

# Desarrollo de una GUI en MATLAB Para la Simulación del Fenómeno de Refracción por FDTD

## Developing a GUI in MATLAB For Simulation Refraction Phenomenon by FDTD

Larry Theran <sup>1</sup>

<sup>1</sup>Departamento de Física  
Facultad de Ciencias Básicas, Universidad del Atlántico, Barranquilla (Colombia)  
E-mail: lrtheran@gmail.com

Sonia Valbuena <sup>2</sup>

<sup>2</sup>Profesora departamento de Matemáticas  
Facultad de Ciencias Básicas, Universidad del Atlántico, Barranquilla (Colombia)  
E-mail: svalbuena@gmail.com

Francisco Racedo <sup>3</sup>

<sup>3</sup>Profesor departamento de Física  
Facultad de Ciencias Básicas, Universidad del Atlántico, Barranquilla (Colombia)  
Email: fran@mail.uniatlantico.edu.co

Recibido: 02/10/2014 - Aceptado: 13/11/2014

---

### Resumen

El método de diferencias finitas en el dominio del tiempo (FDTD) ha demostrado ser una herramienta útil para el análisis de los fenómenos electromagnéticos. En este trabajo se presenta una aplicación grafica basada en la GUIDE de MATLAB, en la cual se muestra la implementación del algoritmo de diferencias finitas en el dominio del tiempo, para la simulación del fenómeno de refracción. Se introducirán unas condiciones de frontera absorbentes (ABC) de tipo Capas perfectamente acopladas (PML), ya que este es un problema de evolución temporal con dominios no acotados.

*Palabras claves:* FDTD; Refraccion; GUI; condiciones de frontera absorbentes.

### Abstract

The finite difference time-domain (FDTD) has proved a useful tool for the analysis of electromagnetic phenomena. This paper presents a graphical application based on MATLAB GUIDE, in which the implementation of the finite difference algorithm is shown in the time domain, to simulate the phenomenon of refraction is presented. Absorbing boundary conditions (ABC) type of perfectly matched layers (PML) will be introduced, as this is a problem of time evolution with unbounded domains.

*Keywords:* FDTD; Refraction; GUI; absorbent boundary conditions.

---

## 1. INTRODUCCION

Mediantes el método FDTD se puede simular numéricamente la propagación de una on-

da electromagnética aproximando las derivadas parciales en el espacio y en el tiempo de las ecuaciones de Maxwell mediante diferencias finitas.

Dentro de las diferentes aplicaciones del electromagnetismo computacional se encuentra el estudio del cambio de dirección que experimenta una onda al pasar de un medio material a otro. Se usó el entorno de programación de Matlab, para implementar el algoritmo FDTD ya que este nos brinda la facilidad de trabajar con vectores y matrices, además nos permite de una manera sencilla realizar una interfaz gráfica (GUI) que contenga los resultados numéricos. Esta GUI permite analizar el fenómeno de refracción de manera sencilla sin tener que utilizar instrumentos que podrían ser costosos, ayudando así a comprender este tipo de fenómenos de una forma más dinámica e interactiva.

## 2. ECUACIONES DE MAXWELL

Las ecuaciones de Maxwell son un conjunto de cuatro ecuaciones que describen por completo los fenómenos electromagnéticos y por medio de las cuales podemos demostrar que la luz es una onda electromagnética. Para un medio lineal homogéneo e isotrópico son:

$$\frac{\partial \vec{D}}{\partial t} = \frac{1}{\sqrt{\mu_0 \epsilon_0}} \nabla \times \vec{H} \quad (1)$$

$$\frac{\partial \vec{H}}{\partial t} = -\frac{1}{\sqrt{\mu_0 \epsilon_0}} \nabla \times \vec{E} \quad (2)$$

