

Федеральное государственное бюджетное образовательное учреждение
высшего образования Национальный исследовательский Саратовский
государственный университет имени Н.Г. Чернышевского

Институт физики

Кафедра физики полупроводников

**Р.А. Щипцов, В.А. Кильдишева, И.В. Хайрушев, Е.А. Рябов,
А.И. Михайлов, И.С. Великанов, Б.В. Сергеева, Н.В. Короневский**

**«ИССЛЕДОВАНИЕ ПАРАМЕТРОВ ЭЛЕКТРОДИНАМИЧЕСКИХ
СТРУКТУР В СВЧ ДИАПАЗОНЕ С ПОМОЩЬЮ ПРОГРАММНОГО
ПАКЕТА OPENEMS»**

Учебное пособие для студентов Института физики, обучающихся по
направлению «Электроника и наноэлектроника»

Саратов 2021

Р.А. Щипцов, В.А. Кильдишева, И.В. Хайрушев, Е.А. Рябов, А.И. Михайлов, И.С. Великанов, Б.В. Сергеева, Н.В. Короневский
«Исследование параметров электродинамических структур в СВЧ диапазоне с помощью программного пакета OpenEMS»: Учеб. пособие для студентов Института физики, обучающихся по направлению «Электроника и наноэлектроника» – Саратов, 2021. –12 с.: ил. Электронное издание.

Представлена лабораторная работа, демонстрирующая возможность расчёта распределения электромагнитного поля в различных системах. Для выполнения лабораторной работы студентам предлагается краткое теоретическое описание симулятора и техническое описание процедуры работы с оборудованием.

Для студентов университета, обучающихся по направлению 11.04.04 «Электроника и наноэлектроника», магистерская программа «Полупроводниковая электроника и молекулярные нанотехнологии», а также научных сотрудников, аспирантов, инженеров, интересующихся современными аспектами электродинамики.

Рекомендует к печати:
Кафедра физики полупроводников
Саратовского государственного университета

© Р.А. Щипцов, В.А. Кильдишева, И.В. Хайрушев и др.

Цель работы: Исследовать параметры электродинамических структур в СВЧ диапазоне с помощью программного пакета OpenEMS

Оборудование: персональный компьютер и программный пакет OpenEMS

Теоретическая часть

Программный пакет OpenEMS

openEMS — электромагнитный симулятор с открытым исходным кодом. openEMS является полноценным электромагнитным симулятором [1].

Здесь изображен пример использования симулятора. Это диаграмма направленности рупорной антенны на частоту 15 ГГц, полученная при помощи openEMS.

