

INFORME DE INVESTIGACIÓN

Softwares de simulación para Tecnología electrónica
y otras cátedras de ingeniería

Compatibilidad electromagnética

Gonzalo Nahuel Vaca



Universidad Nacional de la Matanza
Argentina

8 de diciembre de 2022

INFORME DE INVESTIGACIÓN

Softwares de simulación para Tecnología electrónica
y otras cátedras de ingeniería

Compatibilidad electromagnética

Gonzalo Nahuel Vaca



Universidad Nacional de la Matanza
Argentina

9 de diciembre de 2022

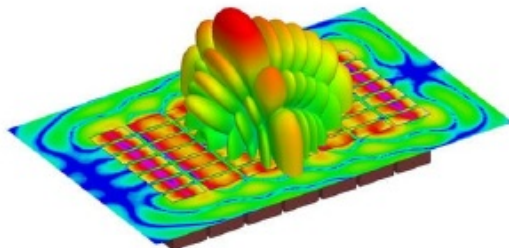


Figura 3: Simulación de un circuito impreso.

2. OpenEMS

El programa OpenEMS fue escrito en el lenguaje C++ y utiliza el método de las diferencias finitas en el dominio temporal (FDTD) para resolver simulaciones electromagnéticas.

El programa fue pensado originalmente para realizar simulaciones de equipos médicos de resonancia magnética (MRI). Sin embargo, se desarrolló un analizador léxico que permite transformar el formato *Hyperlynx* que utiliza KiCad al esquema de geometría de OpenEMS.

El modelo geométrico de OpenEMS es *CSXCAD* y es una biblioteca propia del programa. Además, permite su visualización y trabajo en coordenadas cartesianas y cilíndricas. Finalmente, provee una interfaz para Matlab/Octave y Python.

Las características más relevantes son:

- Diseño y simulación en coordenadas cartesianas.
- Diseño y simulación en coordenadas cilíndricas.
- Soporta modelos *voxel* para simulaciones MRI.
- Multi-hilos y SIMD.
- Intefaz Matlab/Octave.
- Definición de materiales en el espacio.
- Definición de excitaciones en el espacio.
- Generación de archivos en formato *vtk* y *hdf5*
- Materiales dispersivos(Drude/Lorentz/Debye)
- Rutinas de post-procesamiento configurables (Matlab/Octave).

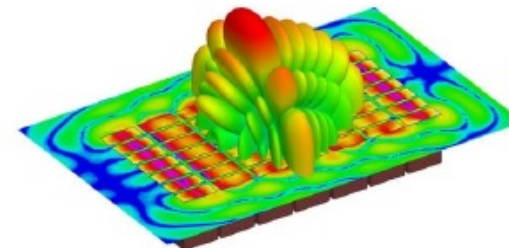


Figura 3: Simulación de un circuito impreso.

2. OpenEMS

En esta sección se encuentran los detalles de la evaluación del programa OpenEMS.

2.1. Introducción

El programa OpenEMS fue escrito en el lenguaje C++ y utiliza el método de las diferencias finitas en el dominio temporal (FDTD) para resolver simulaciones electromagnéticas.

El programa fue pensado originalmente para realizar simulaciones de equipos médicos de resonancia magnética (MRI). Sin embargo, se desarrolló un analizador léxico que permite transformar el formato *Hyperlynx* que utiliza KiCad al esquema de geometría de OpenEMS.

El modelo geométrico de OpenEMS es *CSXCAD* y es una biblioteca propia del programa. Además, permite su visualización y trabajo en coordenadas cartesianas y cilíndricas. Finalmente, provee una interfaz para Matlab/Octave y Python.

Las características más relevantes son:

- Diseño y simulación en coordenadas cartesianas.
- Diseño y simulación en coordenadas cilíndricas.
- Soporta modelos *voxel* para simulaciones MRI.
- Multi-hilos y SIMD.
- Intefaz Matlab/Octave.
- Definición de materiales en el espacio.
- Definición de excitaciones en el espacio.

