marqueur étoile indique la valeur minimale de  $S_{11}$ . Les marqueurs triangles indiquent les points à  $-3dB$  de part et d'autre du pic. Cette valeur est paramétrable.

```

%%% SMITH
plotRefl(port);
drawnow

```

FIGURE A.9 – Paramètre  $S_{11}$  affiché sur un abaque de Smith

### A.7.6 Tracé des diagrammes de rayonnement

Le calcul de la directivité et le tracé des diagrammes de rayonnement implique un calcul de conversion champs proches / champ lointain.

Ci-dessous le repère utilisé pour décrire le rayonnement en champ lointain. L'antenne étant inscrite dans le plan  $o; \vec{x}; \vec{y}$ , l'angle  $\varphi$  évolue dans ce plan tandis que  $\theta$  évolue dans un plan orthogonal à celui ci.

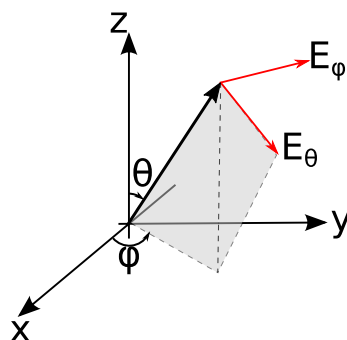


FIGURE A.10 – Repère spatial utilisé pour décrire le rayonnement

On peut afficher des diagrammes de rayonnement de plusieurs façons :

- En linéaire, en dB ou normalisé.

- Sur diagramme orthogonal ou circulaire.

En plus des diagrammes de rayonnement, on peut calculer, la puissance rayonnée, la directivité maximale et l'efficacité de l'antenne.

```

%% NF2FF
% calculate the far field at phi=0 degrees and at phi=90 degrees
disp( 'calculating far field at phi=[0 90] deg...' );
nf2ff = CalcNF2FF(nf2ff, Sim_Path, f_res, [-180:2:180]*pi/180, [0 90]*pi/180);

%% DIRECTIVITY
disp( [' radiated power: Prad = ' num2str(nf2ff.Prad) ' Watt'] );
disp( [' directivity: Dmax = ' num2str(nf2ff.Dmax) ' ( ' num2str(10*log10(nf2ff.Dmax)) ' dBi)'] );
disp( [' efficiency: nu_rad = ' num2str(100*nf2ff.Prad./real(P_in(f_res_ind))) ' %'] );

%% POLAR dB NORMALISE
figure
polarFF(nf2ff,'xaxis','theta','param',[1 2],'normalize',1,'xtics',10)
drawnow
%% POLAR dBi
figure
polarFF(nf2ff,'xaxis','theta','param',[1 2],'logscale',[-20 10], 'xtics', 10);
drawnow
%% RECT dBi
figure
plotFFdB(nf2ff,'xaxis','theta','param',[1 2])
drawnow