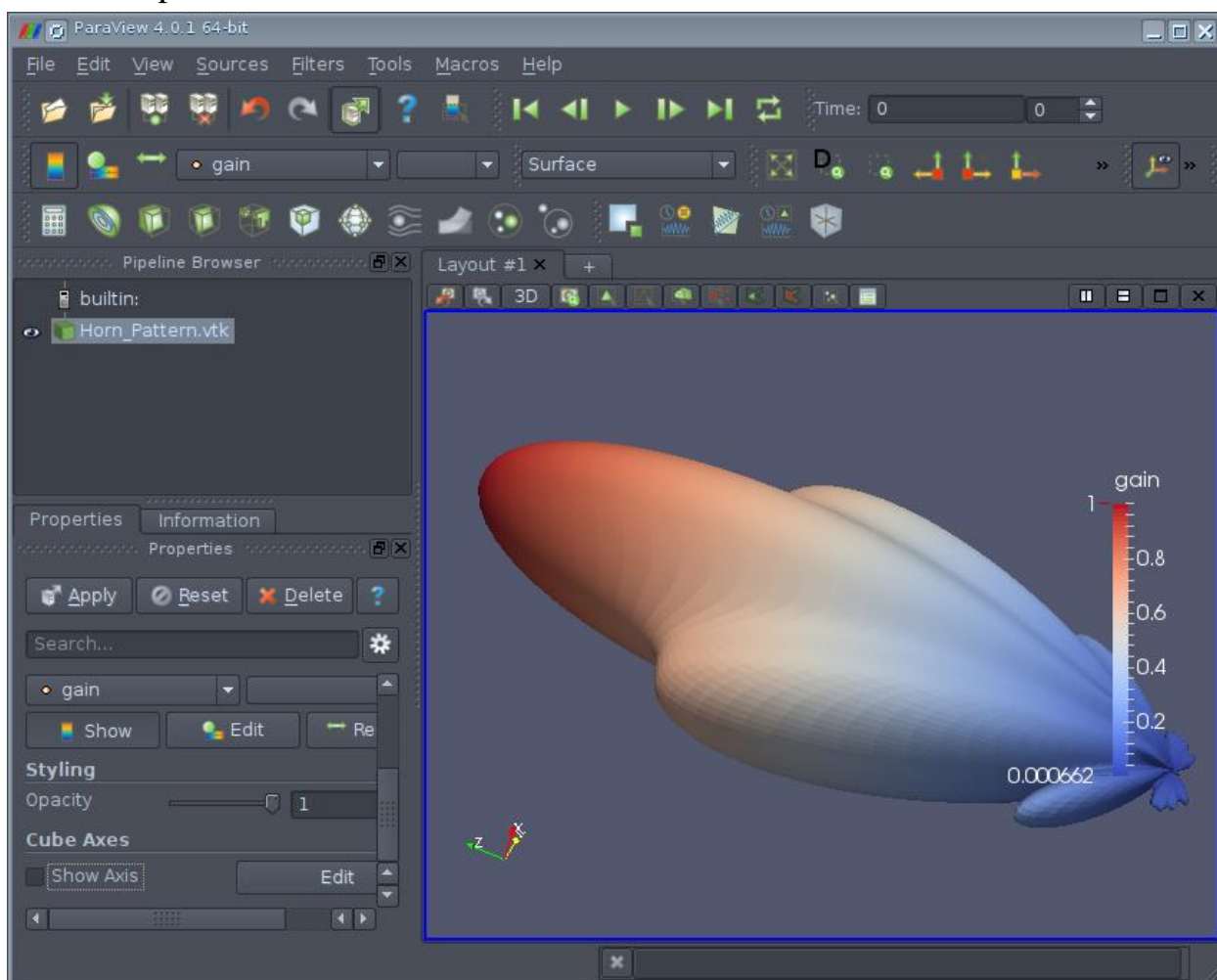


Рисунок 1 - Диаграмма направленности рупорной антенны на частоту 15 ГГц, полученная при помощи openEMS [2].

Ключевые возможности

Отличительной особенностью данного симулятора является то, что он не имеет своего графического интерфейса, а встраивается в **Octave (используется)** или **Matlab**. Задание на моделирование является скриптом **Octave** или **Matlab**, а для просмотра результатов служит программа **Paraview** (не используется).

Области применения

Электромагнитный симулятор предназначен для расчёта распределения электромагнитного поля в различных системах, возбуждаемых как от источника переменного напряжения, так и внешним электромагнитным полем.

1. Расчёт электромагнитного поля (электродинамика) методом FDTD (метод конечных разностей во временной области)
2. Расчёт S-параметров электромагнитных систем
3. Расчёт электромагнитного поля в ближней и дальней зоне
4. Расчёт диаграмм направленности антенн
5. Визуализация результатов

Преимущества

Преимуществом openEMS является то, что он встраивается в **Matlab/Octave**, и следовательно мы имеем неограниченные возможности по параметризации моделей. Для постобработки результатов тоже можно использовать всю мощь этих систем численной математики. При этом не требуется возня с экспортом результатов и их записыванием в математическую систему, а можно сразу же получить графики и вставлять их в статьи.

Недостаток

Недостатком openEMS является повышенный порог вхождения. Симулятор интуитивно не понятен. Чтобы понять что с ним делать, нужно обязательно читать мануалы. Без знания **Matlab/Octave** работать с ним вообще невозможно.