Donde  $\vec{D}$  es la densidad de flujo eléctrico,  $\mu$  es la permeabilidad magnética,  $\epsilon$  es la permitividad eléctrica. Desacoplando las ecuaciones rotacionales en sus componentes obtenemos el siguiente grupo de ecuaciones

$$\frac{\partial \vec{D}_x}{\partial t} = \frac{1}{\sqrt{\mu_0 \epsilon_0}} \left( \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \right) \quad (3)$$

$$\frac{\partial \vec{D}_y}{\partial t} = \frac{1}{\sqrt{\mu_0 \epsilon_0}} \left( \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \right) \quad (4)$$

$$\frac{\partial \vec{D}_z}{\partial t} = \frac{1}{\sqrt{\mu_0 \epsilon_0}} \left( \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) \quad (5)$$

$$-\frac{\partial H_x}{\partial t} = \frac{1}{\sqrt{\mu_0 \epsilon_0}} \left( \frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} \right) \quad (6)$$

$$-\frac{\partial H_y}{\partial t} = \frac{1}{\sqrt{\mu_0 \epsilon_0}} \left( \frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} \right) \quad (7)$$

$$-\frac{\partial H_z}{\partial t} = \frac{1}{\sqrt{\mu_0 \epsilon_0}} \left( \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} \right) \quad (8)$$

## 3. FORMULACIÓN NUMÉRICA

Las ecuaciones anteriores modelan el campo eléctrico y magnético en tiempo y espacio. La idea del algoritmo de Yee es convertir estas ecuaciones que están en forma continua a una forma discreta usando diferencias centrales.

La división de la región de cálculo en un malla discreto está dada por la celda de Yee.

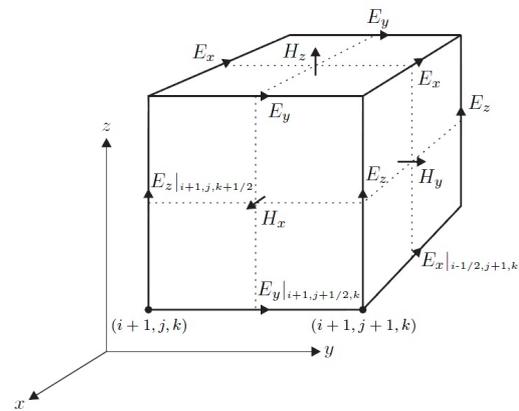


Figura 1. Celda de Yee en tres dimensiones donde se puede observar la posición de los campos magnéticos y eléctricos.

Se tendrán en cuenta como en todo modelo de aproximación la determinación del tamaño de la celda tanto en espacio como en tiempo. Dado que una onda electromagnética no se puede propagar a una velocidad mayor que la de la luz, la onda para propagarse en una celda requiere un tiempo  $\Delta t = \frac{\Delta x}{c_0}$ , esta expresión se puede generalizar para cualquier dimensión por medio de la condición de Courant.

$$\Delta t \leq \frac{\Delta x}{\sqrt{nc_0}} \quad (9)$$

Donde  $n$  es la dimensión de la simulación.

Para llevar la solución a un dominio bi-dimensional, se usara una onda trasversal magnética (TM), la cual contiene el plano  $xy$  polarizado, en donde  $E_y = E_x = 0$  y  $H_z = 0$ . Entonces las ecuaciones en diferencias que se usan para el algoritmo son:

$$\frac{D_z^n(i, j, k + \frac{1}{2}) - D_z^{n-1}(i, j, k + \frac{1}{2})}{\Delta t} = \frac{1}{\sqrt{\mu_0 \epsilon_0}} \left( \frac{H_y^{n-\frac{1}{2}}(i + \frac{1}{2}, j, k + \frac{1}{2}) - H_y^{n-\frac{1}{2}}(i - \frac{1}{2}, j, k + \frac{1}{2})}{\Delta x} \right) \quad (10)$$