```

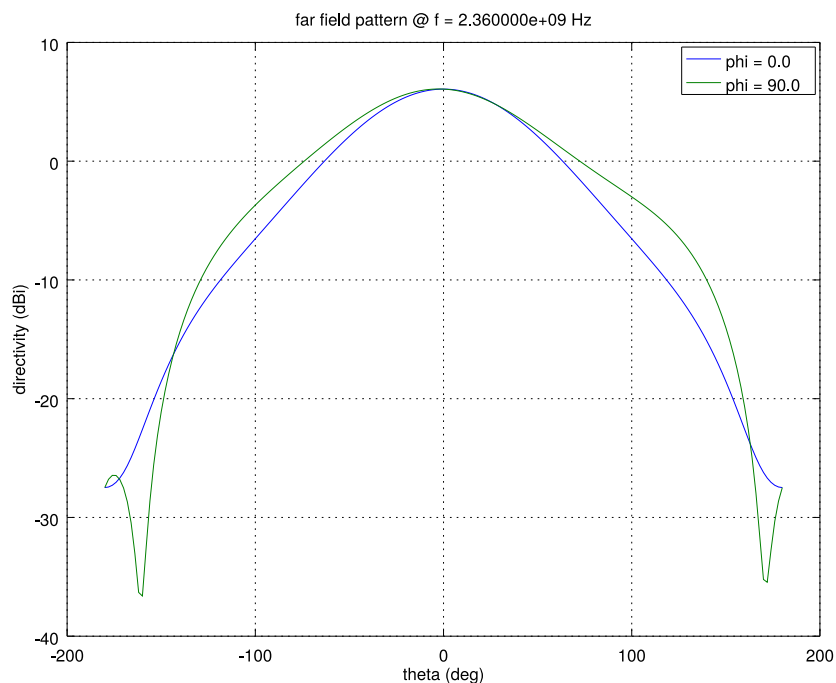


FIGURE A.11 – Directivité en *dBi* sur diagramme orthogonal



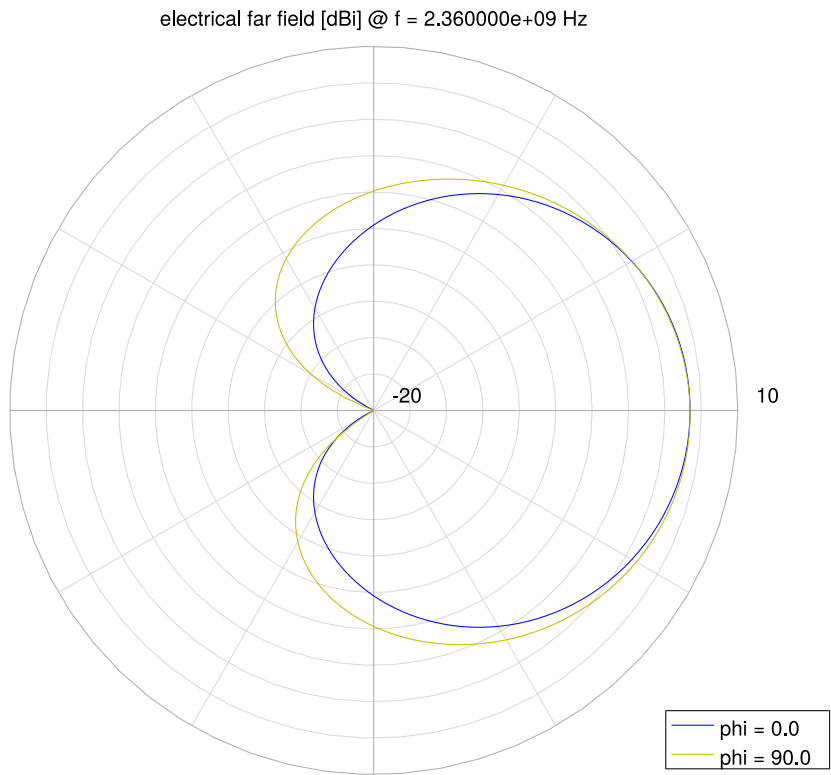


FIGURE A.12 – Directivité en *dBi* sur diagramme polaire

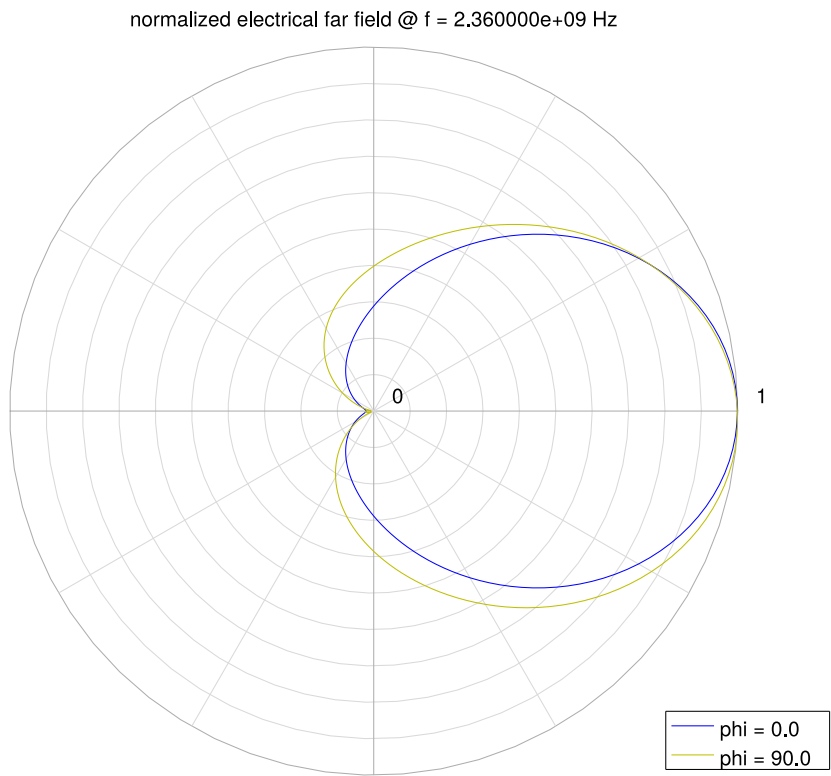


FIGURE A.13 – Directivité normalisée sur diagramme polaire

```
-> radiated power: Prad = 5.1081e-26 Watt
-> directivity: Dmax = 4.0541 (6.079 dBi)
-> efficiency: nu_rad = 46.724 %
```

FIGURE A.14 – Affichage dans le terminal de la puissance rayonnée, de la directivité maximale et de l'efficacité de l'antenne

### A.7.7 Tracé en 3D des diagrammes de rayonnement

Il est également possible d'afficher des diagrammes de rayonnement en 3D. Cela nécessite un calcul de champ lointain plus complet que le précédent.

```
%%% 3D dB
disp( 'calculating 3D far field pattern and dumping to vtk (use Paraview to visualize)...' );