Octave

Octave – это высокоуровневый язык программирования, в первую очередь предназначенный для численных расчетов. Он предоставляет собой удобный консольный интерфейс для решения линейных и нелинейных задач численно, а также для проведения других численных экспериментов используя язык, большей частью совместимый с языком **Matlab**. Его также можно

использовать как язык, ориентированный на пакетную обработку (batch-oriented language) [3, 4].

Достоинства


1. Мощный синтаксис, ориентированный на математику, со встроенными инструментами 2D / 3D построения и визуализации
2. Бесплатное программное обеспечение, работающее на GNU / Linux, macOS, BSD и Microsoft Windows.
3. Drop-in совместим со многими скриптами Matlab

Практическая часть

1. Основные этапы моделирования

1. Настроить режим возбуждения профиля.
2. Создать пробники напряжения и тока, используя режим возбуждения профиля.
3. Рассчитать и построить график импеданса волновода и s-параметров.

2. Пошаговая инструкция для моделирования

1. Создайте свою личную папку, расположив ее по следующему адресу (Домашняя папка\dev\Openems\расположенную Домашняя папка\dev\Openems\ваша папка)
2. Скопируйте подготовленный код (Lab1-cod) в свою личную папку из папки Lab1(Домашняя папка\dev\Openems\расположенную Домашняя папка\dev\Openems\Lab1)
3. Если необходимо то измените формат файла, содержащий код на *****.n
4. В меню пуск открыть вкладку «образование», «наука» и выбрать «GNU Oktave».
5. В открывшемся окне в «диспетчере файлов» выбираем папку с кодом программы и открываем код.
6. Внести необходимые изменения в параметры волновода, структуры и задать электромагнитные свойства.
7. Для запуска моделирования, необходимо выбрать значок .
8. В появившемся окне можно увидеть расположение моделируемой структуры и закрыть окно нажав на красный крестик в верхнем правом углу.
9. Выбрать командное окно, вкладка в которое расположена внизу. В командном окне прописать команду «Yes»
10. Дождаться завершения моделирования.
11. Полученные результаты будут сохранены в вашу личную папку.

3. Пояснения к коду [5-7]

Чтобы запустить сценарий в пустой среде, первые строки:

```
closeall  
clear  
clc
```

Настройте симуляцию, начиная с единиц::

```
physical_constants;
```

```
unit = 1e-3; размерность в миллиметрах
```

Далее укажите размеры волновода для WR90:

```
a = 22.86; % ширины волновода
```

```
b = 10.16; %высота волновода
```

```
c = 100; %длина волновода
```

Установка исследуемого диапазона :

```
f_start = 7e9;
```

```
f_0 = 10e9;
```

```
f_stop = 13e9;
```

```
lambda0 = c0/f_0/unit;
```

Последняя инструкция дает длину волны центральной частоты в миллиметра.

Задайте волноводный режим TE:

```
TE_mode = 'TE10';
```

Задайте разрешение сетки как 30 ячеек на длину волны во всех направлениях:

```
mesh_res = lambda0./[303030];
```

Инициализируем структуру FDTD:

```
FDTD = InitFDTD('NrTS',1e4, 'OverSampling', 5);
```

```
FDTD = SetGaussExcite(FDTD,0.5*(f_start+f_stop),0.5*(f_stop-f_start));
```

Устанавливаем граничные условия. Волна распространяется в направлении z, поэтому концы волновода заканчиваются идеально согласованными слоями (PML или 3). Боковые стенки волновода образованы экстендами x и y пространства моделирования с использованием граничных условий идеального электрического проводника (PEC или 0).

```
BC = [000033]; % pml в поз. и отриц. направление z
```

```
FDTD = SetBoundaryCond(FDTD,BC);
```