$$\frac{H_x^{n+\frac{1}{2}}(i, j + \frac{1}{2}, k + \frac{1}{2}) - H_x^{n-\frac{1}{2}}(i, j + \frac{1}{2}, k + \frac{1}{2})}{\Delta t} = \frac{1}{\sqrt{\mu_0 \epsilon_0}} \left( \frac{E_z^n(i, j + 1, k + \frac{1}{2}) - E_z^n(i, j, k + \frac{1}{2})}{\Delta y} \right) \quad (11)$$

$$\frac{H_y^{n+\frac{1}{2}}(i + \frac{1}{2}, j, k + \frac{1}{2}) - H_y^{n-\frac{1}{2}}(i + \frac{1}{2}, j, k + \frac{1}{2})}{\Delta t} = \frac{1}{\sqrt{\mu_0 \epsilon_0}} \left( \frac{E_z^n(i + 1, j, k + \frac{1}{2}) - E_z^n(i, j, k + \frac{1}{2})}{\Delta x} \right) \quad (12)$$

Debido a que la región de cálculo computacional cuanta con recursos finitos, en el problema se deben tener en cuenta las reflexiones causadas cuando la onda llega el final de la malla de simulación.

Para resolver este problema se implementan las condiciones de frontera absorbentes propuestas por Berenger en 1994 las cuales se conocen con el nombre de Capas Perfectamente Acopladas (PML), de esta manera la onda se puede propagar en extensión infinita, tal que la solución sea válida para todo tiempo.

#### 4. SIMULACIÓN DEL FENÓMENO DE REFRACCIÓN DE LA LUZ

La refracción es el cambio de dirección que experimenta una onda al pasar de un medio material a otro. Solo se produce si la onda incide oblicuamente sobre la superficie de separación de los dos medios y si estos tienen índices de refracción distintos. La refracción se origina en el

cambio de velocidad de propagación de la onda señalada.

El fenómeno de la refracción se rige por la llamada Ley de la Refracción o Ley de Snell:

$$n_1 \sin \theta_1 = n_2 \sin \theta_2 \quad (13)$$

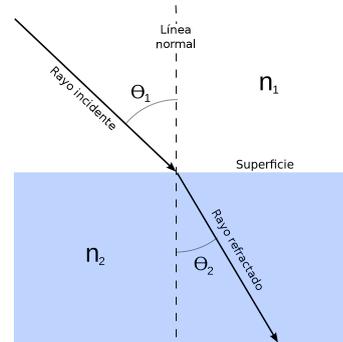


Figura 2. Fenómeno de Refracción.

La simulación se obtiene al resolver la ecuación de Maxwell para una TM onda como se presenta en la sección anterior, teniendo en cuenta las características de este fenómeno se trabajo en un dominio que tiene dos medios de comunicación, el primer medio se puede variar teniendo en cuenta los índices de refracción que se muestran en la siguiente tabla.

Cuadro 1. Tabla de Índices de refracción para diferentes materiales

Material	Índice de refracción
Vacío	1
Dióxido de carbono	1,0004
Hielo	1,31
Agua (a 20° C)	1,333
Cuarzo	1,544
Rubí	1,767
Diamante	2,417

El segundo medio siempre es el vacío de tal manera que siempre obtengamos una simulación correcta, debido a que el fenómeno se da cuando la onda pasa de un medio con índice de refracción mayor a una menor.

A continuación se muestran los resultados obtenidos con medios de diferentes índices de refracción y varios ángulos de incidencia de la onda electromagnética.

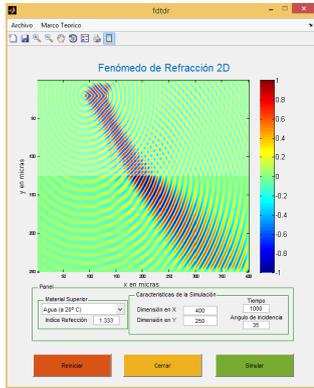


Figura 3. Fenómeno de Refracción entre Agua y vacío con un ángulos de 35°.

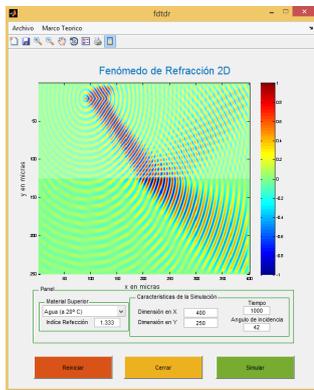


Figura 4. Fenómeno de Refracción entre Agua y vacío con un ángulos de 42°.

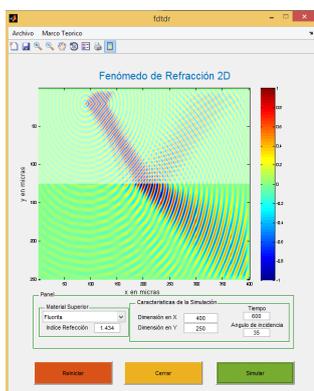


Figura 5. Fenómeno de Refracción entre Fluorita y vacío con un ángulos de 35°.

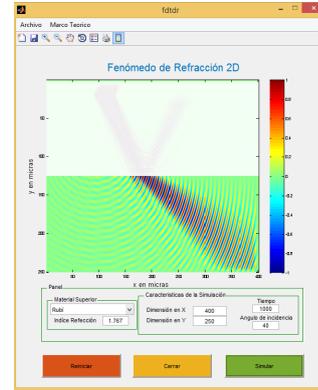


Figura 6. Fenómeno de Refracción entre Rubí y vacío con un ángulos de 40°.

### 5. IMPLEMENTACION DE LA GUI

El desarrollo de la interfaz grafica se llevó a cabo en el entorno de programación grafica que ofrece Matlab, para poder realizar y ejecutar programas de simulación de forma simple e interactiva. Este paquete se conoce como Matlab Guide y tiene características básicas de programas visuales como Visual Basic o Visual C++.

En este caso se utilizó para interactuar de forma sencilla con la simulación del fenómeno de refracción de la luz, el proceso de implementación se ve de la siguiente manera:

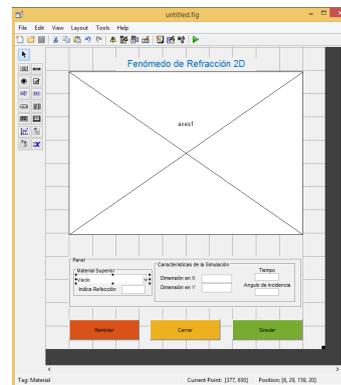


Figura 7. Paquete Guide de Matlab.