thetaRange = (0:2:180);
phiRange = (0:2:360) - 180;
nf2ff = CalcNF2FF(nf2ff, Sim_Path, f_res, thetaRange*pi/180, phiRange*pi/180, 'Verbose', 1, 'Outfile', '3D_Pattern.h5');
figure
plotFF3D(nf2ff, 'logscale', -20);
E_far_normalized = nf2ff.E_norm{1} / max(nf2ff.E_norm{1}(:)) * nf2ff.Dmax;
DumpFF2VTK([Sim_Path '3D_Pattern.vtk'], E_far_normalized, thetaRange, phiRange, 'scale', 1e-3);
drawnow

%%% 3D V/m
phiRange = sort( unique( [-180:5:-100 -100:2.5:-50 -50:1:50 50:2.5:100 100:5:180] ) );
thetaRange = sort( unique( [ 0:1:50 50:2.:100 100:5:180 ] ) );
disp( 'calculating 3D far field...' );
nf2ff = CalcNF2FF(nf2ff, Sim_Path, f_res, thetaRange*pi/180, phiRange*pi/180, 'Verbose', 2, 'Outfile', 'nf2ff_3D.h5');
figure
plotFF3D(nf2ff);
drawnow
```

L'on obtient une fenêtre avec un diagramme en 3D que l'on peut manipuler dans l'espace.