Теперь настройте сетку CSXCAD:

```
CSX = InitCSX();
```

```

mesh.x = SmoothMeshLines([0 a], mesh_res(1));
mesh.y = SmoothMeshLines([0 b], mesh_res(2));
mesh.z = SmoothMeshLines([0 length], mesh_res(3));
CSX = DefineRectGrid(CSX, unit, mesh);

```

Теперь укажите порты волновода. Порт 1 находится на расстоянии 11 ячеек от нижнего конца волновода, а порт 2 - на 13 ячеек от верхнего конца. Порты имеют форму коробки с плоскостью возбуждения, расположенной в начальной позиции, и датчиками напряжения и тока, расположенными в позиции остановки. Положение остановки определяет базовую плоскость порта. Порт 1 активен с амплитудой 1, а второй порт пассивен (амплитуда возбуждения не указана).

```

start=[mesh.x(1)mesh.y(1)mesh.z(11)];
stop =[mesh.x(end)mesh.y(end)mesh.z(15)];
[CSX, port{1}] = AddRectWaveGuidePort( CSX, 0, 1, start, stop, 'z', a*unit,
b*unit, TE_mode, 1);

```

```

start=[mesh.x(1)mesh.y(1)mesh.z(end-13)];
stop =[mesh.x(end)mesh.y(end)mesh.z(end-14)];
[CSX, port{2}] = AddRectWaveGuidePort( CSX, 0, 2, start, stop, 'z', a*unit,
b*unit, TE_mode);

```

Определите окно дампа для захвата результатов электронного поля во временной области:

```

CSX = AddDump(CSX, 'Et', 'FileType', 1, 'SubSampling', '2,2,2');
start = [mesh.x(1)mesh.y(1)mesh.z(1)];
stop = [mesh.x(end)mesh.y(end)mesh.z(end)];
CSX = AddBox(CSX, 'Et', 0, start, stop);

```

Задайте геометрию и свойства структуры:

```

%substrate setup
substrate.epsR = 3.38; % диэлектрическая проницаемость
substrate.kappa = 1e-3 * 2*pi*2.45e9 * EPS0*substrate.epsR; % потери
substrate.length = 60; % длина мм

```

Задайте расположение структуры:

```

% Create Substrate
CSX = AddMaterial(CSX, 'substrate');
CSX = SetMaterialProperty( CSX, 'substrate', 'Epsilon', substrate.epsR);
start = [0 0 length/2-substrate.length/2]; % начальная координата
структуры

```

```
stop = [a b length/2+substrate.length/2]; % конечная координата
структуры
```

```
CSX = AddBox( CSX, 'substrate', 0, start, stop );
```

Напишите имитационную модель и выполните симуляцию:

```
Sim_Path = 'tmp_mod';
```

```
Sim_CSX = 'rect_wg.xml';
```

```
[status, message, messageid] = rmdir(Sim_Path,'s');
```

```
[status, message, messageid] = mkdir(Sim_Path);
```

```
WriteOpenEMS([Sim_Path '/' Sim_CSX],FDTD,CSX);
```

```
CSXGeomPlot([Sim_Path '/' Sim_CSX]); % 1 показывает геометрию
```

```
RunOpenEMS(Sim_Path, Sim_CSX)
```

4. Обработка результатов

Шаги постобработки для расчета s-параметров и импеданса волновода:

```
freq = linspace(f_start,f_stop,201);
```

```
port = calcPort(port, Sim_Path, freq);
```

```
s11 = port{1}.uf.ref./ port{1}.uf.inc;
```

```
s21 = port{2}.uf.ref./ port{1}.uf.inc;
```

```
ZL = port{1}.uf.tot./port{1}.if.tot;
```

```
ZL_a = port{1}.ZL; % аналитического импеданса волновода
```

Обратите внимание, что «uf» относится к напряжениям в частотной области, «if» относится к токам в частотной области, «inc» относится к падающим, «ref» относится к отраженным, а «tot» относится к общим.