Parte del código se muestra a continuación:

```
gui.Singleton = 1;
gui.State = struct('gui.Name',
    'gui.Singleton',
    'gui.OpeningFcn',
    'gui.OutputFcn',
    'gui.LayoutFcn',
    'gui.Callback',
if nargin && ischar(varargin{1})
    gui.State.gui.Callback = str2func
        (varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui.mainfcn
        (gui.State, varargin{:});
else
    gui.mainfcn(gui.State, varargin{:});
end

function untitled.OpeningFcn(hObject,
eventdata, handles, varargin)
A=imread('Refrac.png');
image(A)
axis off
handles.output = hObject;

%Update handles structure
guidata(hObject, handles);

function varargout = untitled.OutputFcn
(hObject, eventdata, handles)

varargout{1} = handles.output;

%-----Aqui##U+FFFD#a Sndü[U+FFFD#
function Simular.Callback
(hObject, eventdata, handles)
time_tot=str2double(get(handles.EntradaT,'string'));
xdim=str2double(get(handles.EntradaX,'string'));
ydim=str2double(get(handles.EntradaY,'string'));
alpha=str2double(get(handles.EntradaA,'string'));
index=str2double(get(handles.Indi,'string'));

function Cerrar.Callback
(hObject, eventdata, handles)
opc=questdlg
({{'FFFD}Desenlir_del_programa?',
'SALIR','Si','No','No'});
if strcmp(opc,'No')
    return;
end
close(untitled)

function Detener.Callback
(hObject, eventdata, handles)

function EntradaX.Callback
(hObject, eventdata, handles)
function EntradaX.CreateFcn
(hObject, eventdata, handles)
if ispc && isequal
(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function EntradaY.Callback
(hObject, eventdata, handles)
function EntradaY.CreateFcn
(hObject, eventdata, handles)
if ispc && isequal(get
(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
set(hObject,'BackgroundColor','white');
end

function EntradaT.Callback
(hObject, eventdata, handles)
function EntradaT.CreateFcn
(hObject, eventdata, handles)
if ispc && isequal(get
(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function EntradaA.Callback
(hObject, eventdata, handles)
function EntradaA.CreateFcn
(hObject, eventdata, handles)
if ispc && isequal
(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%— Executes on selection change in Material.
function Material.Callback
(hObject, eventdata, handles)
v=get(handles.Material,'Value');
set(handles.Indi,'string',v);

switch v
    case 1
        set(handles.Indi,'string',1);
    case 2
        set(handles.Indi,'string',1.0004);
    case 3
        set(handles.Indi,'string',1.333);
    case 4
        set(handles.Indi,'string',1.333);
    case 5
        set(handles.Indi,'string',1.36);
    case 6
        set(handles.Indi,'string',1.36);
    case 7
        set(handles.Indi,'string',1.38);
    case 8
        set(handles.Indi,'string',1.434);
    case 9
        set(handles.Indi,'string',1.473);
    case 10
        set(handles.Indi,'string',1.501);
    case 11
        set(handles.Indi,'string',1.52);
    case 12
        set(handles.Indi,'string',1.544);
    case 13
        set(handles.Indi,'string',1.767);
    otherwise 14
        set(handles.Indi,'string',2.417);

end

function Material.CreateFcn
(hObject, eventdata, handles)

if ispc && isequal
(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

## 6. CONCLUSIONES

En este trabajo el método FDTD se ha aplicado con éxito para analizar el fenómeno de refracción que se presenta cuando la onda electromagnética pasa de un medio a otro, el cual es un problema bien conocido en la óptica. debido a estos se pudo comparar los resultados obtenidos por el método FDTD y las expresiones analíticas en este tipo de configuraciones. También se implementó con éxito el formalismo de las capas perfectamente adaptadas que nos permitió obtener resultados satisfactorios.

## Referencias

- [1] A. TAFLOVE, S. HAGNESS (2005) *Computational Electrodynamics: the finite-difference time-domain method*. Artech House, Boston, London.
- [2] DENNIS M. SULLIVAN (2000) "Electromagnetic Simulation Using the FDTD Method". *IEEE PRESS*.
- [3] Valéria de Magalhaes Iório. *EDP um curso de graduacao*. Instituto de matemática pura y aplicada IMPA, Rio de Janeiro, 1991.
- [4] Moysey Brio, Aramais Zakharian & Gary M. Webb. *Numerical Time-Dependent Partial Differential Equations for Scientists and Engineers*. C.K Chui, Stanford University, 2010.
- [5] Univ. de Extremadura. *Apuntes de ecuaciones diferenciales*, Dpto. de Matemáticas. Badajoz-España, Mayo 2013. [Documento en línea]. <http://matematicas.unex.es/ricarfr/Ec-Diferenciales/LibroEDlat.pdf>.
- [6] Ignacio Gracia Rivas & Narciso Román Roy. *Apuntes de ecuaciones diferenciales*, Departamento de Matemática Aplicada IV, Barcelona España, Octubre 2008. [Documento en línea]. <http://www-ma4.upc.edu/nrr/docs/edteor.pdf>.

Para citar este artículo: Larry Theran et al . 2014, "Desarrollo de una GUI en MATLAB Para la Simulación del Fenómeno de Refracción por FDTD". Disponible en Revistas y Publicaciones de la Universidad del Atlántico en <http://investigaciones.uniatlantico.edu.co/revistas/index.php/MATUA>.