- Sub-espacios de simulación para optimizar densidades de cálculo.
- Simulaciones remotas por medio de *ssh*.
- Es libre y gratuito.

Se creó un entorno de trabajo en Ubuntu 2020 y en contenedores de *docker* para las versiones de Ubuntu 2018 y 2016. Además se estudió como utilizarlo con Octave y se obtuvieron los siguientes resultados:

- El analizador léxico se encuentra abandonado y no es posible hacerlo funcionar
- KiCad 6 cambió su formato *Hyperlynx* y el analizador léxico es por lo tanto obsoleto.
- Las geometrías de los circuitos impresos se deben construir a mano.
- Los errores de geometría no son útiles para depurar los datos ingresados.

Dado el estado actual de este programa se hace imposible utilizarlo como herramienta de las cátedras si antes no se repara y actualiza el analizador léxico.

En la figura 4 se puede observar una simulación realizada en OpenEMS.

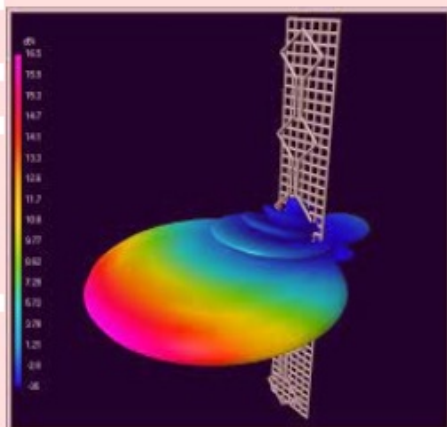


Figura 4: Simulación una antena en OpenEMS.

- Generación de archivos en formato *vtk* y *hdf5*
- Materiales dispersivos (Drude/Lorentz/Debye)
- Rutinas de post-procesamiento configurables (Matlab/Octave).
- Sub-espacios de simulación para optimizar densidades de cálculo.
- Simulaciones remotas por medio de *ssh*.
- Es libre y gratuito.

Se creó un entorno de trabajo en Ubuntu 2020 y en contenedores de *docker* para las versiones de Ubuntu 2018 y 2016. Además se estudió como utilizarlo con Octave y se obtuvieron los siguientes resultados:

- El analizador léxico se encuentra abandonado y no es posible hacerlo funcionar
- KiCad 6 cambió su formato *Hyperlynx* y el analizador léxico es por lo tanto obsoleto.
- Las geometrías de los circuitos impresos se deben construir a mano.
- Los errores de geometría no son útiles para depurar los datos ingresados.

Dado el estado actual de este programa se hace imposible utilizarlo como herramienta de las cátedras si antes no se repara y actualiza el analizador léxico.

En la figura 4 se puede observar una simulación realizada en OpenEMS.

2.2. Instalación

La instalación del programa no es trivial y a continuación se detallan los pasos a seguir para tener el programa en funcionamiento. Esto tiene la finalidad de demostrar la dificultad que enfrentaría el cuerpo de estudiantes y docentes.

A continuación se detallan los pasos a seguir:

1. Satisfacer las dependencias del programa, a continuación se muestran los comandos para el sistema Ubuntu 22.04 LTS:
 - Dependencias mínimas obligatorias:


```
sudo apt-get install
build-essential cmake git libhdf5-dev libvtk7-dev
libboost-all-dev libcglib-dev libtinyxml-dev
qtbase5-dev libvtk7-qt-dev
```

3. Ansys

Ansys HFSS (simulador de estructura de alta frecuencia) es un solucionador comercial de métodos de elementos finitos para estructuras electromagnéticas (EM) de Ansys que ofrece varias tecnologías de resolución de última generación. Cada solucionador en ANSYS HFSS es un procesador de solución automatizado para el cual el usuario dicta la geometría, las propiedades del material y el rango requerido de frecuencias de solución.