Построим и сохраним s-параметры:

```
%% compare analytic and numerical wave-impedance
```

```
%%%%%%%%%
```

```
hf = figure();
```

```
plot(freq*1e-6,real(ZL),'Linewidth',2);
```

```
hold on;
```

```
grid on;
```

```
plot(freq*1e-6,imag(ZL),'r--','Linewidth',2);
```

```
plot(freq*1e-6,ZL_a,'g-','Linewidth',2);
```

```
ylabel('ZL (\Omega)','FontSize',12);
```

```
xlabel('frequency (MHz) \rightarrow','FontSize',12);
```

```
xlim([freq(1) freq(end)]*1e-6);
```

```
l = legend('\Re(Z_L)', '\Im(Z_L)', 'Z_L analytic', 'Location', 'Best');
```

```
set(l,'FontSize',12);
```

```
saveas(hf, 'Zl.png', 'png');
```



```
delete(hf);
```

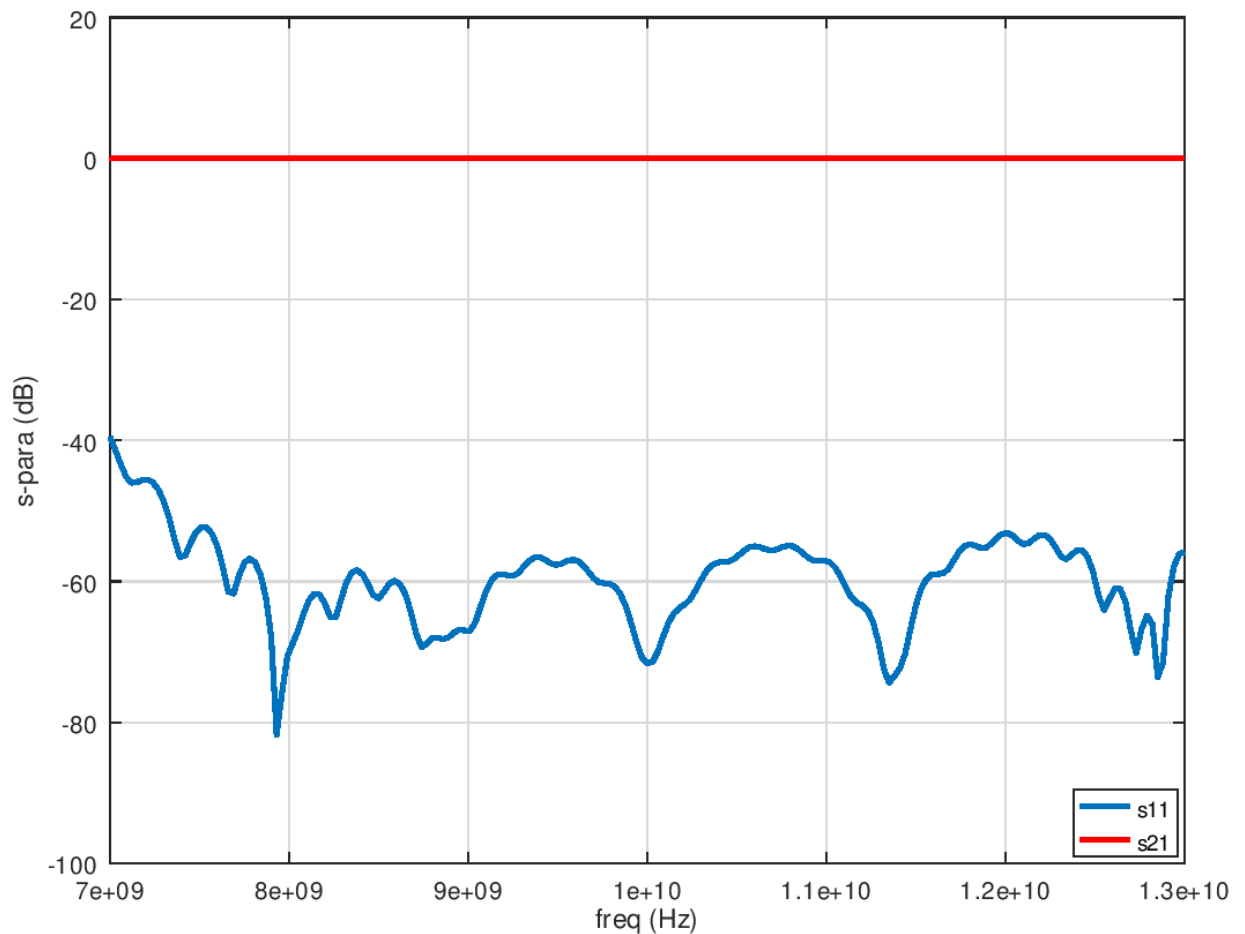


Рисунок 2 - Зависимость S-параметра от частоты для пустого волновода WR90

Сравнение и сохранение аналитического и численного волнового сопротивления:

```
%% compare analytic and numerical wave-impedance  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
hf = figure();  
plot(freq*1e-6,real(ZL),'Linewidth',2);  
hold on;  
grid on;  
plot(freq*1e-6,imag(ZL),'r--','Linewidth',2);  
plot(freq*1e-6,ZL_a,'g-.','Linewidth',2);  
ylabel('ZL (\Omega)','FontSize',12);  
xlabel('frequency (MHz) \rightarrow','FontSize',12);  
xlim([freq(1) freq(end)]*1e-6);  
l = legend('\Re(Z_L)','\Im(Z_L)','Z_L analytic','Location','Best');  
set(l,'FontSize',12);  
saveas(hf, 'Zl.png', 'png');  
delete(hf);
```

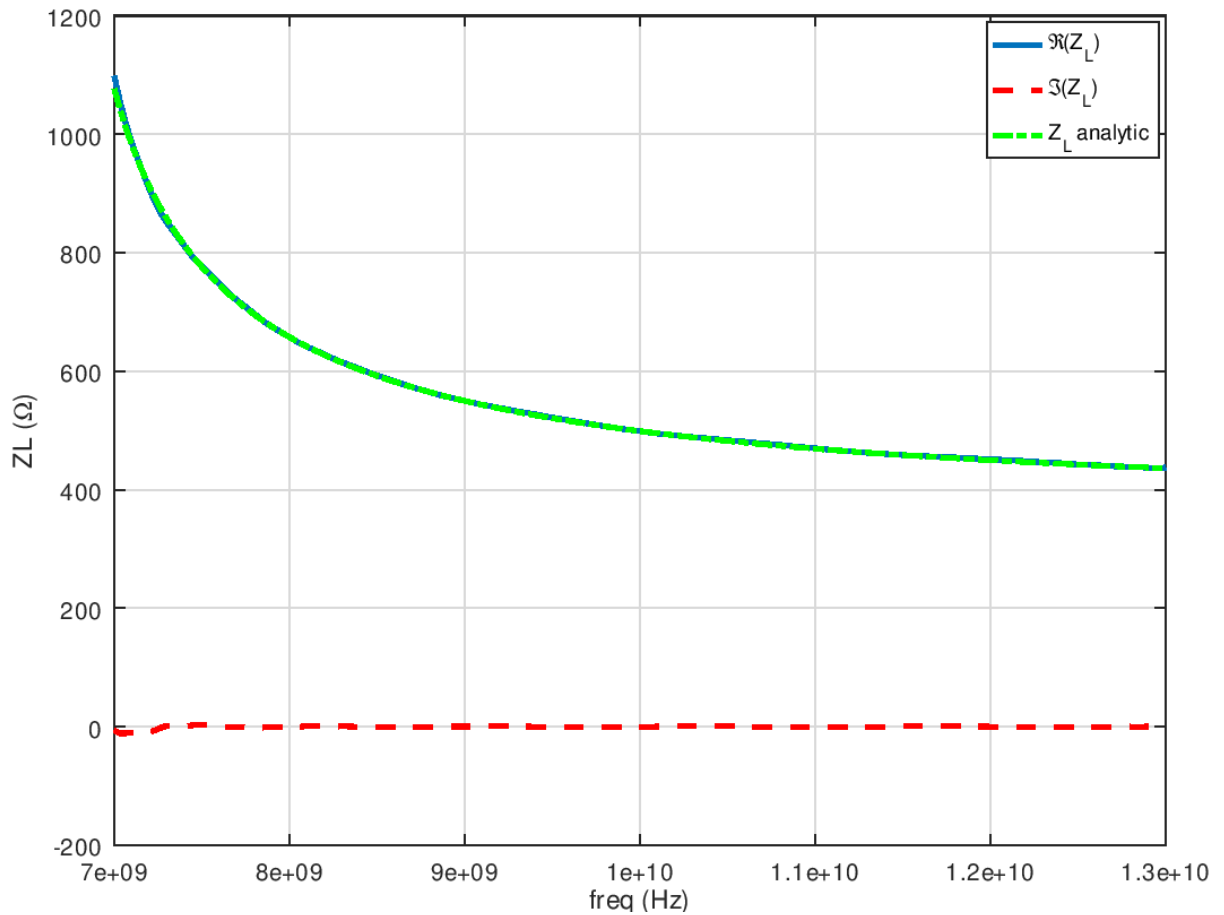


Рисунок 3 - Зависимость импеданса от частоты для пустого волновода WR90

Задание

1. Задать параметры волновода и исследуемой структуры в программном пакете OpenEMS.
2. Определить S-параметры исследуемой структуры для заданного волновода
3. Определить импеданс исследуемой структуры для заданного волновода

Список литературы

- 1 <https://habr.com/ru/post/255317/> - обзор openEMS
- 2 <https://openems.de/start/> - официальный сайт openEMS
- 3 https://ru.wikipedia.org/wiki/GNU_Octave - обзор на википедии
- 4 <https://www.gnu.org/software/octave/> - официальный сайт Octave
- 5 T. Liebig, A. Rennings, S. Held, and D. Erni, "openEMS - A free and open source equivalent-circuit (EC) FDTD simulation platform supporting cylindrical coordinates suitable for the analysis of traveling wave MRI applications," International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, online, DOI: 10.1002/jnm.1875, Dec. 10, 2012.
- 6 T. Liebig, A. Rennings, and D. Erni, "openEMS - A free and open source FDTD software package supporting Cartesian and cylindrical coordinates ideally suited for MRI applications," Magnetic Resonance in Physics, Biology and