Aplicación del método de diferencias finitas en el dominio del tiempo a la simulacion del campo electromagnetico usando Matlab

M.A. Lopez-Esquer
Programa de posgrado en Ciencias (Fisica) de la Universidad de Sonora, Calle Rosales y Boulevard Luis Encinas, Hermosillo, Sonora 83000, Mexico.

J. Gaspar-Armenta y J. Manzanares-Martinez
Centro de Investigación en Física de la Universidad de Sonora, Apartado Postal 5-088, Hermosillo, Sonora 83190, México.

Recibido el 2 de junio de 2005; aceptado el 4 de agosto de 2005

En este artículo se describe una aplicación del método de diferencias finitas en el dominio del tiempo (MDFDT) para simular la propagación del campo electromagnético en un medio homogéneo. La estrategia de este método se basa en escribir las ecuaciones diferenciales de Maxwell en una formulación discreta en el espacio y en el tiempo. El objetivo es trasladar esta formulación a un algoritmo que pueda ser resuelto numéricamente por medio de un código computacional. Este trabajo pretende motivar a los estudiantes a la simulación numérica de fenómenos electromagnéticos ondulatorios. Por medio de lenguaje Matlab generamos un código que, con modificaciones fáciles, permite obtener animaciones que simulan diferentes situaciones físicas.

Descriptores: Diferencias finitas; electromagnetismo; simulación; Matlab.

In this paper we describe an application of the Finite Difference Time Domain (FDTD) method to simulate the propagation of the electromagnetic field in a homogeneous medium. The strategy of this method is to formulate the differential Maxwell equations in space and time finite differences in order to write a computational code. This work tries to motivate the undergraduate students to the numerical simulation of the electromagnetic wave propagation. We propose an interactive matlab code that with easy modifications is able to simulate different physical situations.

Keywords: Finite difference; electromagnetism; simulation; Matlab.

PACS: 02.70.Bf; 42.25.H2; 42.70.Qs; 41.20.-9

1. Introducción

En teoría, cuando un campo electromagnético incide sobre una estructura de una forma arbitraria, las ecuaciones de Maxwell pueden ser utilizadas para conocer el valor exacto de la reflexión en cualquier punto del espacio. Sin embargo, una solución analítica de la distribución del campo está restringida a un cierto número de geometrías; aquellas en donde es posible realizar los métodos tradicionales de solución de ecuaciones diferenciales. Usualmente para obtener soluciones analíticas, el fenómeno de dispersión del campo electromagnético se resuelve aplicando condiciones a la frontera y realizando separación de variables. Desafortunadamente este procedimiento sólo es válido para unas pocas geometrías. En consecuencia, cuando la geometría se vuelve más complicada, sólo es posible conocer el valor del campo por medio de métodos computacionales.

Durante las últimas décadas han sido desarrollados diversos métodos para resolver numéricamente las ecuaciones de Maxwell. Entre estas técnicas se encuentra el método de diferencias finitas en el dominio del tiempo (MDFDT) [1, 2]. Actualmente, el MDFDT es ampliamente utilizado para la descripción del campo electromagnético en diversas situaciones físicas y en especial en el campo de los cristales fotónicos [3].

Este articulo está dedicado a hacer una presentación sencilla de este método. Discutimos la formulación del MDFDT para simular la propagación del campo electromagnético en un medio homogéneo. Ciertamente, la propagación en un medio homogéneo o en el vacío no es un problema que implique un gran interés físico. Sin embargo, el objetivo principal de este trabajo es la ilustración del algoritmo.

Nuestra presentación es apropiada para los estudiantes de licenciatura o ingeniería que han llevado un curso intermedio [4] de electromagnetismo en donde se hayan presentado las ecuaciones de Maxwell. La experiencia computacional requerida para leer este trabajo es mínima. Nuestra idea es que el código sea fácil de entender, de tal forma que el estudiante desarrolle una buena intuición del fenómeno físico y de la lógica del programa en Matlab. Cabe destacar que en esta presentación tratamos de exponer en la forma más clara que nos ha sido posible el algoritmo del MDFDT. Hemos tomado en cuenta que probablemente el estudiante tendrá un par de obstáculos para entender el contenido de este trabajo: la inexperiencia en la formulación discreta de las ecuaciones diferenciales de Maxwell y la inexperiencia en programación.

En este trabajo nos restringimos a la descripción del campo en un medio homogéneo, en particular el vacío. En nuestro algoritmo sólo son necesarios dos vectores para indexar los valores del campo eléctrico y magnético. La presentación grafica de estos valores de los campos se muestra mediante una evolución en el tiempo gracias a nuestro programa en Matlab. En la Sec. 2 hacemos una presentación básica de la formulación discreta de las ecuaciones de Maxwell. En la

Sec. 3 discutimos la implementación del algoritmo en Matlab. En la Sec. 4 discutimos las condiciones absorbentes de frontera. Finalmente en la Sec. 5 ilustramos la flexibilidad del algoritmo implementando diferentes situaciones físicas.

2. Las ecuaciones de Maxwell en forma discreta

EL MDFDT analiza el problema de la propagación electromagnética en pequeñas particiones espaciales. En estas celdas los campos eléctricos y magnéticos están alternadamente distribuidos. Las celdas o nodos también están intercaladas en el tiempo. Para resolver este conjunto de ecuaciones se desarrolla un proceso iterativo en el tiempo.

Comenzamos con las ecuaciones de Maxwell en el sistema centímetro-gramo-segundo (cgs) [4]. Consideramos que las ondas se propagan en un medio homogéneo y que no existen fuentes de carga ni de corrientes ($\rho = \boldsymbol{J} = 0$). Si escogemos una polarización tal que el campo eléctrico ($\boldsymbol{E}$) sea paralelo al eje x y que se propague en la dirección z, entonces el campo magnético ($\boldsymbol{H}$) está sobre el eje y. Las ecuaciones rotacionales de Maxwell quedan de la siguiente manera

$$\frac{\partial}{\partial t}E_x(z,t) = -\frac{c}{\varepsilon}\frac{\partial}{\partial z}H_y(z,t), \tag{1}$$

$$\frac{\partial}{\partial t}H_y(z,t) = -\frac{c}{\mu}\frac{\partial}{\partial z}E_x(z,t). \tag{2}$$

La primera de estas ecuaciones es también llamada de Ampère-Maxwell y la segunda se conoce como la ecuación de Faraday. Ambas son *ecuaciones diferenciales puntuales*, es decir, que son validas para cada valor continuo del espacio y tiempo. Para lograr una formulación discreta, aproximamos las derivadas espacial y temporal por su representación en diferencias finitas. Recordemos la definición de derivada en diferencias centrales [5]:

$$\frac{df(z_0)}{dz} = \lim_{\Delta z \to o}\frac{f(z_o + \Delta z/2) - f(z_o - \Delta z/2)}{\Delta z}. \tag{3}$$

En esta ecuación, para un valor finito de Δz,la ecuación diferencial se transforma en una ecuación de diferencias finitas. La derivada está definida por el valor de la función en puntos discretos contiguos a z_o.

La estrategia del MDFDT consiste en cambiar los valores continuos de la coordenada espacial 'z' por valores discretos indexados por el número entero 'k', de tal forma que los valores espaciales son obtenidos por medio de la ecuación $z = k\Delta z$. Con estas consideraciones, la forma discreta de la derivada espacial para el campo eléctrico (E) es

$$\frac{\partial}{\partial z}E_x(k\Delta z, n\Delta t) \cong \frac{E_x[(k+1/2)\Delta z, n\Delta t] - E_x[(k-1/2)\Delta z, n\Delta t]}{\Delta z}. \tag{4}$$

Similarmente, los valores discretos de la coordenada temporal están indexados por el entero 'n' por medio de la relación $t = n\Delta t$ y la forma discreta temporal queda

$$\frac{\partial}{\partial t}E_x(k\Delta z, n\Delta t) \cong \frac{E_x[k\Delta z, (n+1/2)\Delta t] - E_x[k\Delta z, (n-1/2)\Delta t]}{\Delta t}. \tag{5}$$

Relaciones similares pueden obtenerse para el campo H_y. Al implementar las ecuaciones en diferencias finitas para $E_x(z,t)$ y $H_y(z,t)$en la Ec. (1) se obtiene forma discreta de la ecuación de Ampère-Maxwell:

$$\frac{E_x^{n+1/2}(k) - E_x^{n-1/2}(k)}{\Delta t} = -\frac{c}{\varepsilon}\frac{H_y^n(k+1/2) - H_y^n(k-1/2)}{\Delta z}. \tag{6}$$

Para simplificar la notación, hemos puesto como superíndice el índice discreto temporal, por ejemplo $E_x(n+1/2,k) \to E_x^{n+1/2}(k)$.

Para obtener la forma discreta de la Ec. (2) vamos a considerar las derivadas en el punto $(z+\Delta z/2, t+\Delta t/2)$. La ecuación de Faraday en su forma discreta, es entonces,

$$\frac{H_y^{n+1}(k+1/2) - H_y^n(k+1/2)}{\Delta t} = -\frac{c}{\mu}\frac{E_x^{n+1/2}(k+1) - E_x^{n+1/2}(k)}{\Delta z}. \tag{7}$$

Al considerar la Ec. (1) en el punto (z,t) y la Ec. (2) en el punto $(z+\Delta z/2, t+\Delta t/2)$ logramos describir una situación en donde los campos están intercalados:

$$E_x^{n+1/2}(k) = E_x^{n-1/2}(k) - \frac{c}{\varepsilon}\frac{\Delta t}{\Delta z}\left[H_y^n(k+1/2) - H_y^n(k-1/2)\right], \tag{8}$$

$$H_y^{n+1}(k+1/2) = H_y^n(k+1/2) - \frac{c}{\mu}\frac{\Delta t}{\Delta z}\left[E_x^{n+1/2}(k+1) - E_x^{n+1/2}(k)\right]. \tag{9}$$

En cada punto del espacio, el valor del campo se obtiene de los valores de los campos vecinos. Esta forma de escribir las ecuaciones de Maxwell es ilustrada en la Fig. 1. El campo $E_x^{n+1/2}(k)$ *es inducido* por el valor del campo un paso temporal anterior $E_x^{n-1/2}(k)$ *más* la contribución de los campos magnéticos $H_y^n(k+1/2)$ y $H_y^n(k-1/2)$. Estos campos están en puntos espaciales contiguos y un medio tiempo espacial anterior.

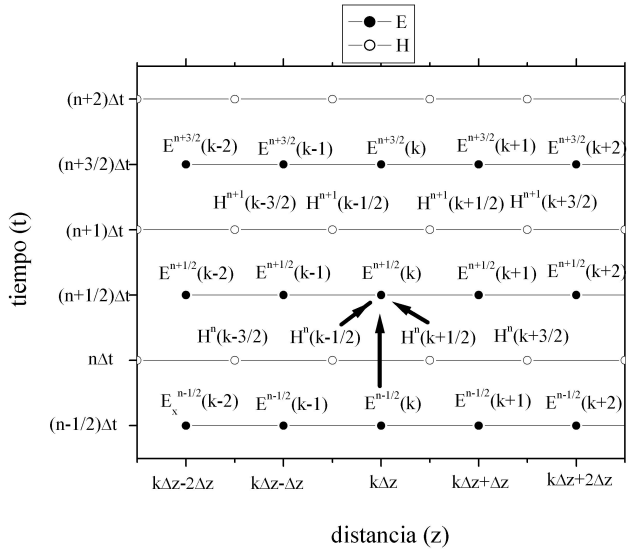

FIGURA 1. Representación de la evolución espacio-temporal de un campo electromagnético.

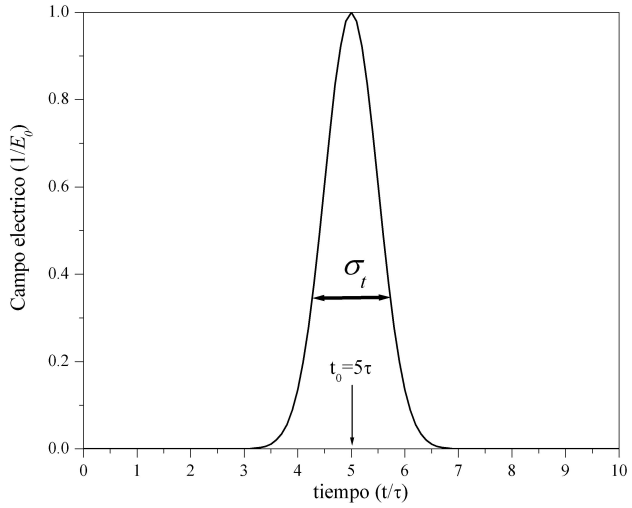

FIGURA 2. Ilustración de un pulso gaussiano, cuyo centro esta en t = 5τ y tiene un espesor σ_τ.

En este punto, es necesario determinar las cantidades que afectan a los corchetes en las Ecs. (8) y (9). En el vacío, la función dieléctrica y la permeabilidad magnética son iguales a 1; $\varepsilon = \mu$ =1. Una onda electromagnética necesita un mínimo de tiempo $\Delta t = \Delta z/c$ para propagarse dentro de la malla discreta espacial Δz. Si tomamos un paso temporal más grande, la onda ya habría pasado la distancia Δz *y ya no estaría en nuestra malla de simulación.* Por esta razón es necesario escoger un paso temporal de la forma [2,3,6]

$$\Delta t \leq \frac{\Delta z}{c}. \tag{10}$$

En este trabajo vamos a utilizar la convencion de escoger un paso temporal $\Delta t = \Delta z/2c$. Por su parte, el paso espacial lo escogemos tomando 10 puntos de la longitud de onda de la luz en el vacío λ_0:

$$\Delta z = \frac{\lambda_0}{10}. \tag{11}$$

3. Implementación en Matlab

El primer campo electromagnético que vamos a simular es un pulso gaussiano que tiene la forma

$$E(t) = \begin{cases} 0, & t < 0, \\ E_o e^{-\frac{1}{2}\left(\frac{t-t0}{\sigma_t}\right)^2}, & t > 0. \end{cases}$$

Ésta es una función temporal que asemeja un pulso de luz y esta ilustrada en la Fig. 2. Tiene la característica de que su amplitud máxima es E_0. Sólo toma valores diferente de cero en la vecindad de t_0, tiempo al cual está centrada y tiene una duración media $\sigma_{t.}$

La implementación de las Ecs. (8) y (9) en un algoritmo computacional requiere dos vectores unidimensionales, uno para E_xy otro para H_y, para describir la coordenada espacial. En cada tiempo 'n' los campos son calculados en Matlab mediante las *líneas de programa* son los siguientes:

ex(k) = ex(k) + 0.5*(hy(k-1) – hy(k)) (12)

hy(k) = hy(k) + 0.5*(ex(k) – ex(k+1)) (13)

En estas *relaciones* ya no existe el superíndice. El tiempo es una variable implícita en el MDFDT. En la primera de estas ecuaciones, el nuevo valor de ex(k) a la izquierda de la igualdad [para el tiempo $(n + 1/2)$] proviene del antiguo valor ex(k) [al tiempo$(n-1/2)$] que la máquina ya ha calculado y tenía en la memoria *más* los campos magnéticos hy(k-1)y hy(k) [propios del tiempo n]. Para el vector hy los índices se han redondeado de los índices k+1/2 y k-1/2 a k y k-1 para poder especificar la posición de los campos en el arreglo computacional.

En el Apéndice A mostramos el programa 'simple.m' que es nuestra primera implementación del algoritmo MDFDT en Matlab. Ya que el MDFDT es un proceso iterativo, puede ser presentado en forma concisa en un resumen de pasos lógicos o pseudo-código:

- ***Definición de parámetros***
- ***Iteración principal sobre el tiempo***
  - ***Calculo del campo eléctrico para todos los puntos espaciales***
  - ***Inserción de una fuente de campo eléctrico***
  - ***Calculo del campo magnético para todos los puntos espaciales***
  - ***Graficación***

Fin de la iteración principal

El programa comienza inicializando constantes y parámetros. Los campos eléctricos y magnéticos son inicializados a cero. El cálculo comienza obteniendo los campos eléctricos en cada nodo usando la Ec. (11). A continuación, se introduce el valor de la fuente de campo eléctrico. El siguiente paso es calcular los campos magnéticos. Finalmente,

el programa itera el procedimiento al siguiente valor temporal.

En la Fig. 3 mostramos el resultado del programa para 200 iteraciones temporales. El comportamiento del pulso en el espacio y en el tiempo puede ser visualizado en la Fig. 4. A medida que el pulso se propaga en el tiempo, se extiende hacia fuera del centro de la malla espacial.

El programa detiene la simulación antes de tocar el límite de la malla de simulación. En nuestro método, para calcular el campo **E** en un punto, es necesario conocer los valores **H** que rodean dicho punto. Sin embargo, en el borde de la malla de simulación no conocemos el punto correspondiente que está fuera de la malla. De esta forma, no es posible calcular el campo **E** en los valores límites de la malla.

Si el programa continuase, es decir, si hacemos el ejercicio de modificar el número de pasos temporales a N_t=200, veremos la evolución temporal de los valores espurios que toma el campo en el borde de la malla. El efecto neto es una reflexión hacia el interior de la malla. Esto es algo que es necesario evitar. Por ello introducimos las condiciones de frontera absorbentes en la siguiente sección.

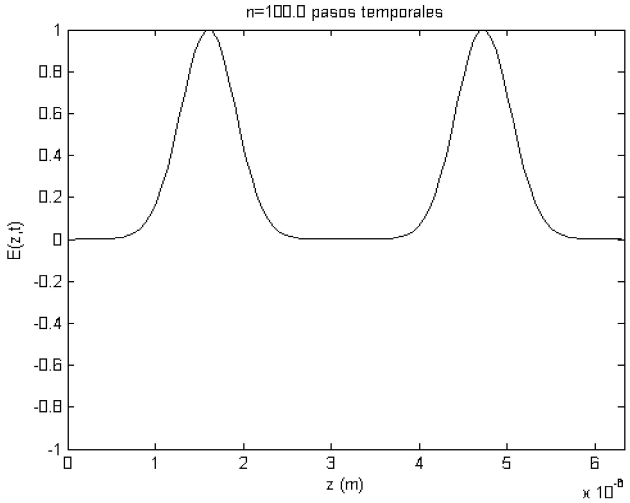

FIGURA 3. Pulso gaussiano despues de 100 pasos temporales.

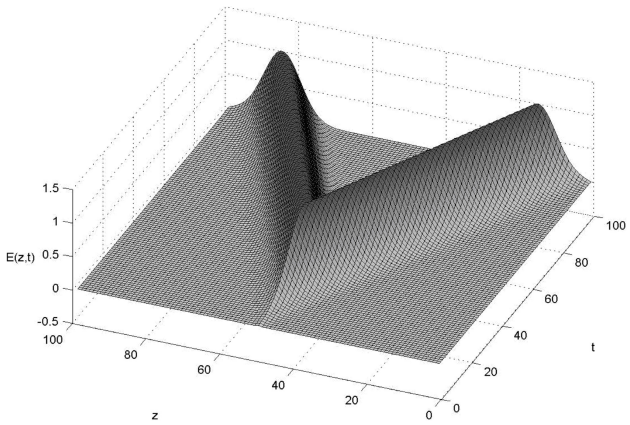

FIGURA 4. Variación del puslo gaussiano en el espacio y en el tiempo.

4. Condiciones de frontera absorbentes

Las condiciones de frontera absorbentes son necesarias para simular adecuadamente la propagación de los campos electromagnéticos una vez que éstos han llegado a la frontera de nuestra malla de simulación. En el algoritmo del MDFDT, los valores de los campos se determinan mediante un promedio de los campos en los puntos vecinos. El problema consiste en que en la frontera de la malla de simulación, este promedio no se puede dar porque no conocemos el valor del campo fuera de la malla. De esta forma, si la simulación continua, los campos toman valores espurios.

Físicamente, esperamos que los campos se propaguen hacia fuera de la malla de simulación, ya que consideramos que fuera de nuestra malla no existen fuentes. La distancia que la onda viaja en un paso temporal está determinada por la relación

$$distancia = c_0 \Delta t = c_0(\Delta z/2c_0) = \Delta z/2. \qquad (14)$$

Esta relación implica que al campo le toma un par de pasos temporales viajar un paso espacial. La forma de escribir esta condición es

$$E_x^n(1) = E_x^{n-2}(2). \qquad (15)$$

Con esta expresión estamos asignando el valor del campo en la frontera en lugar de calcularlo. Esta condición es fácil de implementar. Solamente es necesario guardar el valor de $E_x(2)$ un par de pasos temporales y luego asignarlos a $E_x(1)$.Las condiciones de frontera han sido implementadas en el programa 'simple_ABC.m' que se encuentra en el Apéndice B. En la Fig. 5 mostramos la propagación de una onda gaussiana. En esta ocasión, la onda pasa la frontera sin reflexión, que es lo que es esperado que ocurra.

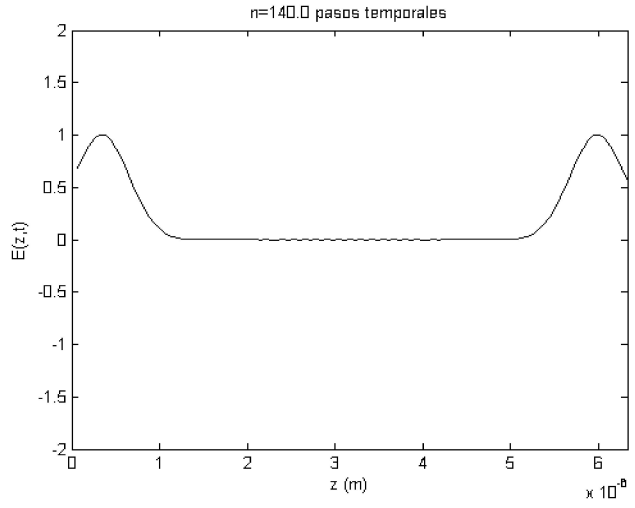

FIGURA 5. Pulso gaussiano despues de 140 pasos temporales incluyendo condiciones absorbentes de frontera.

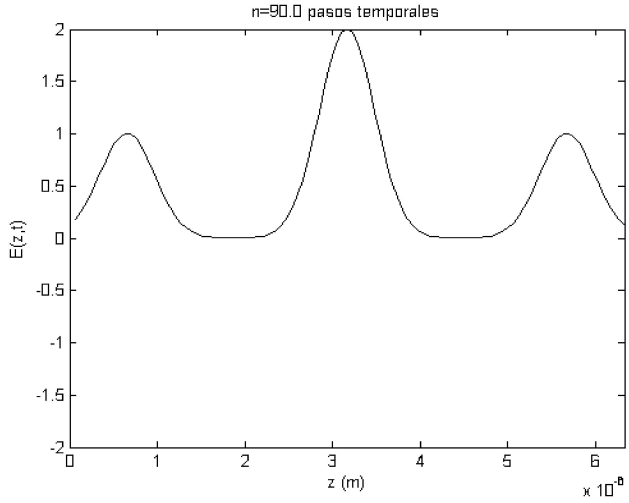

FIGURA 6. Superposición de dos pulsos gaussianos.

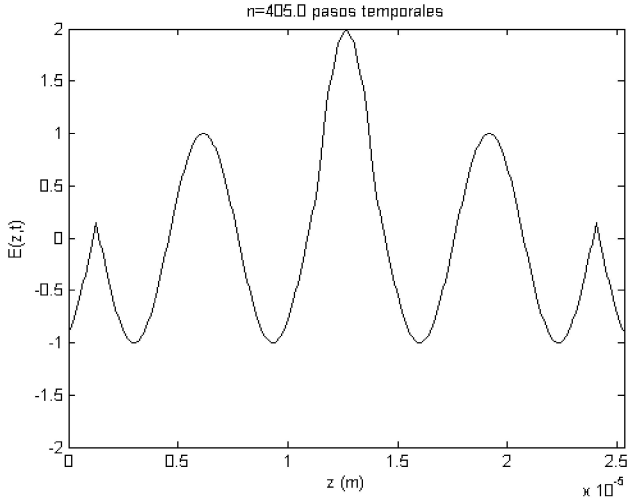

FIGURA 7. Superposición de dos ondas sinusoidales.

5. Aplicaciones

Uno de las ventajas más significativas del MDFDT es que es flexible para realizar simulaciones tanto de campos estacionarios como de pulsos que varían en el tiempo. También es posible simular tanto una sola frecuencia, por ejemplo una onda sinusoidal o un pulso que contenga varias frecuencias. A continuación veremos algunos ejemplos de superposición de diferentes tipos de fuentes. Estos ejemplos pretenden ilustrar la flexibilidad del algoritmo.

5.1. Superposición de dos pulsos gaussianos

El primer ejemplo de modificación del programa es obtenido al analizar la superposición de dos pulsos gaussianos. Uno de los pulsos está ubicado en k=30 y el segundo está ubicado en k=70. En la Fig. 6 ilustramos la superposición de ambos pulsos después de 90 pasos temporales.

Para lograr esta simulación es necesario escribir una nueva función llamada 'pulso_2gauss.m' la cual mostramos en el Apéndice C. Asimismo, hay que modificar una línea del programa 'simple_ABC.m'. La línea 28 donde viene la instrucción

ex=pulso(ex,n,t,tao,t0,spread);

debe ser cambiada por

ex=pulso_2gauss(ex,n,t,tao,t0,spread);

Con esta variación estamos llamando a la nueva función 'pulso_2gauss.m' en vez de a la función 'pulso.m'.

5.2. Superposición de dos ondas sinusoidales

Este ejemplo requiere más cambios sobre el programa 'simple_ABC', Ahora vamos a requerir de una malla espacial más grande, N_z=400 y las iteraciones temporales van a ser N_t=350. La onda sinusoidal que vamos a simular es

$$E(t) = \sin\left(0{,}1\frac{2\pi}{\tau}t\right).$$

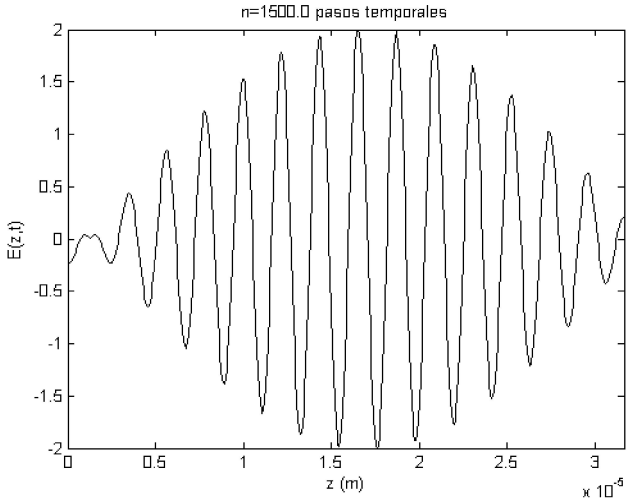

FIGURA 8. Evolución de la suma de dos ondas sinuosidades ligeramente desfasadas.

Esta fuente va a ser ubicada en k=20 y k=380. Este cambio está contenido en la función 'pulso_2sin.m', la cual se llama dentro del programa modificando la línea 28 donde viene la instrucción

ex=pulso(ex,n,t,tao,t0,spread);

debe ser cambiada por

ex=pulso_2sin(ex,n,t,tao,t0,spread);

En la Fig. 7 mostramos la superposición de dos ondas después de 405 pasos temporales.

5.3. Evolucion de un grupo de ondas

Como un ultimo ejemplo vamos a presentar la evolución de un grupo de ondas ligeramente desfasadas que obedecen a la función:

$$E(t) = \sin\left(0{,}3\frac{2\pi}{\tau}t\right) + \sin\left(0{,}28\frac{2\pi}{\tau}t\right)$$

Similarmente, la línea 28, donde viene la instrucción

```
ex=pulso(ex,n,t,tao,t0,spread);
```

debe ser cambiada por

```
ex=pulso_2group(ex,n,t,tao,t0,spread);
```

En este ejemplo consideramos una malla de tamaño N_z=500 y en la Fig. 8 mostramos la situación después de 1500 pasos temporales.

6. Conclusiones

Hemos presentado una implementación sencilla del MDFDT utilizando en forma elemental las características de Matlab. Este algoritmo puede ser trasladado fácilmente a Fortran o C++, en caso de ser necesario por no contar con Matlab. Esta breve ilustración del método pretende motivar a los estudiantes, mediante la presentación de ejemplos, de las potencialidades de este algoritmo. Este trabajo puede también verse como una actividad suplementaria a un curso de electromagnetismo. La flexibilidad del programa al introducir los diferentes casos mediante funciones, ayuda al estudiante a explorar nuevas posibilidades de aplicación del algoritmo. Finalmente, esperamos que la metodología expuesta en este trabajo interese a los estudiantes para adquirir más conocimientos en la técnica del MDFDT.

Agradecimientos

J.M.M. agradece el apoyo económico del proyecto CONACYT 44066 y del proyecto DCEN-UNISON Ref. PI 05/DCEN06.

Apéndice A

El primer programa es llamado ‘simple.m’ y el código es presentado a continuacion.

```
% simple.m
clear;
% Definicion de parametros
c = 3.0E+8; % velocidad de la luz
lo = 632.8E-9; % longitud de onda del laser
tao = lo/c; % periodo
N_z = 100; % particiones en z
N_t = 100; % pasos temporales
delta_z = lo/10 % particion espacial
delta_t = delta_z/(2*c) % particion temporal
t = 0.0 % valor inicial del tiempo
zf = N_z*delta_z/1e-6 % valor final de z (micras)
ex(1:N_z) = 0; % inicializa valores del campo
hy(1:N_z) = 0; % inicializa valores del campo
z = delta_z*(1:N_z); % inicializa valores de z
t0 = 2.5*tao;
spread = 0.5*tao;
for n = 1: N_t % inicia ciclo MDFDT
for k = 2: N_z-1
ex(k) = ex(k) + 0.5*(hy(k-1)-hy(k));
end
ex(50) = ex(50)+exp(-0.5*((t-t0).*(t-t0)/spread 2));
for k = 1 : N_z-1
hy(k) = hy(k)+0.5*(ex(k)-ex(k+1));
end
axis([0 zf -1 1])
plot(z/1e-6,ex)
axis([0 zf -1 1])
xlabel('z (micras)')
ylabel('E(z,t)')
title(['n=',num2str(n,' %4.1f'),'pasos temporales'])
frame = getframe(gca);
t = n* delta_t;
t_v(n) = t/tao;
pulse_v(n) = exp(-0.5*((t-t0).*(t-t0))/spread 2);
end % termina MDFDT
```