Los ingenieros utilizan Ansys HFSS principalmente para diseñar y simular componentes electrónicos de alta velocidad y alta frecuencia en sistemas de radar, sistemas de comunicación, satélites, ADAS, microchips, placas de circuito impreso, productos IoT y otros dispositivos digitales y dispositivos RF. El solucionador también se ha utilizado para simular el comportamiento electromagnético de objetos como automóviles y aviones. ANSYS HFSS permite a los diseñadores de sistemas y circuitos simular problemas de EM, como pérdidas por atenuación, acoplamiento, radiación y reflexión.

HFSS captura y simula objetos en 3D, teniendo en cuenta la composición de los materiales y las formas/geometrías de cada objeto. Además, está preparada para ser utilizadas en el diseño de antenas y elementos complejos de circuitos electrónicos de radiofrecuencia, incluidos filtros, líneas de transmisión y empaques.

Durante el transcurso del 2022 se logró un contacto comercial que realizó una propuesta de trabajo a la UNLaM a cambio de licencias y capacitación. Se dió una capacitación sobre la funcionalidad y sinergia entre los distintos motores de físicas.

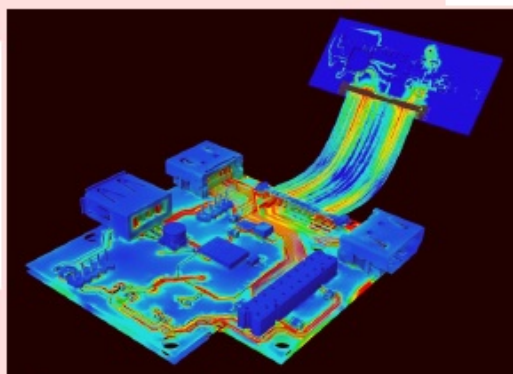


Figura 5: Simulación de PCB flexible en Ansys.

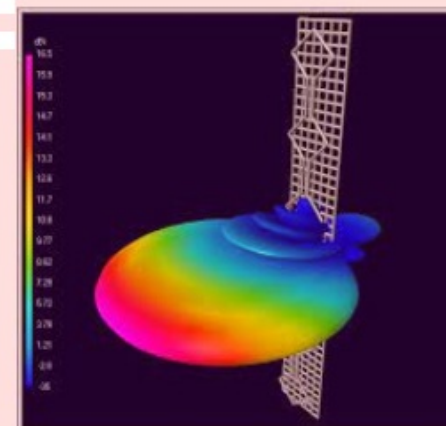


Figura 4: Simulación una antena en OpenEMS.

- GNU Octave (equivalente GNU de Matlab, opcional):
`sudo apt-get install octave liboctave-dev`
- Interfaz Python (opcional):
`sudo pip3 install matplotlib cython h5py`

2. Clonar e instalar

- Clonar el repositorio y sus submódulos:
`git clone --recursive https://github.com/thliebig/openEMS-Project.git`
- Dentro del repositorio ejecutar:
`./update_openEMS.sh ~/opt/openEMS --python`

2.3. Verificación

La prueba más simple para verificar la instalación es ejecutar el binario de openEMS. Para lograrlo debe navegar hasta la carpeta de la instalación, la cuál normalmente es `opt/openEMS/bin` dentro de su usuario; luego se debe ejecutar el comando `./openEMS` y deberá observar una salida como la siguiente:

4. Elmer

Elmer es un software de simulación multi-física de código abierto desarrollado principalmente por *IT Center for Science (CSC)*. El desarrollo de Elmer se inició como una colaboración nacional con las universidades finlandesas, los institutos de investigación y la industria.

Elmer incluye modelos físicos de dinámica de fluidos, mecánica estructural, electromagnetismo, transferencia de calor y acústica. Estos se describen mediante ecuaciones diferenciales parciales que Elmer resuelve mediante el método de elementos finitos (FEM). Elmer admite computación paralela.

Actualmente los campos de uso más destacados son la glaciología computacional y el electromagnetismo computacional.