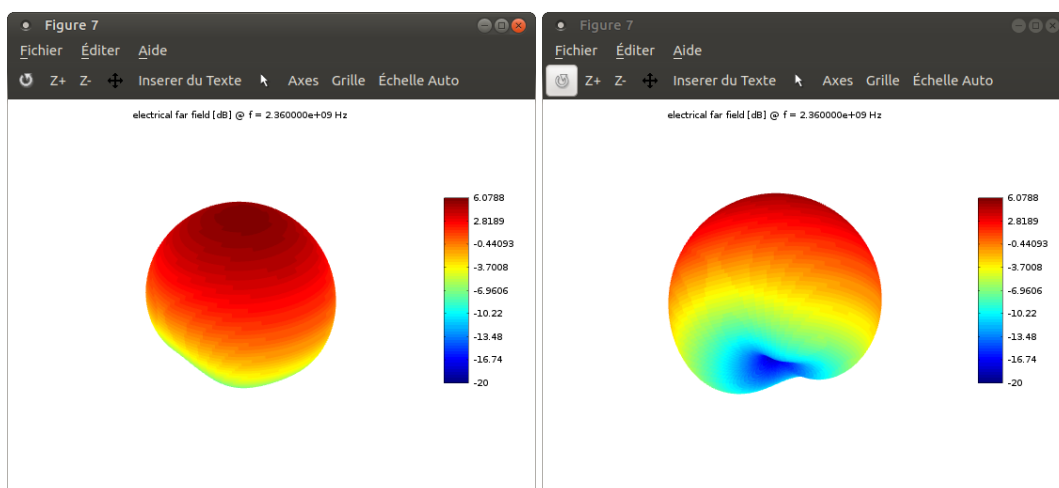
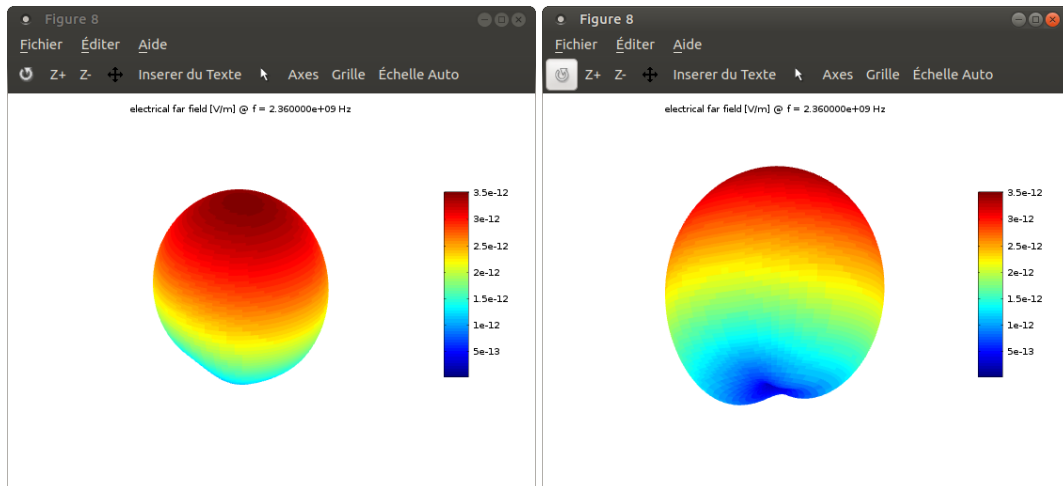


FIGURE A.15 – Directivité en dB dans l'espace

FIGURE A.16 – Directivité en  $V/m$  dans l'espace

### A.7.8 Sauvegarde des diagrammes générés

La commande Octave `print()` permet d'exporter un graphique.

```
%%% SAVE PLOTS
print(1, ["Zin_", suffix, ".pdf"]);
print(2, ["S11_", suffix, ".pdf"]);
print(3, ["VSWR_", suffix, ".pdf"]);
print(4, ["smith_", suffix, ".pdf"]);
print(5, ["dir_polar_dB_norm_", suffix, ".pdf"]);
print(6, ["dir_polar_dB_", suffix, ".pdf"]);
print(7, ["dir_dB_", suffix, ".pdf"]);
print(8, ["farfield_3D_dB_", suffix, ".png"]);
print(9, ["farfield_3D_Vm_", suffix, ".png"]);
```

### A.7.9 Export de données dans un fichier texte

Il est possible de sauvegarder dans un fichier texte les données que l'on affiche dans le terminal avec la commande Octave `fdisp()`.

```
%%% SAVE DATA
FID = fopen([suffix, '.txt'], 'w');
fdisp(FID, ['tres = ', num2str(tres)]);
fdisp(FID, ['mres = lambda/', num2str(mdiv), ' = ', num2str(mres), 'mm']);
fdisp(FID, ['sres = lambda/', num2str(sdiv), ' = ', num2str(sres), 'mm']);
fdisp(FID, ['']);
fdisp(FID, ['-> S11 @ ', num2str(f_res/1e9), 'GHz = ', num2str(20*log10(abs(min(s11)))), ' dB']);
fdisp(FID, ['-> Zin @ ', num2str(f_res/1e9), 'GHz = ', num2str(Zin(f_res_ind)), []]);
fdisp(FID, ['-> |Zin| @ ', num2str(f_res/1e9), 'GHz = ', num2str(abs(Zin(f_res_ind))), ' Ohm' []]);
fdisp(FID, ['-> VSWR @ ', num2str(f_res/1e9), 'GHz = ', num2str(sigmap./sigmam) []]);
fdisp(FID, ['-> radiated power: Prad = ', num2str(nf2ff.Prad) ' Watt']);
fdisp(FID, ['-> directivity: Dmax = ', num2str(nf2ff.Dmax) ' (', num2str(10*log10(nf2ff.Dmax)) ' dBi)']);
fdisp(FID, ['-> efficiency: nu_rad = ', num2str(100*nf2ff.Prad./real(P_in(f_res_ind))) ' %']);
fdisp(FID, ['->']);
```

### A.7.10 Export d'un fichier Touchstone.s1p

Cette fonctionnalité est particulièrement intéressante car elle permet l'interface avec d'autres logiciels comme par exemple Qucs, ce qui facilite la confrontation des résultats de plusieurs simulations / mesures ainsi que l'optimisation de l'antenne.

Cette fonctionnalité nécessite d'avoir installé une boîte à outils Octave :

```
~$ git clone https://github.com/thliebig/CTB
```

```
addpath('~'/CTB');

%%% SAVE TOUCHSTONE
spara = [];
spara(1,1,:) = s11;
write_touchstone('s',freq,spara,[Sim_Path '/' suffix '_simulation.s1p']);
```

### A.7.11 Tracé d'un abaque de Smith via Qucs

Qucs permet de traiter les données contenues dans un fichier Touchstone. Cela peut s'avérer utile notamment pour comparer des résultats de simulation et de mesure, ou pour comparer plusieurs simulations.

Qucs peut afficher plusieurs types de diagrammes différents, prenons l'exemple de l'abaque de Smith.

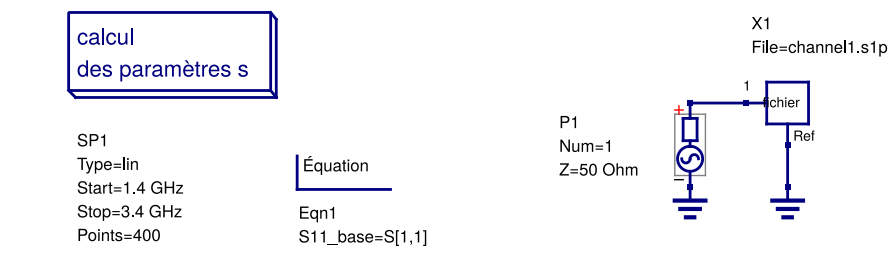


FIGURE A.17 – Qucs : Schéma de traitement du fichier .s1p

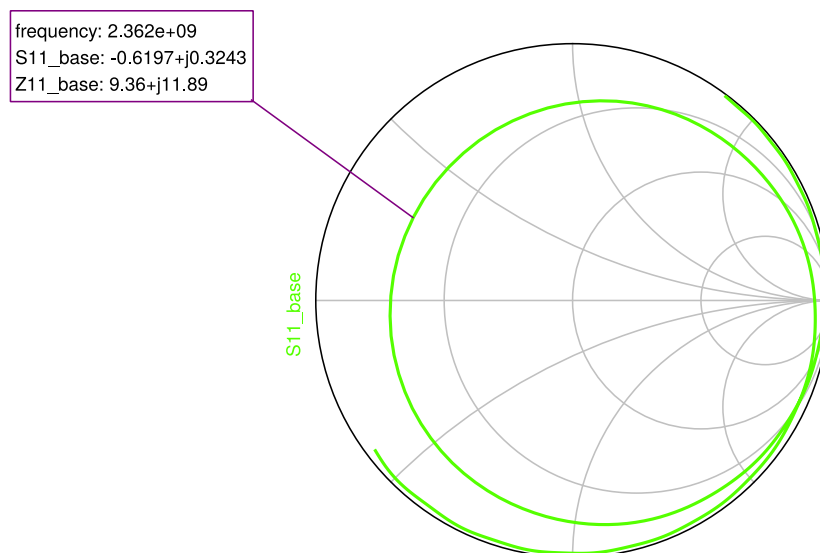


FIGURE A.18 – Qucs : Abaque de Smith du paramètre  $S_{11}$  de l'antenne

## A.8 Visualisation de l'évolution des champs avec Paraview

Le logiciel Paraview permet de lire les fichiers créés avec les dumps pour visualiser les champs suivant dans le domaine du temps ou de la fréquence :

- $\vec{E}$  C'est-à-dire le champ électrique.
- $\vec{H}$  C'est-à-dire le champ magnétique.
- $\vec{J}$  C'est-à-dire le courant.
- $\vec{rot}(\vec{H})$  C'est-à-dire la densité de courant.

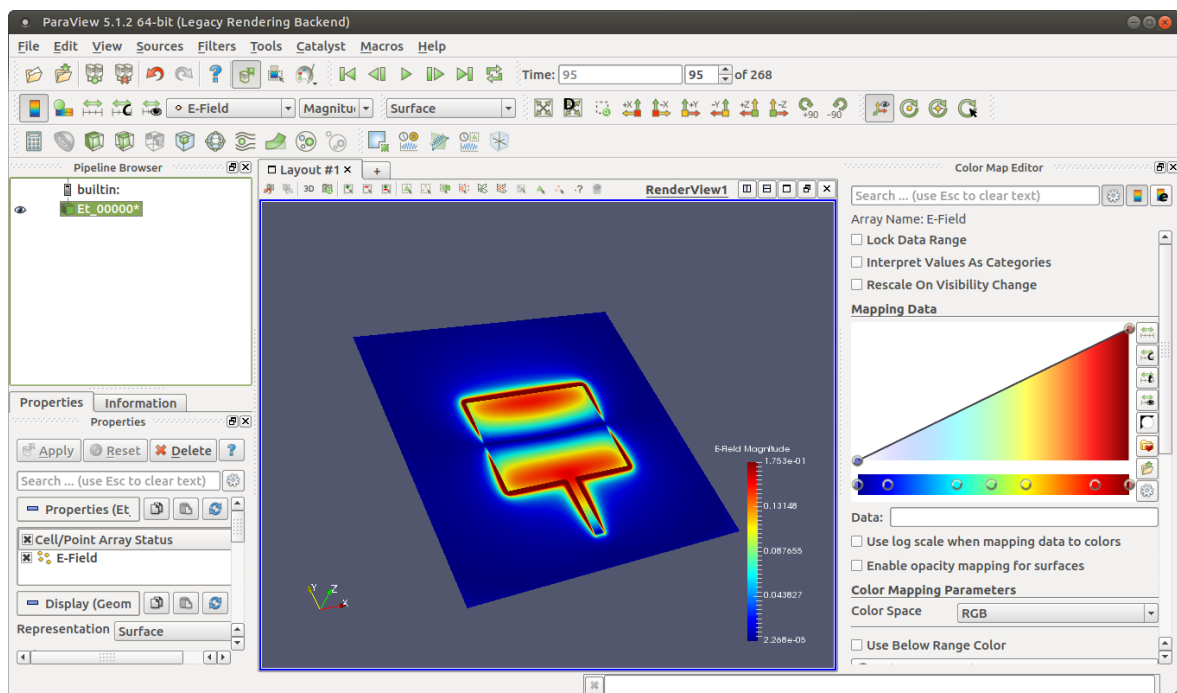


FIGURE A.19 – Paraview : Affichage du champ électrique au cours du temps

- Il faut tout d'abord ouvrir les fichiers dumps .vtk générés par le script.
- On peut choisir d'observer du champ sélectionné
  - Sa magnitude.
  - Sa composante sur les  $\vec{x}$ .
  - Sa composante sur les  $\vec{y}$ .
  - Sa composante sur les  $\vec{z}$ .
- Le bouton *Choose preset* de l'onglet *Mapping data* permet de choisir le jeu de couleur utilisé.
- Les trois boutons *Rescale to data range* permettent d'ajuster les bornes de l'échelle de couleur.

- Les boutons *Play*, *Pause*, etc. permettent de faire évoluer le temps de façon continue ou bien de se déplacer dans le temps image par image.
- La rubrique *Save animation...* du menu *File* permet de sauvegarder l'évolution temporelle du champ sous forme de vidéo.
- Le bouton *Glyph* permet de représenter les champs par des vecteurs :

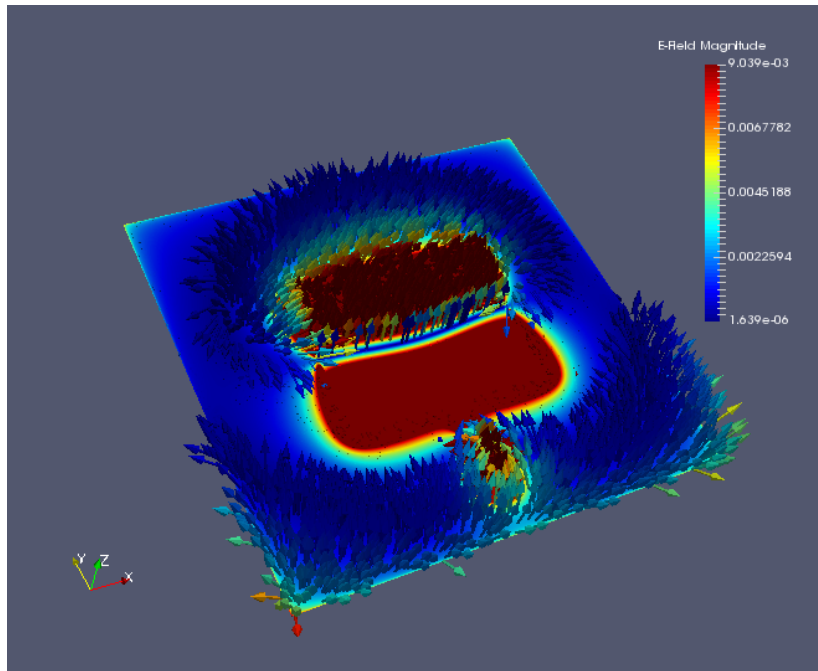


FIGURE A.20 – Représentation vectorielle du champ électrique  $\vec{E}$

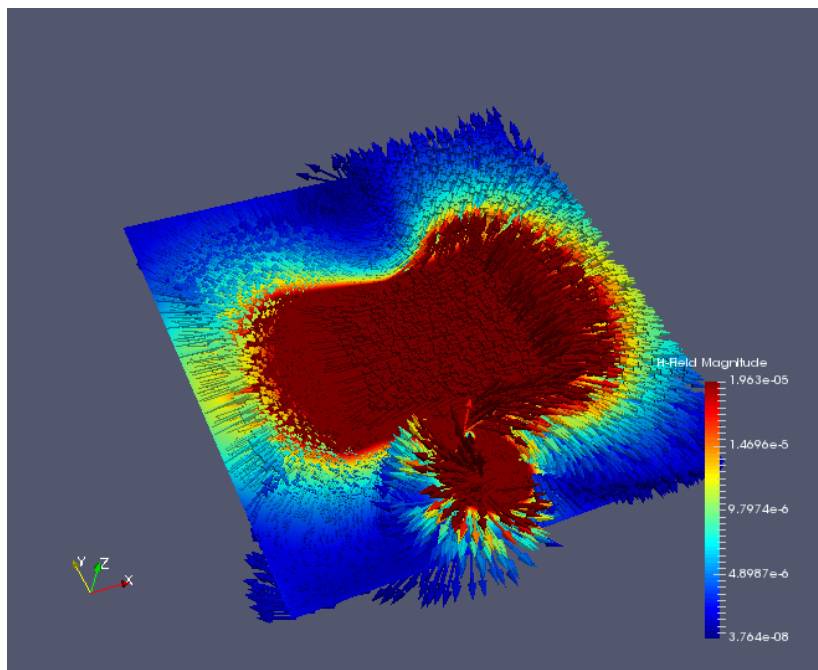


FIGURE A.21 – Représentation vectorielle du champ magnétique  $\vec{H}$