Apendice B

La segunda versión del algoritmo es el programa ‘simple_ABC.m’, el cual incorpora las condiciones de frontera y, además, el pulso es introducido por medio de una función llamada ‘pulso.m’, esta función debe de estar en el mismo subdirectorio que el programa ‘simple_ABC.m’

```
% simple_ABC.m
clear;
% Definicion de parametros
c = 3.0E+8; % velocidad de la luz
lo = 632.8E-9; % longitud de onda del laser
tao = lo/c; % periodo
N_z = 100; % particiones en z
N_t = 140; % pasos temporales
delta_z = lo/10 % particion espacial
delta_t = delta_z/(2*c) % particion temporal
t = 0.0 % valor inicial del tiempo
zf = N_z*delta_z % valor final de z
ex(1:N_z) = 0; % inicializa campo ex
hy(1:N_z) = 0; % inicializa campo hy
z = delta_z*(1:N_z); % inicializa valores de z
t0 = 2.5*tao;
spread = 0.5*tao;
ex_low1 = 0;
ex_low2 = 0;
ex_high2 = 0;
ex_high1 = 0;
for n = 1: N_t % inicia ciclo MDFDT
for k = 2: N_z-1
ex(k) = ex(k) + 0.5*(hy(k-1)-hy(k));
end
ex=pulso(ex,n,t,tao,t0,spread);
% condiciones absorbentes
ex(1) = ex_low2;
ex_low2 = ex_low1;
ex_low1 = ex(2);
ex(N_z) = ex_high2;
ex_high2 = ex_high1;
ex_high1 = ex(N_z-1);
```

```
for k = 1 : N_z-1
hy(k) = hy(k)+0.5*(ex(k)-ex(k+1));
end
axis([0 zf -2 2])
plot(z,ex)
axis([0 zf -2 2])
xlabel('z (m)')
ylabel('E(z,t)')
title(['n=',num2str(n,'%4.1f'),' pasos temporales'])
frame = getframe(gca);
t = n* delta_t;
end % termina MDFDT
```

A continuación mostramos el programa 'pulso.m', el cual introduce el mismo pulso que el caso 'simple.m'

```
% pulso.m
function campo=pulso(ex,n,t,tao,t0,spread)
ex(50) = ex(50)+exp(-0.5*((t-t0).*(t-t0)/spread 2));
campo = ex;
```

La ventaja de introducir una funcion, es que nos permite flexibilizar el algoritmo. Ahora, los casos de estudio se obtendran mediante la utilización de diferentes funciones sin modificar el programa principal. Éste es un concepto importante en la programación estructurada.

Apéndice C

```
% pulso_2gaus.m
function campo=pulso_2gauss(ex,n,t,tao,t0,spread)
ex(30) = ex(30)+exp(-0.5*((t-t0).*(t-t0)/spread 2));
ex(70) = ex(70)+exp(-0.5*((t-t0).*(t-t0)/spread 2));
campo = ex;
% pulso_2sin.m
function campo=pulso_2sin(ex,n,t,tao,t0,spread)
ex(20) = ex(20)+sin(0.1*2*pi*t/tao);
ex(380) = ex(380)+sin(0.1*2*pi*t/tao);
campo = ex;
%pulso_2group.m
function campo=pulso_2group(ex,n,t,tao,t0,spread)
ex(20) = ex(20)+sin(0.3*2*pi*t/tao)+sin(0.28*2*pi*t/tao);
campo = ex;
```


1. A. Taflove, *Computational Electrodynamics: The Finite-Difference Time-Domain Method* (Artech House, Boston, 1995).
2. D.M. Sullivan, *Electromagnetic simulation using the FDTD method* (IEEE Press Series on MIcrowave technology, New York, 1994).
3. K. Busch, S. Lolkes, R.B. Wehrspohn, and H. Foll, *Photonic crystals* (Wiley, New York, 2004).
4. J.R. Reitz, *Foundations of Electromagnetic Theory* (Addison Wesley, NY, 1992).
5. L. Leithold, *Calculus and Analytical Geometry* (Harpercollins, NY, 1997).
6. K.S. Kunz y R.J. Luebbers, *The Finite Difference Time Domain Method for Electromagnetics* (CRC Press, Boca Raton, 1993).