Medicine (MAGMA), vol. 25, no. suppl 1, pp. 627-628, Oct. 2012, special supplement on Europ. Soc. for Magn. Reson. in Med. and Biol. 29th Annual Sci. Meeting (ESMRMB 2012), Oct. 4-6, Lisbon, Portugal, paper 857 (demo and poster), 2012.
- 7 T. Liebig, A. Rennings, and D. Erni, "openEMS – A Free and Open Source Cartesian and Cylindrical EC-FDTD Simulation Platform Supporting Multi-Pole Drude/Lorentz Dispersive Material Models for Plasmonic Nanostructures," 8th Workshop on Numerical Methods for Optical Nano Structures (organized by OptETH and the Fred Tischer Lecture Series), July 2-4, ETH Zurich, Session on Metamaterials, Periodic Structures, and Numerical Methods I, Switzerland, 2012

Учебное издание

**Щипцов Роман Александрович,
Кильдишева Вероника Андреевна,
Хайрушев Игорь Валентинович,
Рябов Евгений Александрович,
Михайлов Александр Иванович,
Великанов Илья Сергеевич,
Сергеева Бэла Владимировна,
Корневский Никита Владимирович**

**Исследование параметров электродинамических структур в СВЧ
диапазоне с помощью программного пакета OpenEMS**

Учебное пособие
для студентов Института Физики, обучающихся по направлению
«Электроника и наноэлектроника»

Форма издания: Электронная