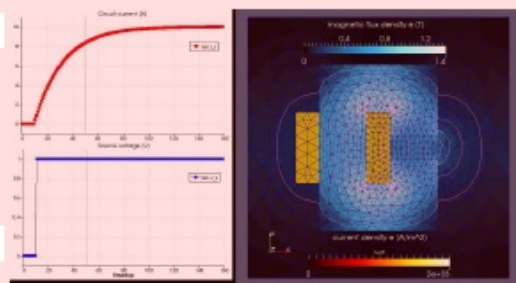


Figura 6: Simulación en Elmer.

```
| openEMS 64bit -- version v0.0.35-76-gd4448fa
| (C) 2010-2018 Thorsten Liebig <thorsten.liebig@gmx.de>
| GPL license
```

Used external libraries:

```
CSXCAD -- Version: v0.6.2-109-gcd9decb
hdf5 -- Version: 1.10.7
compiled against: HDF5 library version: 1.10.7
tinyxml -- compiled against: 2.6.2
fparser
boost -- compiled against: 1_74
vtk -- Version: 7.1.1
compiled against: 7.1.1
```

A continuación se detallan los pasos a seguir para verificar la correcta instalación de openEMS con GNU Octave.

Como se muestra en la figura 5 se debe colocar el *path* de openEMS en la ventana de comandos.

Los comandos para agregar los *path* son:

```
addpath('/opt/openEMS/share/openEMS/matlab');
addpath('/opt/openEMS/share/CSXCAD/matlab');
```

Command Window

```
GNU Octave, version 6.4.0
Copyright (C) 2021 The Octave Project Developers.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.

Octave was configured for "x86_64-pc-linux-gnu".

Additional information about Octave is available at https://www.octave.org.

Please contribute if you find this software useful.
For more information, visit https://www.octave.org/get-involved.html

Read https://www.octave.org/bugs.html to learn how to submit bug reports.
For information about changes from previous versions, type 'news'.

>> addpath('/opt/openEMS/share/openEMS/matlab');
addpath('/opt/openEMS/share/CSXCAD/matlab');
>> |
```

Figura 5: Configuración de *path* en GNU Octave.

El siguiente paso es verificar la interfaz CXCAD, a continuación se muestra el comando y respuesta en la ventana de comandos de GNU Octave:

5. Conclusiones

- OpenEMS: el programa no logró cumplir las expectativas y no se ajusta a las necesidades académicas. No se recomienda para su uso ni continuar su investigación.
- Ansys: a pesar que el programa es propietario y demandaría un costo económico para la universidad, se ajusta a las necesidades académicas. Se recomienda continuar con su investigación.
- Elmer: se debe comenzar su investigación. Esto significa comenzar con sus pruebas de simulación y flujo de trabajo.

```
>> InitCSX
ans =
Properties: []
```

Con la interfaz inicializada se puede realizar una verificación de sus funciones como se muestra a continuación:

```
>> InitFDTD('NrTS', 0, 'EndCriteria', 0)
ans =
ATTRIBUTE: [1x1 struct]
```

El siguiente paso es verificar el correcto funcionamiento del simulador como se muestra a continuación:

```
>> RunOpenEMS( '.', 'nonexistent.xml', '' )
[...]
Read openEMS xml file: nonexistent.xml ...
openEMS: Error File-Loading failed!!! File: nonexistent.xml
```

El error que se muestra significa que el simulador funciona correctamente ya que su argumento es un archivo inexistente. Sin embargo, el binario del simulador se invocó sin problemas.

Es importante verificar que el programa de visualización de definiciones de geometría funcione de forma correcta, esto se logra con el siguiente comando:

```
CSXGeomPlot('nonexistent.xml')
```

La ventana de comandos debe arrojar un error ya que el archivo de geometría no existe, sin embargo se debe ejecutar el visualizador como se muestra en la figura 6.

Llegado a este punto se puede concluir que openEMS fue instalado de forma correcta.

Se recomienda instalar el programa ParaView para tener una herramienta para visualizar las simulaciones. Los ejemplos de este documento utilizan ParaView para representar de forma gráfica los resultados.

2.4. Ejemplos de uso

2.4.1. Guía de onda superficial

Las instrucciones para generar la geometría, su excitación y la definición de las variables a observar son las siguientes: