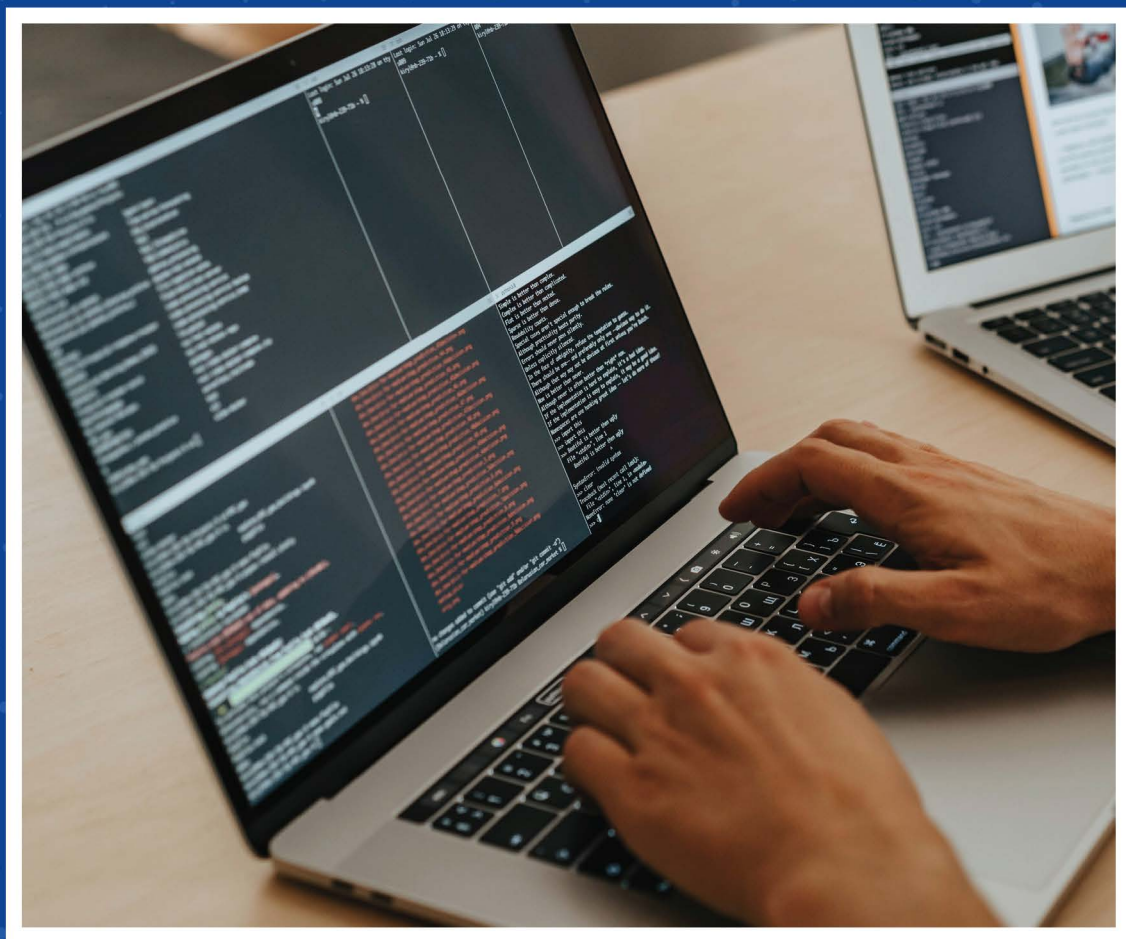


GUÍA DE CONSTRUCCIÓN DE UN ALGORITMO BASADO EN EL MÉTODO DE DIFERENCIAS FINITAS EN DOMINIO TEMPORAL (FINITE DIFFERENCE TIME DOMINE - FDTD) PARA SU APLICACIÓN EN EL ESTUDIO DE LAS PROPIEDADES ÓPTICAS DE CRISTALES FOTÓNICOS 2D



COLECCIÓN MATERIAL DOCENTE INGENIERÍAS



Danny Manuel Calvo Velasco.
Docente del programa de
Ingeniería de Sistemas de
Unicomfauca en Santander de
Quilichao. Físico, Magister en
Ciencias Físicas y Doctor
en Ciencias Físicas de la
Universidad del Valle.

**GUÍA DE CONSTRUCCIÓN DE UN ALGORITMO
BASADO EN EL MÉTODO DE DIFERENCIAS FINITAS
EN DOMINIO TEMPORAL (*FINITE DIFFERENCE TIME
DOMAIN* – FDTD) PARA SU APLICACIÓN EN EL
ESTUDIO DE LAS PROPIEDADES ÓPTICAS DE
CRISTALES FOTÓNICOS 2D**

Danny Manuel Calvo Velasco



Catalogación en la publicación – Biblioteca Nacional de Colombia

Calvo Velasco, Danny Manuel

Guía de construcción de un algoritmo basado en el método de diferencia finitas en dominio temporal (Finite Difference Time Domain – FDTD) para su aplicación en el estudio de las propiedades ópticas de cristales fotónicos 2D./ Danny Manuel Calvo Velasco. ---1ª ed.--- Popayán: Sello editorial Unicomfacaucá, 2022.

p. 77

Contiene datos de los autores.

ISBN (Digital): 978-628-95397-1-4

Guía de construcción de un algoritmo basado en el método de diferencia finitas en dominio temporal (Finite Difference Time Domain – FDTD) para su aplicación en el estudio de las propiedades ópticas de cristales fotónicos 2D.

© Corporación Universitaria Comfacaucá – Unicomfacaucá

© Autor: Danny Manuel Calvo Velasco

Primera edición en español

Sello Editorial Unicomfacaucá, septiembre de 2022

ISBN Digital: 978-628-95397-1-4

Diseño Editorial: Sello Editorial Unicomfacaucá – Corporación Universitaria Comfacaucá - Unicomfacaucá

Corrección de Estilo: La Peregrina Estudio – larailustracion@gmail.com

Diagramación: La Peregrina Estudio – larailustracion@gmail.com

Editor General de Publicaciones: Julio Eduardo Mejía Manzano

Sello Editorial Unicomfacaucá

Calle 4ta # 8-30 Centro Histórico

Popayán, Colombia

Teléfono: 602 8386000 Ext. 118

www.unicomfacaucá.edu.co/investigacion/sello-editorial/



Licencia Atribución-NoComercial-SinDerivadas 2.5 Colombia (CC BY-NC-ND 2.5 CO)

Contenido

Resumen	9
Introducción	10
Fundamentación teórica	12
Fundamentación metodológica	12
Fundamentación curricular y didáctica	13
Capítulo 1: Método de diferencias finitas en dominio temporal (Finite Difference Time Domain – FDTD)	14
· Resumen del capítulo	14
· Diferencias finitas centradas	14
· Descripción de las fórmulas para los modos de propagación TE y TM	15
· Implementación del método FDTD al caso no dispersivo	19
· Forma de los algoritmos y su implementación en arreglos computacionales	21
Capítulo 2: Análisis espectral	28
· Resumen del capítulo	28
· Transformada Discreta de Fourier (Discret Fourier Transformation – DFT)	28
· Análisis espectral sin considerar un arreglo para la parte temporal	30
· Anotaciones sobre el uso de archivos en Fortran	31
· Implementación para el cálculo de los modos propios del sistema variando el vector de onda	32
· Análisis del espectro de frecuencias comparando los picos de resonancia de los espectros para diversos puntos usando la transformada discreta de Fourier	35
· Implementación FDTD y análisis DFT	37
Capítulo 3: Método FDTD aplicado a sistemas dispersivos	40
· Resumen del capítulo	40
· Respuesta material dispersiva dieléctrica arbitraria	40
· Método FDTD al considerar la dispersión tipo Drude, materiales metálicos	44
· Método FDTD aplicado a sistemas metamateriales	46
Capítulo 4: Método FDTD utilizando un esquema de derivadas, otra deducción	48
· Resumen del capítulo	48
· Formulación	48
Anexos: Algoritmos desarrollados	54
· Consideraciones para la implementación algorítmica	54
· Algoritmo FDTD aplicado al modo TE	54
· Algoritmo FDTD aplicado al modo TM	58
· Algoritmo FDTD para el modo TE al considerar dispersión tipo Drude	62
· Algoritmo FDTD para el modo TE al considerar la dispersión tipo Drude y múltiples puntos	67
· Algoritmo DFT para comparar varios puntos	71
Referencias	74

RESUMEN

Esta guía tiene el propósito de introducir a los estudiantes al estudio de las simulaciones numéricas, por medio de la construcción de algoritmos basados en el método de diferencias finitas, en el cual las funciones de interés evolucionan tanto en el tiempo como en el espacio. Se aplica el formalismo desarrollado a un problema específico de propagación de las ondas electromagnéticas en un cristal fotónico 2D, formado por barras paralelas de sección transversal circular dispuestas en un arreglo rectangular, contenidas en un material con diferente respuesta óptica. El texto describe cómo las propiedades ópticas del cristal fotónico se pueden estudiar al tener en cuenta dos polarizaciones principales, denominadas transversal eléctrica (te) y transversal magnética (tm), desarrolla el formalismo de las ecuaciones del electromagnetismo de Maxwell y presenta su discretización y aplicación al mundo computacional. Además, se describe el análisis espectral de las fluctuaciones de los campos electromagnéticos, para hallar los modos de propagación permitidos en el cristal.

Palabras clave: diferencias finitas, cristales fotónicos, polarización, ecuaciones de Maxwell, transformada de Fourier.

INTRODUCCIÓN

El estudio de la propagación de la luz es uno de los problemas que más frutos ha brindado a las ciencias físicas. En particular, ha permitido el desarrollo de ideas y conceptos que han permeado diferentes áreas del conocimiento como la química, la biología, la teoría de campos entre otros (Barut, 1980; Chouhan *et al.*, 2016; Engheta y Ziolkowski, 2006; Jackson, 1998; Li *et al.*, 2008).

La importancia en la comprensión del comportamiento ondulatorio de la luz ha permitido el desarrollo de tecnologías que son aplicadas en diversidad de situaciones, desde la polarización de las imágenes en un cine 3D, hasta las mejoras en el desempeño de celdas solares basadas en distribuciones materiales que semejan estructuras fractales (Su *et al.*, 2013; Yao *et al.*, 2015).

Si bien la propagación de las ondas electromagnéticas se puede describir mediante las ecuaciones de Maxwell, su aplicación en diversos tipos de sistemas puede llevar a imposibilidades analíticas debido, en parte, a las condiciones de frontera que se deben satisfacer, así como a la diversidad de respuestas materiales que pasan de ser pasivas como en los medios dieléctricos, a ser activas, como en los metales, materiales ferromagnéticos y materiales no lineales que varían su respuesta óptica cuando son sometidos a campos electromagnéticos con intensidades muy altas, como los producidos por láseres (He y Cada, 1992; Kumar *et al.*, 2013; Zohrabi y Namdar, 2017). Esta diversidad de fenómenos hace que no exista una sola técnica para el estudio de la propagación de la luz. Algunas son propuestas con base en respuestas ideales a las cuales se les agregan perturbaciones o métodos numéricos en los cuales las soluciones son halladas al considerar pequeñas variaciones espaciales y temporales que se dejan evolucionar desde una condición inicial en la cual se simula la aplicación de energía al sistema (Kuzmiak *et al.*, 1994; Moreno *et al.*, 2002; Soto-Puebla *et al.*, 2004).

Además de ello, en años recientes el estudio de los cristales fotónicos ha cobrado mucho interés en la comunidad científica. Esto se debe a la cantidad de posibles aplicaciones de este tipo de estructuras, pues presentan brechas en su espectro de transmisión que son posibles de modificar y modular al tener en cuenta parámetros como: diferencia entre los índices de refracción de los materiales usados, disposición de los materiales en el espacio (geometría espacial) y agentes externos que actúan como controles en las propiedades eléctricas y magnéticas de los materiales (Huang *et al.*, 2003; Joannopoulos *et al.*, 2008). Algunas de las aplicaciones actuales de este tipo de sistemas se pueden hallar en dispositivos optoelectrónicos que funcionan como sensores, así como cristales fotónicos que son usados como sustratos para el control de moléculas ultra frías y guías de onda para la construcción de plataformas que faciliten la implementación de la computación cuántica (Engheta y Ziolkowski, 2006).

En este documento se pretende entregar al estudiante una guía para el desarrollo de un algoritmo basado en el método de diferencias finitas en dominio temporal para la simulación de la propagación de la luz en un cristal fotónico 2D. De esa manera, se mostrará cuales son los retos tanto de la descripción teórica de las ecuaciones de propagación de la luz, como de su implementación computacional en el lenguaje de programación Fortran, un lenguaje de bajo nivel diseñado para la optimización de cálculos numéricos.

Con este documento, el estudiante podrá ver como los aspectos técnicos de elaborar un algoritmo de simulación se basan en el conocimiento de la dinámica propia del sistema a estudiar, junto con las ideas matemáticas, desarrolladas en los cursos de cálculo vectorial y de ecuaciones diferenciales, que permiten su solución numérica. Los métodos aquí desarrollados pueden ser usados en cualquier tipo de simulación computacional en la cual se desee evaluar el comportamiento de campos vectoriales con variación espacial y temporal.

La estructura de este documento es la siguiente: en el segundo capítulo se describe en qué consiste el método de diferencias finitas centradas, con una explicación de la discretización de las ecuaciones de Maxwell siguiendo la sugerencia dada por Yee (1996), junto con el uso de la celda que lleva su nombre, la cual relaciona los campos eléctricos y magnéticos a partir del concepto de rotacional. Se describirán los pasos a seguir para la construcción del algoritmo que se implementará, junto con observaciones que serán útiles para la mejora de los primeros algoritmos. En el tercer capítulo, se describe cómo realizar una transformada de Fourier junto con su implementación algorítmica. En el capítulo cuarto, se describe la generalización del método de diferencias finitas en dominio temporal, para el caso de medios dispersivos con respuesta tipo metálica, así como de su aplicación en estructuras metamateriales. Por último, en el capítulo cinco se describe otra forma de discretización para los campos electromagnéticos que puede ser implementada en el método de diferencias finitas desarrollado.

Para complementar todo este documento, se anexan las formas básicas de los algoritmos en el lenguaje de programación Fortran para que los estudiantes encuentren una herramienta de la cual partir y que puedan modificar a su gusto.

FUNDAMENTACIÓN TEÓRICA

Para el desarrollo de las ideas presentadas en este documento son necesarios los conocimientos básicos de la teoría electromagnética, en especial los relacionados con el estudio de las ondas electromagnéticas en el espacio, junto con la descripción diferencial de las ecuaciones de Maxwell a las cuales se aplica el método de diferencias finitas. Además, se requieren conocimientos básicos de programación, sobre todo del diseño de algoritmos con bucles de repetición, los cuales son extensamente usados en el método aquí presentado.

Por otra parte, se espera que el estudiante maneje aspectos del cálculo diferencial e integral, vitales para la solución de las ecuaciones relacionadas con las respuestas materiales de los sistemas. Así mismo, al ser la radiación electromagnética una combinación de campos vectoriales, las soluciones propuestas están descritas en lenguaje vectorial, para lo cual es necesario tener bases sólidas de álgebra lineal.

Por último, se incita al estudiante a conocer el análisis espectral por medio de la transformada de Fourier, la cual es la técnica clave para el análisis de las señales simuladas, lo que permitirá conocer las propiedades de los sistemas estudiados por medio de la identificación de modos resonantes.

FUNDAMENTACIÓN METODOLÓGICA

En este documento se desarrollan las ideas con un esquema de: método de solución, ecuaciones a resolver e interpretación algorítmica. Todo esto les permitirá a los estudiantes, a través del ejemplo resuelto en estas páginas, comenzar a construir sus propios modelos de simulación, a partir de las ecuaciones diferenciales que definen su dinámica.

De manera específica se presentan las ideas aplicadas al ejemplo de cristales fotónicos en el siguiente orden:

- a. Método de diferencias finitas, el cual se aplicará como propuesta de solución de las ecuaciones diferenciales estudiadas.
- b. Ecuaciones de Maxwell en medios materiales sin presencia de cargas ni corrientes. Estas ecuaciones se resuelven numéricamente para el caso de un cristal fotónico 2D formado por barras paralelas de sección transversal circular dispuestas en un arreglo 2D rectangular.
- c. Estructura algorítmica para la búsqueda de la solución. Este apartado está relacionado con el desarrollo de los algoritmos a partir de la interpretación de las ecuaciones que se desea simular. Es importante destacar que este tipo de conocimiento es básico para que los estudiantes aprendan a representar ecuaciones dinámicas en el lenguaje computacional, con esto se demuestra cómo se puede convertir el lenguaje formal.

- d. Discusión de las dificultades que se enfrentan para la construcción del programa de simulación. Desde el punto de vista del desarrollo de las ideas presentadas, se discuten las necesidades propias que enfrentará el estudiante al implementar las soluciones propuestas. En especial, se menciona del manejo de memoria, cómo las ecuaciones se pueden resolver con el uso arreglos de múltiples dimensiones, así como la mejora de ciertos aspectos computacionales y de rendimientos si se hace uso de otros tipos de objetos como los archivos.
- e. Desarrollo del algoritmo de análisis espectral para el estudio de las propiedades ópticas de los sistemas simulados por medio de la transformada de Fourier. En este sentido, se hace especial énfasis en la descripción numérica de las integrales relacionadas con las transformada de Fourier cuando se pasa de una representación temporal a una representación en frecuencias.

Cabe destacar que al final del documento se presentan todos los algoritmos en el lenguaje Fortran, los cuales pueden ser aplicados por el estudiante. Para hacerlo no se requieren paquetes especiales ni de simulación, solo un compilador que puede ser descargado de manera gratuita en la red. Estos algoritmos se presentan en texto plano, para que el estudiante pueda explorar como se modifican las diferentes soluciones para las posibles condiciones que desee simular. Este aporte es fundamental para continuar un estudio de las propiedades de los sistemas fotónicos.

FUNDAMENTACIÓN CURRICULAR Y DIDÁCTICA

Este documento se ha pensado como una aplicación práctica de las ideas desarrolladas en los cursos de Física II, Programación, Calculo diferencial e integral de una y varias variables, Ecuaciones diferenciales y Algebra lineal. Presenta al estudiante un sistema que puede ser estudiado a partir de las ecuaciones diferenciales que lo definen y con un método de fácil implementación, pues desarrolla paso a paso los algoritmos necesarios para su estudio. Es esencial este tipo de ejercicios académicos, en cuanto permiten al estudiante tener una visión un poco más general del alcance de las ideas que se presentan en el aula de clase y con las cuales se consigue explicar las propiedades ópticas de sistemas como los cristales fotónicos sin la necesidad de ningún tipo de software especializado de simulación. Al principio muchas de estas ideas se presentan como un conjunto de cajas negras a las cuales no hay acceso.

Se espera este texto que pueda ser el comienzo de nuevas líneas de investigación y desarrollo en la institución y que los estudiantes se sientan atraídos por este tipo de propuestas.

CAPÍTULO 1:

Método de diferencias finitas en dominio temporal (*Finite Difference Time Domain – FDTD*)

Resumen del capítulo

En este capítulo se presentará el método de diferencias finitas en dominio temporal (*Finite Difference Time Domain – FDTD*) y su aplicación al caso de propagación de las ondas electromagnéticas en un cristal fotónico. Primero, se presentan las ecuaciones en diferencias finitas centradas, las cuales permiten la discretización de variaciones espaciales y temporales. Después, se presentan las ecuaciones de Maxwell para aplicarlas al caso de un cristal fotónico 2D, se desarrolla el formalismo para las dos polarizaciones principales con las cuales se estudian las propiedades ópticas del cristal, además se expondrá la discretización de las ecuaciones asociadas. Finalmente, se presenta la construcción de los respectivos algoritmos en el lenguaje de programación Fortran.

Diferencias finitas centradas

El método fdtd es una propuesta de solución de las ecuaciones del electromagnetismo de Maxwell en un medio material. Este método está basado en un sistema de ecuaciones en diferencias parciales, en las cuales se aproxima el valor de la derivada de los campos usando una aproximación de diferencias finitas centradas, todo aplicado sobre la celda de Yee (Sullivan, 2000; Yee, 1966).

Para este caso, se puede describir los campos eléctrico y magnético en dos modos o polarizaciones bien diferenciadas para la propagación de la luz, estas se denominan transversal eléctrico (te) y transversal magnético (tm), las cuales tienen asociado un sistema de ecuaciones que describe el comportamiento de los campos en un cristal fotónico 2D (Joannopoulos *et al.*, 2008).

Es importante resaltar que, al ser un cristal fotónico, existe una periodicidad en los operadores diferenciales que satisfacen los campos, cuyas soluciones pueden escribirse como

$$\vec{A}(\vec{r}) = \exp \exp (i\vec{k} \cdot \vec{r}) \vec{\xi}(\vec{r}) \quad (1)$$

En donde $\vec{A}(\vec{r})$ es un campo vectorial (que puede ser el campo eléctrico o magnético), $\vec{k}$ es el vector de onda de la radiación incidente y $\vec{\xi}(\vec{r})$ es una función periódica descrita en una pequeña región del cristal fotónico.

Por medio de la ecuación (1), la forma de la dependencia del campo vectorial en el cristal fotónico queda definida como

$$\vec{A}(\vec{r} + \vec{a}) = \exp (i\vec{k} \cdot \vec{a}) \vec{A}(\vec{r}) \quad (2)$$

en donde $\vec{a}$ es un vector que satisface la simetría de traslación del cristal fotónico. Esta última relación será la que defina las condiciones de frontera para hallar los modos de propagación de la luz al interior del cristal. A esta condición para los campos se la conoce como la condición de frontera de Bloch (Kittel, 1966).

Considere una función f que depende del tiempo y el espacio, $f(t, \vec{r})$. Para discretizar los valores de la función tanto espacial como temporalmente se debe considerar las siguientes representaciones

Un lugar del espacio (x, y, z) se describirá por los índices (i, j, k) .

Un lugar (i, j, k) estará separado de otro lugar $(i + 1, j + 1, k + 1)$ por las distancias Δx , Δy y Δz .

La dependencia temporal t se representará con un índice n .

La separación temporal entre un valor n y $n + 1$ será Δt .

Con estas consideraciones es posible describir el valor de la función en su representación discreta como

$$f(t, x, y, z) \rightarrow f^n(i, j, k) \quad (3)$$

Las variaciones de la función con respecto de cada una de las variables, al considerar una aproximación en diferencias finitas centradas, serán

$$\frac{\partial f(t, x, y, z)}{\partial t} \rightarrow \frac{f^{n+1/2}(i, j, k) - f^{n-1/2}(i, j, k)}{\Delta t} \quad (4)$$

$$\frac{\partial f(t, x, y, z)}{\partial x} \rightarrow \frac{f^n(i + 1/2, j, k) - f^n(i - 1/2, j, k)}{\Delta x} \quad (5)$$

$$\frac{\partial f(t, x, y, z)}{\partial y} \rightarrow \frac{f^n(i, j + 1/2, k) - f^n(i, j - 1/2, k)}{\Delta y}$$

$$\frac{\partial f(t, x, y, z)}{\partial z} \rightarrow \frac{f^n(i, j, k + 1/2) - f^n(i, j, k - 1/2)}{\Delta z}$$

Estas ecuaciones constituyen el núcleo del método FDTD y, por lo tanto, del algoritmo a desarrollar.

Descripción de las fórmulas para los modos de propagación TE y TM

Considérese un material con permitividad dieléctrica $\epsilon(\vec{r})$ y permeabilidad magnética $\mu(\vec{r})$, sin presencia de corrientes ni cargas libres. Aquí las ecuaciones de Maxwell (Joannopoulos et al., 2008) se pueden escribir de la siguiente manera:

$$\nabla \times \vec{E} = -\frac{1}{c} \frac{\partial \vec{B}}{\partial t}, \quad \nabla \times \vec{H} = \frac{1}{c} \frac{\partial \vec{D}}{\partial t} \quad (6)$$

Y debido a las propiedades del material las relaciones de los campos se pueden escribir como:

$$\vec{D}(\vec{r}) = \epsilon(\vec{r})\vec{E}(\vec{r}), \quad \vec{B}(\vec{r}) = \mu(\vec{r})\vec{H}(\vec{r}) \quad (7)$$

El método FDTD, se basa en la relación que existe entre los campos en el sistema de ecuaciones diferenciales y supone que el conocimiento del valor del campo $\vec{H}$ en un lugar del espacio depende de los valores del campo $\vec{E}$ a su alrededor y viceversa. Para poder hacer la discretización de las ecuaciones de Maxwell, Yee creó la celda que lleva su nombre y que muestra las relaciones que existen entre los campos en un lugar del espacio.

Para el modo de propagación te se puede escribir para los campos que

$$\vec{E} = (E_x, E_y, 0), \quad \vec{H} = (0, 0, H_z) \quad (8)$$

De forma tal que la ecuación (6) toma la forma

$$\nabla \times \vec{E} = \frac{-1}{c} \frac{\partial \vec{B}}{\partial t} \rightarrow \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} = -\frac{1}{c} \frac{\partial B_z}{\partial t} \quad (9)$$

$$\nabla \times \vec{H} = \frac{1}{c} \frac{\partial \vec{D}}{\partial t} \rightarrow \frac{\partial H_z}{\partial y} = \frac{1}{c} \frac{\partial D_x}{\partial t}, \quad \frac{\partial H_z}{\partial x} = -\frac{1}{c} \frac{\partial D_y}{\partial t} \quad (10)$$

Para aplicar el método ftdt a las ecuaciones (9) y (10) se debe escoger cuál de los campos se actualiza primero. Esto implica que se deben escribir las ecuaciones para los tiempos n y $n + \frac{1}{2}$, lo que corresponde a una variación temporal. En este caso se actualizarán la ecuación (9) en el tiempo $n + \frac{1}{2}$ y la ecuación (10) en el tiempo n . Por otra parte, la relación entre los campos en la celda de Yee para el caso 2D será como se ve en la Figura 1. Como se puede apreciar la simulación se relaciona para un punto con coordenadas (i, j) en la grilla de simulación. En este caso existirán dos subgrillas solapadas en las cuales se referencian los campos eléctrico y magnético. En la propuesta de Yee (1996), el valor de uno de los campos está relacionado con el rotacional del otro a su alrededor. En el caso presentado en la figura se presenta la simulación del rotacional del campo eléctrico alrededor del punto donde se evalúa el campo magnético, de esta manera los valores del campo eléctrico quedan localizados en los puntos medios de los lados de una de las sub-grillas, mientras que el campo magnético se localiza en los extremos de la otra.

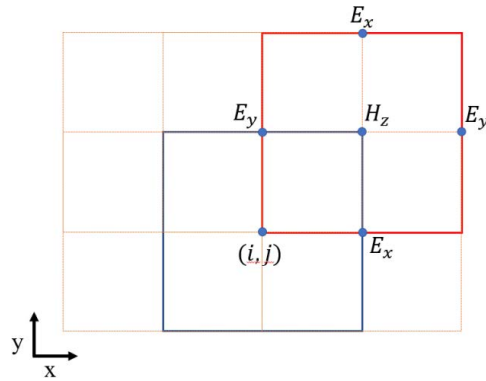


Figura 1

Relación de los campos en la celda de Yee para el caso 2D, considerando la polarización TE.

Fuente: Elaboración propia

Nota: cómo se puede apreciar son dos mallas solapadas, en las cuales los puntos señalados presentan la localización de los campos con respecto a los cuales se realiza la discretización.

Para la ecuación (9)

$$\begin{aligned} & \frac{-1}{c} \frac{B_z^{n+1}\left(i + \frac{1}{2}, j + \frac{1}{2}\right) - B_z^n\left(i + \frac{1}{2}, j + \frac{1}{2}\right)}{\Delta t} \\ &= \frac{E_y^{n+\frac{1}{2}}\left(i + 1, j + \frac{1}{2}\right) - E_y^{n+\frac{1}{2}}\left(i, j + \frac{1}{2}\right)}{\Delta x} \\ & \quad - \frac{E_x^{n+\frac{1}{2}}\left(i + \frac{1}{2}, j + 1\right) - E_x^{n+\frac{1}{2}}\left(i + \frac{1}{2}, j\right)}{\Delta y} \end{aligned} \quad (11)$$

Para la ecuación (10) se obtendrán las ecuaciones

$$\frac{1}{c} \frac{D_x^{n+\frac{1}{2}}\left(i + \frac{1}{2}, j\right) - D_x^{n-\frac{1}{2}}\left(i + \frac{1}{2}, j\right)}{\Delta t} = \frac{H_z^n\left(i + \frac{1}{2}, j + \frac{1}{2}\right) - H_z^n\left(i + \frac{1}{2}, j - \frac{1}{2}\right)}{\Delta y} \quad (12)$$

$$-\frac{1}{c} \frac{D_y^{n+\frac{1}{2}}\left(i, j + \frac{1}{2}\right) - D_y^{n-\frac{1}{2}}\left(i, j + \frac{1}{2}\right)}{\Delta t} = \frac{H_z^n\left(i + \frac{1}{2}, j + \frac{1}{2}\right) - H_z^n\left(i - \frac{1}{2}, j + \frac{1}{2}\right)}{\Delta x} \quad (13)$$

En cuanto a la implementación del algoritmo primero se actualizan los campos $\vec{D}$ y $\vec{E}$, y luego se actualizan los campos $\vec{B}$ y $\vec{H}$.

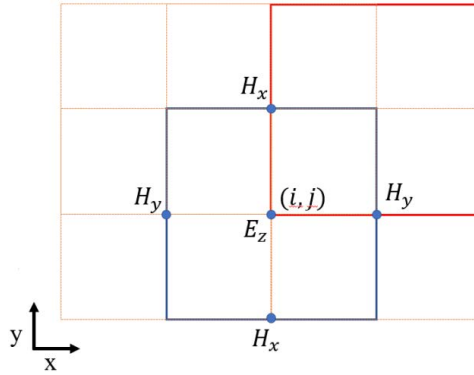


Figura 2

Relación de los campos en la celda de Yee para el caso 2D, considerando la polarización TM.
Fuente: Elaboración propia

De forma análoga para los campos en el modo tm, la relación dada por Yee en su propuesta de discretización se muestra en la Figura 2. Con esta propuesta se tendrá que

$$\vec{H} = (H_x, H_y, 0), \quad \vec{E} = (0, 0, E_z) \quad (14)$$

Para esta polarización las ecuaciones (6) toman la forma

$$\nabla \times \vec{E} = \frac{-1}{c} \frac{\partial \vec{B}}{\partial t} \rightarrow \frac{\partial E_z}{\partial y} = \frac{-1}{c} \frac{\partial B_x}{\partial t}, \quad \frac{\partial E_z}{\partial x} = \frac{1}{c} \frac{\partial B_y}{\partial t} \quad (15)$$

$$\nabla \times \vec{H} = \frac{1}{c} \frac{\partial \vec{D}}{\partial t} \rightarrow \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} = \frac{1}{c} \frac{\partial D_z}{\partial t} \quad (16)$$

En su descripción fdtd las ecuaciones (15) serán

$$\frac{-1}{c} \frac{B_x^{n+1}\left(i, j + \frac{1}{2}\right) - B_x^n\left(i, j + \frac{1}{2}\right)}{\Delta t} = \frac{E_z^{n+\frac{1}{2}}(i, j+1) - E_z^{n+\frac{1}{2}}(i, j)}{\Delta y} \quad (17)$$

$$\frac{1}{c} \frac{B_y^{n+1}\left(i + \frac{1}{2}, j\right) - B_y^n\left(i + \frac{1}{2}, j\right)}{\Delta t} = \frac{E_z^{n+\frac{1}{2}}(i+1, j) - E_z^{n+\frac{1}{2}}(i, j)}{\Delta x} \quad (18)$$

En tanto que la ecuación (16) se podrá escribir como

$$\frac{1}{c} \frac{D_z^{n+\frac{1}{2}}(i, j) - D_z^{n-\frac{1}{2}}(i, j)}{\Delta t} = \frac{H_y^n\left(i + \frac{1}{2}, j\right) - H_y^n\left(i - \frac{1}{2}, j\right)}{\Delta x} - \frac{H_x^n\left(i, j + \frac{1}{2}\right) - H_x^n\left(i, j - \frac{1}{2}\right)}{\Delta y} \quad (19)$$

Nota: los valores de los campos en su evolución temporal se toman en la grilla con índice (i, j) de la celda de Yee, de ahí que se relacionen los campos alrededor del punto con índices (i, j) de la manera sugerida por Yee (1996). En cuanto a la variación temporal, se escogen los campos de la grilla y se relacionan temporalmente asignando a los campos diferentes tiempos que permitan satisfacer la relación (4).

Para que el método sea estable es necesario imponer un criterio para la dependencia temporal, en este caso el paso en el tiempo debe satisfacer la condición de Courant-Friedrichs-Levy (CFL) (Sullivan, 2000). En el sistema 2D que se presenta aquí, se puede escribir como

$$\Delta t \leq \Delta t_m = \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)^{-\frac{1}{2}} \quad (20)$$

Este criterio puede interpretarse como el tiempo mínimo permitido para que la información de un punto de la grilla de simulación viaje a otro punto de la grilla, de tal forma que respete el postulado de máxima velocidad de propagación permitida por la relatividad especial.

Otra condición necesaria para la evolución de los campos en el tiempo es una fuente que brinde energía al sistema, pues al inicio los campos son nulos, es decir, se supone que sus valores en el sistema estudiado son cero. En el caso de un medio dispersivo, es posible escoger un pulso Gaussiano como fuente de excitación que toma la forma

$$f_u(t) = A_m \exp \exp (i * 2\pi f t) \left(-\frac{t - t_0}{\tau} \right)^2 \quad (21)$$

Aunque en principio cualquier forma de excitar el sistema es permitida, lo que asegura el pulso Gaussiano es una manera de excitar modos de propagación permitidos en el sistema alrededor de una frecuencia central f .

Implementación del método fdtd al caso no dispersivo

Para esto, se implementará como guía el procedimiento descrito en el artículo de Ashutosh y Pradip (2012). En primer lugar, hay que definir el tipo de material y la geometría de su sección transversal, en este caso la forma del cristal puede observarse en la Figura 3. Este sistema consiste en un material en bloque con permitividad dieléctrica ϵ_1 con inserciones de barras paralelas de un material con permitividad dieléctrica ϵ_2 , de sección transversal circular de radio r , dispuestas en un arreglo rectangular 2D, de tal manera que los ejes principales de las barras se encuentran orientados en la dirección z .

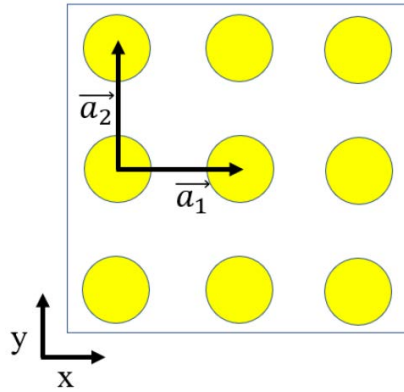


Figura 3
Descripción de la distribución de los materiales en el cristal fotónico estudiado.
Fuente: Elaboración propia

Nota: las barras tienen una permitividad dieléctrica 2 en tanto que el material que las rodea es de permitividad dieléctrica ϵ_1 . $\vec{a}_1$ y $\vec{a}_2$ son los vectores de traslación definidos para el cristal.

Para la simulación FDTD hay que definir una grilla de puntos en los cuales se calcularán los valores de los campos y su evolución temporal. Como se muestra en la Figura 4, se puede definir una grilla para la celda unitaria que define el cristal de tal forma que tenga n_{gx} particiones a lo largo de la dirección x y n_{gy} particiones a lo largo de la dirección y .

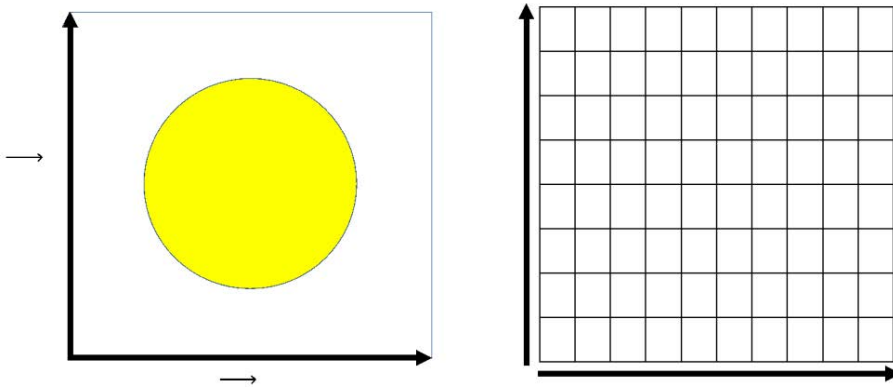


Figura 4
Celda unitaria y grilla de simulación.
Fuente: Elaboración propia

Notas. A la izquierda la celda unitaria definida para el cristal fotónico, a la derecha grilla de simulación para el algoritmo fdttd, como puede verse esta grilla tiene n_{gx} particiones a lo largo de la dirección x y n_{gy} particiones a lo largo de la dirección y .

Con estas consideraciones se puede definir el paso en las direcciones x e y en términos de la longitud de los vectores de traslación de la red así:

$$\Delta x = \frac{|\vec{a}_1|}{n_{gx}}, \quad \Delta y = \frac{|\vec{a}_2|}{n_{gy}} \quad (22)$$

Ahora, para poder implementar esta grilla de simulación en el algoritmo FDTD, hay que definir la función material de permitividad dieléctrica para otorgar a los diferentes puntos de simulación las propiedades de los materiales considerados. En este caso, si el centro de las barras coincide con el centro de la celda unitaria se puede decir que para la función dieléctrica

$$\epsilon(\vec{r}) = \{\epsilon_1, \left[\left(x - \frac{a_1}{2} \right)^2 + \left(y - \frac{a_2}{2} \right)^2 \right] > r^2 \epsilon_2, \left[\left(x - \frac{a_1}{2} \right)^2 + \left(y - \frac{a_2}{2} \right)^2 \right] \leq r^2 \} \quad (23)$$

donde $a_1 = |\vec{a}_1|$ y $a_2 = |\vec{a}_2|$.

En términos de los índices de puntos de la grilla i, j la función dieléctrica se puede definir como

$$\epsilon(i, j) = \{\epsilon_1, \left[\left(i\Delta x - \frac{n_{gx}\Delta x}{2} \right)^2 + \left(j\Delta y - \frac{n_{gy}\Delta y}{2} \right)^2 \right] > f_0 \Delta x \Delta y n_{gx} n_{gy} \epsilon_2, \left[\left(i\Delta x - \frac{n_{gx}\Delta x}{2} \right)^2 + \left(j\Delta y - \frac{n_{gy}\Delta y}{2} \right)^2 \right] \leq f_0 \Delta x \Delta y n_{gx} n_{gy} \epsilon_2 \} \quad (24)$$

donde al número f_0 se lo conoce como el factor de ocupación de las barras en la celda unitaria, el cual es definido como

$$f_0 = \frac{\pi r}{a_1 a_2} \quad (25)$$

donde $a_1 = n_{gx} \Delta x$ y $a_2 = n_{gy} \Delta y$.

Forma de los algoritmos y su implementación en arreglos computacionales

Una de las principales dificultades en la construcción de un algoritmo basado en el método FDTD es el uso de memoria, es decir, cómo representar la información que se va a propagar en el sistema de interés. En este caso se construirán diferentes arreglos en los cuales se dispondrá la información tanto de la variación temporal como espacial, para luego realizar el análisis espectral que permitirá calcular los modos propagantes.

Para esto en la implementación una propuesta para conocer las evoluciones temporales y espaciales para los campos es crear arreglos de la forma $A[n, i, j]$, donde n hace referencia al paso del tiempo, i a la posición en la dirección x , j a la posición en la dirección y . Así en el caso del modo TE las ecuaciones (11), (12) y (13) se pueden escribir como

$$\frac{1}{c} \frac{D_x[n, i, j] - D_x[n-1, i, j]}{\Delta t} = \frac{H_z[n-1, i, j] - H_z[n-1, i, j-1]}{\Delta y} \quad (26)$$

$$\frac{-1}{c} \frac{D_y[n, i, j] - D_y[n - 1, i, j]}{\Delta t} = \frac{H_z[n - 1, i, j] - H_z[n - 1, i - 1, j]}{\Delta x} \quad (27)$$

$$\begin{aligned} \frac{-1}{c} \frac{B_z[n, i, j] - B_z[n - 1, i, j]}{\Delta t} \\ = \frac{E_y[n, i + 1, j] - E_y[n, i, j]}{\Delta x} - \frac{E_x[n, i, j + 1] - E_x[n, i, j]}{\Delta y} \end{aligned} \quad (28)$$

En el caso presentado la evolución temporal de los campos necesita que las ecuaciones anteriores sean secuenciales, es decir, se deben ejecutar en el orden planteado. Esto asegura que los campos estén bien relacionados espacialmente teniendo en cuenta las actualizaciones temporales escogidas al principio.

En cuanto a la conversión de los campos por medio de las ecuaciones materiales (7) estará dada por

$$\vec{D}[n, i, j] = \epsilon[i, j] \vec{E}[n, i, j], \quad \vec{B}[n, i, j] = \mu[i, j] \vec{H}[n, i, j], \quad (29)$$

En el caso de no considerar respuesta magnética se tendrá que $\vec{B} = \vec{H}$.

De forma análoga, se tiene que para el modo tm las ecuaciones (17), (18) y (19) tomarán la forma

$$\begin{aligned} \frac{1}{c} \frac{D_z[n, i, j] - D_z[n - 1, i, j]}{\Delta t} \\ = \frac{H_y[n - 1, i, j] - H_y[n - 1, i - 1, j]}{\Delta x} \\ - \frac{H_x[n - 1, i, j] - H_x[n - 1, i, j - 1]}{\Delta y} \end{aligned} \quad (30)$$

$$\frac{-1}{c} \frac{B_x[n, i, j] - B_x[n - 1, i, j]}{\Delta t} = \frac{E_z[n, i, j + 1] - E_z[n, i, j]}{\Delta y} \quad (31)$$

$$\frac{1}{c} \frac{B_y[n, i, j] - B_y[n - 1, i, j]}{\Delta t} = \frac{E_z[n, i + 1, j] - E_z[n, i, j]}{\Delta x} \quad (32)$$

Resuelto el problema de cómo se realiza la evolución temporal y espacial, otro problema que se presenta es el de las condiciones de frontera, estas deben aplicarse en la frontera de la grilla de simulación. En el caso elegido, al ser un cristal fotónico 2D estas condiciones estarán definidas a través del teorema de Bloch, ecuaciones (1) y (2), por lo cual el conocimiento del comportamiento de los campos puede lograrse solo con una pequeña región del cristal fotónico, que en este caso corresponderá a la celda unitaria que lo define (Kittel, 1966).

Como la implementación de estos algoritmos originalmente se desarrolló en Fortran, se hará la discusión de los problemas enfrentados en este lenguaje. En primer lugar, los arreglos considerados son finitos, por lo tanto los bordes de la grilla en su representación computacional serán los puntos $(1,1)$ y $(n_{gx}+1, n_{gy}+1)$ como se muestra en la Figura 5.

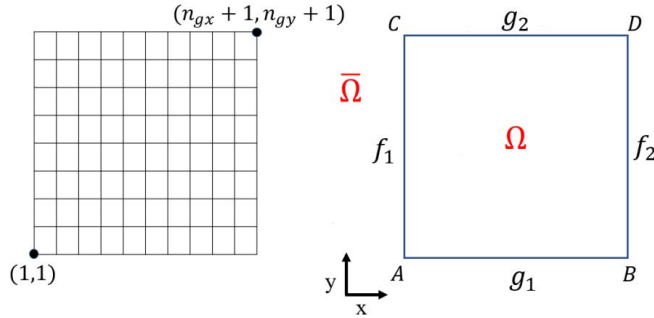


Figura 5
Representación de los puntos de la grilla y descripción de las fronteras.
Fuente: Elaboración propia.

Nota: izquierda representación de los puntos de la grilla en los arreglos para la evolución temporal de los campos. Derecha, descripción de las fronteras sobre las cuales se aplican las condiciones de frontera de Bloch del cristal fotónico. Las regiones interior y exterior de la grilla se designan por Ω y $\bar{\Omega}$ respectivamente.

Es importante resaltar que el aspecto computacional y el aspecto de distribución espacial de los campos son diferentes, el primero es discreto y el otro una representación de un mosaico. La aplicación de las condiciones de frontera de Bloch se puede entender como un conjunto de relaciones de recurrencia para algunos valores de los campos. En el caso mostrado en la Figura 5, los puntos de la frontera son aquellos que se encuentran sobre f_1, f_2, g_1 y g_2 , estos puntos estarán descritos sobre la grilla como:

Sobre $f_1 \rightarrow (1, j), j \in [1, n_{gy} + 1]$

Sobre $f_2 \rightarrow (n_{gx} + 1, j), j \in [1, n_{gy} + 1]$

Sobre $g_1 \rightarrow (i, 1), i \in [1, n_{gx} + 1]$

Sobre $g_2 \rightarrow (i, n_{gy} + 1), i \in [1, n_{gx} + 1]$

Cabe resaltar que en este esquema los puntos A, B, C y D son contados dos veces.

Ahora, de acuerdo con las condiciones de Bloch en el cristal fotónico, ecuación (2)

$$\vec{A}(\vec{r} + \vec{a}) = \exp(i\vec{k} \cdot \vec{a})\vec{A}(\vec{r})$$

Donde $\vec{r}$ es un vector posición en la celda unitaria y $\vec{a}$ es un vector de translación de la forma

$$\vec{a} = m\vec{a}_1 + n\vec{a}_2 \quad (33)$$

A partir de esta ecuación se definen dos nuevos vectores

$$\vec{R}_x \rightarrow \text{para } m = 1 \text{ y } n = 0,$$

$$\vec{R}_y \rightarrow \text{para } m = 0 \text{ y } n = 1,$$

Que serán utilizados para definir la dependencia de los campos en las fronteras. Mientras, hay que notar que para los puntos en la región Ω de la Figura 5, los campos y sus dependencias funcionales con los puntos alrededor están bien definidos. No obstante, para los puntos en que se encuentran alrededor de la frontera f_1 , los campos estarán definidos por

$$\vec{A}(\vec{r} - \vec{R}_x) = \exp(-i\vec{k} \cdot \vec{R}_x) \vec{A}(\vec{r}) \quad (34)$$

Y así para las demás fronteras de la grilla. Esto hace que los campos en Ω estén definidos con respecto de los campos en Ω por medio de las relaciones de Bloch, así

$$\vec{A}(\vec{r} - \vec{R}_x) = \exp \exp(-i\vec{k} \cdot \vec{R}_x) \vec{A}(\vec{r}), \quad \text{para } \vec{r} \in f_1$$

$$\vec{A}(\vec{r} - \vec{R}_y) = \exp \exp(-i\vec{k} \cdot \vec{R}_y) \vec{A}(\vec{r}), \quad \text{para } \vec{r} \in g_1$$

$$\vec{A}(\vec{r} + \vec{R}_x) = \exp \exp(i\vec{k} \cdot \vec{R}_x) \vec{A}(\vec{r}), \quad \text{para } \vec{r} \in f_2$$

$$\vec{A}(\vec{r} + \vec{R}_y) = \exp \exp(i\vec{k} \cdot \vec{R}_y) \vec{A}(\vec{r}), \quad \text{para } \vec{r} \in g_2$$

Claro está que las relaciones de recurrencia se aplican sobre los puntos que así lo necesiten, pues la evolución de los campos ocurre en todos los puntos del sistema a simular.

Hay que analizar la aplicación de estas relaciones de frontera a las ecuaciones computacionales que rigen la evolución de los campos para el modo TE, ecuaciones (26), (27) y (28).

La ecuación (26) puede escribirse para la región Ω y la frontera g_1 en forma de algoritmo como

```

Do i = 1, ngx + 1
  Do j = 2, ngy + 1
    Dx[n, i, j] = Dx[n - 1, i, j] +  $\frac{c\Delta t}{\Delta y}$  (Hz[n - 1, i, j] - Hz[n - 1, i, j - 1])
  End do
  Dx[n, i, 1] = Dx[n - 1, i, 1] +  $\frac{c\Delta t}{\Delta y}$  (Hz[n - 1, i, 1] - exp(-i $\vec{k} \cdot \vec{R}_y$ ) Hz[n - 1, i, ngy + 1])
End do

```

La justificación para los campos en la frontera g_1 es la siguiente. Como se sabe la relación de Bloch para los puntos en g_1 es

$$\vec{A}(\vec{r} - \vec{R}_y) = \exp \exp(-i\vec{k} \cdot \vec{R}_y) \vec{A}(\vec{r}),$$

En términos de las posiciones (i, j) se tiene que para $j = 1$

$$H_z[n - 1, i, j - 1] \rightarrow H_z[n - 1, i, 0]$$

el cual no está definido en el arreglo computacional, a cambio de esto se usa la condición de Bloch como

$$\vec{A}[i, j - (n_{gy} + 1)] = \exp \exp(-i\vec{k} \cdot \vec{R}_y) \vec{A}[i, j]$$

de donde se tiene que

$$\vec{A}[i, (n_{gy} + 1) - (n_{gy} + 1)] = \vec{A}[i, 0] = \exp \exp(-i\vec{k} \cdot \vec{R}_y) \vec{A}[i, n_{gy} + 1]$$

Que es el lugar buscado. En esta justificación no se toma en consideración el componente temporal pues los campos están evaluados en el mismo tiempo.

De forma análoga se tiene que para la ecuación (27) y la frontera f_1

<pre> Do j=1, n_{gy} + 1 Do i=2, n_{gx} + 1 D_y[n, i, j] = D_y[n - 1, i, j] - $\frac{c\Delta t}{\Delta x} (H_z[n - 1, i, j] - H_z[n - 1, i - 1, j])$ End do D_y[n, 1, j] = D_y[n - 1, 1, j] - $\frac{c\Delta t}{\Delta x} (H_z[n - 1, 1, j] - \exp \exp(-i\vec{k} \cdot \vec{R}_x) H_z[n - 1, n_{gx} + 1, j])$ End do </pre>

Ahora, para la ecuación (28) y las fronteras f_1 g_2

<pre> Do i=1, n_{gx} Do j=1, n_{gy} B_z[n, i, j] = B_z[n - 1, i, j] - cΔt $\left(\frac{E_y[n-1, i+1, j] - E_y[n, i, j]}{\Delta x} - \frac{E_x[n-1, i, j+1] - E_x[n, i, j]}{\Delta y} \right)$ End do B_z[n, i, n_{gy} + 1] = B_z[n - 1, i, n_{gy} + 1] - cΔt $\left(\frac{E_y[n-1, i+1, n_{gy}+1] - E_y[n, i, n_{gy}+1]}{\Delta x} - \frac{\exp(i\vec{k} \cdot \vec{R}_y) E_x[n-1, i, 1] - E_x[n, i, n_{gy}+1]}{\Delta y} \right)$ </pre>
--

End do

Do j=1, n_{gy}

$$B_z[n, n_{gx} + 1, j] = B_z[n - 1, n_{gx} + 1, j] - c\Delta t \left(\frac{\exp(i\vec{k} \cdot \vec{R}_x) E_y[n-1, 1, j] - E_y[n, n_{gx} + 1, j]}{\Delta x} - \frac{E_x[n-1, n_{gx} + 1, j+1] - E_x[n, n_{gx} + 1, j]}{\Delta y} \right)$$

End do

$$\begin{aligned} B_z[n, n_{gx} + 1, n_{gy} + 1] \\ &= B_z[n - 1, n_{gx} + 1, n_{gy} + 1] \\ &- c\Delta t \left(\frac{\exp(i\vec{k} \cdot \vec{R}_x) E_y[n - 1, 1, n_{gy} + 1] - E_y[n, n_{gx} + 1, n_{gy} + 1]}{\Delta x} \right. \\ &- \left. \frac{\exp(i\vec{k} \cdot \vec{R}_y) E_x[n - 1, n_{gx} + 1, 1] - E_x[n, n_{gx} + 1, n_{gy} + 1]}{\Delta y} \right) \end{aligned}$$

En esta parte del código la última relación es la correspondiente con el punto $(n_{gx} + 1, n_{gy} + 1)$ que no se ha incluido anteriormente.

Así como se tiene que las ecuaciones computacionales para los modos te es posible tener ecuaciones similares para los modos tm, ecuaciones (30), (31) y (32) así

Do i = 2, $n_{gx} + 1$

Do j = 2, $n_{gy} + 1$

$$D_z[n, i, j] = D_z[n - 1, i, j] + c\Delta t \left(\frac{H_y[n-1, i, j] - H_y[n-1, i-1, j]}{\Delta x} - \frac{H_x[n-1, i, j] - H_y[n-1, i, j-1]}{\Delta y} \right)$$

End do

$$\begin{aligned} D_z[n, i, 1] &= D_z[n - 1, i, 1] + c\Delta t \left(\frac{H_y[n-1, i, 1] - H_y[n-1, i-1, 1]}{\Delta x} - \right. \\ &\left. \frac{H_x[n-1, i, 1] - \exp(-i\vec{k} \cdot \vec{R}_y) H_y[n-1, i, n_{gy} + 1]}{\Delta y} \right) \end{aligned}$$

End do

Do j = 2, $n_{gy} + 1$

$$D_z[n, 1, j] = D_z[n - 1, 1, j] + c\Delta t \left(\frac{H_y[n-1, 1, j] - \exp(-i\vec{k} \cdot \vec{R}_x) H_y[n-1, n_{gx} + 1, j]}{\Delta x} - \frac{H_x[n-1, 1, j] - H_y[n-1, 1, j-1]}{\Delta y} \right)$$

End do

$$\begin{aligned} D_z[n, 1, 1] &= D_z[n - 1, 1, 1] \\ &+ c\Delta t \left(\frac{H_y[n - 1, 1, 1] - \exp(-i\vec{k} \cdot \vec{R}_x) H_y[n - 1, n_{gx} + 1, 1]}{\Delta x} \right. \\ &- \left. \frac{H_x[n - 1, 1, 1] - \exp(-i\vec{k} \cdot \vec{R}_y) H_y[n - 1, 1, n_{gy} + 1]}{\Delta y} \right) \end{aligned}$$

```

Do i=1, ngx + 1
  Do j=1, ngy
     $B_x[n, i, j] = B_x[n - 1, i, j] - \frac{c\Delta t}{\Delta y} (E_z[n, i, j + 1] - E_z[n, i, j])$ 
  End do
   $B_x[n, i, n_{gy} + 1] = B_x[n - 1, i, n_{gx} + 1] - \frac{c\Delta t}{\Delta y} (\exp(i\vec{k} \cdot \vec{R}_y) E_z[n, i, 1] - E_z[n, i, n_{gy} + 1])$ 
End do

Do j=1, ngy + 1
  Do i=1, ngx
     $B_y[n, i, j] = B_y[n - 1, i, j] + \frac{c\Delta t}{\Delta x} (E_z[n, i + 1, j] - E_z[n, i, j])$ 
  End do
   $B_y[n, n_{gx} + 1, j] = B_y[n - 1, n_{gx} + 1, j] + \frac{c\Delta t}{\Delta x} (\exp(i\vec{k} \cdot \vec{R}_x) E_z[n, 1, j] - E_z[n, n_{gx} + 1, j])$ 
End do

```

No está de más mencionar que al emplear los campos $\vec{E}, \vec{D}, \vec{B}$ y $\vec{H}$ en todo momento se necesita realizar la equivalencia de un campo a otro por medio de las relaciones materiales presentadas en la ecuación (29).

Para resumir

- Hasta ahora lo realizado es una pequeña introducción al método FDTD. En las páginas anteriores se presentaron las ecuaciones que permiten calcular cómo se comportan los campos electromagnéticos en un cristal fotónico formado por barras paralelas de sección transversal circular formadas por un material dieléctrico no dispersivo sin respuesta magnética, dispuestas en un arreglo rectangular.
- Se describió la construcción de la función permitividad eléctrica del sistema, así como las relaciones de recurrencia para los campos electromagnéticos considerando las polarizaciones TE y TM.
- También, se describió la aplicación de las condiciones de frontera de Bloch, así como, su implementación algorítmica, específicamente al tomar en cuenta el lenguaje de programación Fortran. Se muestra la relación de estabilidad numérica basada en el criterio de Courant-Friedrichs-Lewy.

CAPÍTULO 2: Análisis espectral

Resumen del capítulo

Para realizar el análisis de los modos de propagación permitidos en un cristal fotónico se debe contar con un análisis espectral de los modos permitidos en el sistema. En este caso la respuesta de la evolución temporal de los algoritmos basados en fdtd deben convertirse a su representación en frecuencias, lo cual puede lograrse por medio de la transformada de Fourier. En este capítulo se mostrará cómo se puede discretizar la transformada de Fourier para lograr el espectro de frecuencias. Una vez conocido el espectro de frecuencias, se escogerán aquellos valores de frecuencia en los cuales se presentan picos, como los modos resonantes del sistema cristal fotónico, los cuales son calculados para un valor del vector de onda de la radiación incidente.

Transformada Discreta de Fourier (Discret Fourier Transformation – DFT)

Considérese la transformada de Fourier

$$F(\omega) = \int_0^T g(t) \exp(-i\omega t) dt \quad (35)$$

Donde $g(t)$ es una función periódica con periodo T , $g(t + T) = g(t)$.

Al escribir de forma discreta se tiene que si N es el número de intervalos entre los cuales está dividido el periodo, esto es $T = N\Delta t$, siendo Δt el espacio temporal, entonces se puede escribir una frecuencia fundamental o mínima

$$\Delta f = \frac{1}{T} = \frac{1}{N\Delta t} \quad (36)$$

Con esta definición la forma discreta de la transformación (35) será

$$F(k) = \sum_{n=0}^N g(n) \exp(-i * 2\pi * k * \frac{n}{N}) \Delta t = 0 \quad (37)$$

Donde el valor de k se toma para cada frecuencia estudiada en unidades de Δf , por otra parte, la función $g(t) \rightarrow g(n)$ donde n corresponde al valor de la función en el tiempo $g(n\Delta t)$ solo que por simplicidad se escribe el tiempo t en unidades de Δt

Esta última función es la utilizada para realizar el análisis de la evolución temporal de los campos.

Algunos puntos importantes en la implementación de este algoritmo son:

- La función se supone periódica con periodo T , esto implica que en la implementación del algoritmo FDTD se debe esperar a que la señal se estabilice y escoger una parte de toda la respuesta temporal para realizar el análisis.
- En este análisis no se tienen en cuenta frecuencias por debajo del valor de Δf
- La señal se analiza después de que se estabilice; sin embargo, la elección del momento en el cual se alcanza la estabilidad es definido por quien realiza el análisis de la señal.
- Como ya se mencionó, los datos obtenidos están limitados en el muestreo por el tamaño de las matrices usadas en el programa. No obstante, esto puede resolverse si se realiza directamente la transformación mientras los campos evolucionan o puede exportarse la señal de interés a un archivo para un análisis posterior.

La ecuación (37) equivale a la implementación algorítmica de la DFT. Sin embargo, en el algoritmo FDTD aquí presentado el primer valor del tiempo es el lugar $n=1$ en el arreglo. Este lugar corresponde a un tiempo $t=0$, así la interpretación del DFT procedimentalmente será

$$F(k) = \sum_{n=0}^N g(n) \exp \exp \left(-i * 2\pi * k * \frac{n-1}{N} \right) \Delta t = 0 \quad (38)$$

La forma del algoritmo para el procesamiento de la señal será

```

Do fre=1, maxfre
  Suma=0
  Do n=1, maxtime
    Suma = suma + A[n,i,j] * exp(-i * 2π * k * (n-1)/maxtime) * Δt
  End do
  Write(:,*), fre, real(suma*conjg(suma))
End do

```

En esta parte del archivo se tiene que:

- La multiplicación por Δt es opcional, pues solo importarán los picos de resonancia.
- La información que se procesa solo se realiza para ciertos puntos de interés, en este caso la información procesada es sobre uno de los campos en el lugar (i, j) de la grilla.
- Se guarda un archivo para la respuesta con cada valor de frecuencia en la cual se realiza el análisis.
- Como ya se mencionó se puede guardar la información para muchos lugares en la grilla, para incrementar la localización de diversos modos de resonancia del sistema.
- Hay que recordar que solo los modos propios del sistema sobreviven, lo que equivale a un pico de resonancia en el gráfico espectral.
- Se puede optimizar el análisis utilizando la Transformada Rápida de Fourier (*Fast Fourier Transformation* – FFT) pero ese algoritmo no se implementará en esta guía introductoria.

Análisis espectral sin considerar un arreglo para la parte temporal

Se crea este algoritmo para realizar el análisis de resultados del método FDTD pero sin tener un arreglo temporal, en este caso el análisis de los datos se realiza directamente en el algoritmo a medida que se van generando las variaciones temporales.

- El análisis se realiza con el método DFT.
- En este algoritmo se usarán diversos puntos para hallar la transformada de Fourier de la señal, para hallar los picos de resonancia se sumarán diversos puntos en la celda, el propósito es luego tomar estos y realizar un gráfico espectral.

Para realizar la escogencia de los picos se hace necesario un algoritmo que permita la comparación de los valores de los puntos en intensidad. Entonces, se procederá como sigue: sea $f(x)$ una función discreta que presenta una forma como en la figura 6.

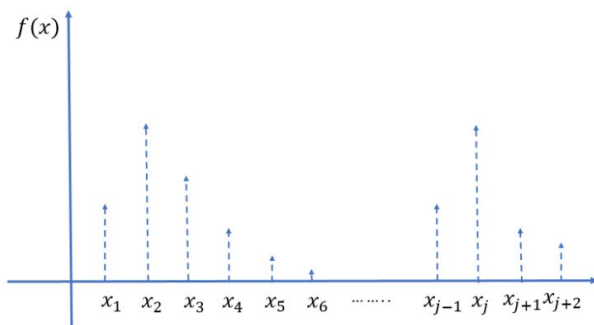


Figura 6

Esquema del diagrama espectral sobre el cual se escogen los máximos entre valores vecinos.
Fuente: Elaboración propia

Entonces los máximos locales se pueden hallar por medio de la comparación de estos en grupos de tres, por medio de una condición que definirá el máximo local del grupo. Supongase que un máximo se encuentra en la posición j , entonces se tendrá un máximo local si se cumple que

$$f(x_j) > f(x_{j-1}), f(x_{j+1}) \quad (39)$$

Esta condición de forma algorítmica puede expresarse como

```

Do j=2, N-1
  If ( $f(x_j) > f(x_{j-1})$  and  $f(x_j) > f(x_{j+1})$ ) then
    Write(:,*)j
  End if
End do
  
```

En este cálculo no se incluyen los puntos inicial y final, pues las condiciones impuestas sobre ellos pueden dar lugar a máximos locales falsos en la escritura del archivo.

Con respecto de la transformada de Fourier se tiene que al no usar un arreglo que tenga la parte temporal se puede tomar como tiempo inicial $n = 0$, por lo tanto, se modifica esta parte del algoritmo de $n - 1$ a n .

Por otra parte, para mejorar el rendimiento del algoritmo se puede crear un arreglo diferente que guarde la información de uno de los lugares o mejor aún, es posible utilizar la escritura de archivos y luego leerlos cuando sea necesario. En la implementación al sustituir los valores de los (n, i, j) por solo (i, j) y hacer las variaciones temporales término a término usando la recurrencia del método fdtd resulta ser un poco más demorado que las asignaciones de memoria (n, i, j) para cada tiempo.

En los programas mostrados en el anexo para el caso de la polarización TM se escriben dos archivos:

- fdtdTMvsT.dat – que contiene información de la variación temporal
- fdtdTMvsW.dat – que contiene la transformada de Fourier del primer archivo.

Esto se realiza con el propósito de un mejor uso de memoria, pues al ejecutar se observa que no es práctico el uso de arreglos para valores temporales por encima de 11 000 pasos.

Anotaciones sobre el uso de archivos en Fortran

Es posible guardar la información para después utilizarla; sin embargo, hay que recordar algunas reglas de uso de los archivos en Fortran, por ejemplo:

- Se debe abrir el archivo con los datos importantes o relevantes del problema.
- Cuando se lee un archivo es importante indicar al puntero de lectura de archivos que vuelva a la primera línea del archivo, pues en caso contrario ocurre un error de lectura. Esto pasa cuando el archivo guardado debe leerse varias veces, así en un ciclo de lectura del archivo es importante añadir la orden rewind(unidad del archivo) para posicionarlo de nuevo al inicio del mismo y continuar con una nueva lectura de los datos guardados.
- Un archivo usado para lectura no será cambiado, es mejor crear otro y solicitar luego con la orden Close que se destruya el archivo y se remplace.
- Un archivo que contenga los datos así guardados no consumirá o hará uso de memoria, pues los datos se acceden al momento de usarlos.
- La orden read(unidad del archivo,*) línea de datos, asocia los datos del archivo con las variables de interés del problema elegido. Se espera que el número de columnas en el archivo sea igual al número de variables que se usan en el análisis de los datos.
- Es conveniente crear archivos con los datos de programación o las variables usadas para así tener información de los sistemas estudiados.

Una forma de crear el archivo se muestra a continuación

```

Open (unit=2, file:"fdtdTMvsT.dat",status="old")
Open (unit=3, file:"fdtdTMvsW.dat",status="unknown")
Do fre=0, maxfre
  Suma=0
  Do n=0, maxtime
    Read(2,*) tiempo,campo
    Suma=suma+campo*exp(-im*2*pi*tiempo*fre/maxtime)
  End do
  Rewind(2)
  Print*,fre,real(suma*conjg(suma))
  Write(3,*)fre,real(suma*conjg(suma))
End do
Close(unit=2)
Close(unit=3)

```

Implementación para el cálculo de los modos propios del sistema variando el vector de onda

Para el cálculo de las propiedades del cristal fotónico, el análisis espectral debe realizarse considerando un vector de onda, en este caso el vector de onda indica la dirección de propagación de la radiación electromagnética al interior del cristal. Es conocido que las diferentes direcciones de propagación pueden darse para diferentes valores de frecuencia, de esta forma un diagrama de bandas es la relación entre el vector de onda y la frecuencia para la cual la radiación electromagnética puede propagarse (Joannopoulos *et al.*, 2008; Kittel, 1966). Esto en el análisis espectral se ve como picos de resonancia, los cuales son escogidos para construir la estructura de bandas. En esta guía se implementará el cálculo de los picos de resonancia en una dirección específica para el vector de onda, esta es conocida como dirección $\Gamma - X$, en la cual el vector de onda indica la dirección de propagación de la dirección x en el cristal.

Por otra parte, si se desea conocer los modos propios del sistema hay que considerar que el lugar en el cual se suministra energía también debe evolucionar en el tiempo, de acuerdo con la evolución del método FDTD. Para esto se permitirá que se suministre energía por un tiempo y luego se dejara oscilar libremente el lugar donde se aplicó, dejándolo evolucionar bajo el efecto de los demás puntos que lo rodean. En este caso, hay que hacer que el punto donde se genera el pulso actúe hasta que haya pasado un tiempo PT para el cual

$$f(PT) = \frac{f(t_0)}{10^{10}} \quad (40)$$

Así pues, si $f(n) = \exp(-0.5 * \frac{(t_0-n)^2}{spread})$ (Sullivan, 2000), el valor de PT será

$$PT = t_0 + \sqrt{\frac{spread * \ln(10^{10})}{0.5}} \quad (41)$$

Esta condición establece que se deja evolucionar libremente después que la excitación al sistema tiene un $1/10^{10}$ de su valor original. Es a partir de aquí que el sistema evoluciona con sus modos de propagación permitidos. Así, hay que añadir una condición para el pulso en el núcleo de la evolución fdtd, una propuesta es

```
If (n<= PT) then
  Rutina para agregar implementar el pulso en la
  posición (i,j)
End if
```

Por otra parte, la implementación para el cálculo de la estructura de bandas en la dirección $\Gamma - X$ debe considerar que los valores del vector de onda en esta dirección tienen la forma

$$\vec{k}(r) = (k_1(r), 0) = \left(r * \frac{\pi}{|\vec{a}_1| * N}, 0 \right) \quad (42)$$

Donde N es el número de puntos en los cuales se divide el segmento $\Gamma - X$ y r es el contador de los puntos en el segmento. Por lo tanto, la estructura de un programa para esta dirección tendrá la forma

```
K(2)=0
Do r=0, N
  K(1)=real(r)*pi/(a*real(N))
  Lugar donde se da la
  condición inicial para todos
  los campos.
  Do n=0, maxtime
    Implementación FDTD: en esta parte del programa se implementará las relaciones de
    recurrencia del método FDTD y la modificación para la duración del pulso

    Coordenadas en la grilla del punto a evaluar.

    write(unidad del archivo, *), r,n,campo(i,j)
  End do
End do
```

Ahora, con la información de la variación temporal guardada en un archivo, se puede utilizar este para analizar y realizar el análisis espectral, el cual a su vez se guarda en otro archivo, el cual deberá contener la información:

- maxtime – número de pasos temporales usados en la simulación
- N – número de particiones del segmento gamma-X
- dt – duración del intervalo temporal
- df – unidad del intervalo de frecuencias

los datos que determinan el tamaño del archivo serán: maxtime y N.

Algo de vital importancia para que el algoritmo que realiza la transformada de Fourier y el análisis de los picos de resonancia es poder leer correctamente el archivo con la información generada de la evolución temporal los cuales son: r , n , campo. Este último valor es una variable compleja. Para hacer el análisis espectral utilizando DFT, se necesita un buen manejo del archivo que contiene la información de la evolución temporal, lo cual se implementara de la siguiente manera:

```

Do fre=0, maxfre
  Do r=0, N
    Suma=0
    Do n=0, maxtime
      read(unidad archivo de lectura,*),l,tiempo, campo
      suma=suma+campo*exp(-im*2*pi*tiempo*fre/maxtime)*dt
    end do
    write(unidad archivo de escritura,*),fre,real(suma*conjg(suma))
  End
  Rewind(unidad archivo de lectura)
End do

```

En el archivo que se escribe se encontrará la información del espectro de frecuencias para los diferentes puntos en la dirección $\Gamma - X$ calculados con el algoritmo de evolución temporal.

Para escoger los picos de resonancia se utilizará un arreglo que contenga la información del lugar y de la frecuencia que ha sido calculada, para esto se hará algo de la forma

```

CampovsW(maxfre+1,N+1)
Do fre=0, maxfre
  Do i=0, N
    read(unidad archivo con la información espectral,*),j,r,campovsW(fre+1,i+1)
  end do
end do

```

Así, se guarda en un arreglo la información espectral contenida en el archivo. Esto puede realizarse después y no necesita ejecutarse en el mismo momento.

Para poder realizar el análisis de los picos se usa el arreglo, pues permite comparar rápidamente la información sobre los mismos. Es importante recordar que esto puede ser implementado para cualquier dirección en la celda unitaria.

Ahora, para el análisis de los picos se puede proceder de la siguiente manera

```

Do i =1, N+1
  Do j=2,maxfre

```

```

    Rutina para comparación para hallar los picos de resonancia y escritura de
    archivo con la información de los picos de resonancia (resonanciasTM-g-x.dat)

```

```

  End do
End do

```

La información contenida en el archivo de los picos de resonancia carece de unidades, por lo que solo son lugares en los cuales se ha realizado la comparación, no obstante, estos pueden ser cambiados a las unidades que se deseen por medio de la multiplicación de factores de escala.

El archivo entregado tiene entradas de la forma: lugar donde se calcula el pico, valor del pico resonancia.

En el programa los archivos creados y utilizados son:

- fdtTMvsT-g-x.dat – archivo donde se guarda la evolución temporal
- fdtTMvsW-g-x.dat – archivo donde se guarda el análisis espectral

todo lo anterior también puede ser aplicado al caso del modo TE.

Análisis del espectro de frecuencias comparando los picos de resonancia de los espectros para diversos puntos usando la transformada discreta de Fourier

Para realizar este tipo de análisis se recogerá la información de diversos puntos de la grilla en un archivo que presentará la siguiente estructura: r, n, p, campo(p). donde r es un entero asociado al vector de onda, n es el tiempo de simulación, p un entero asociado al punto analizado y campo(p) el valor del campo eléctrico o magnético en el punto escogido.

En el núcleo de la evolución temporal fdt se escogen los puntos y se guardan de acuerdo con el siguiente esquema

```

Do p=1, np
  Write(unidad del archivo, *) r, n, p, campo(p)
End do

```

Para hacer el análisis para cada punto p escogido se implementará el siguiente algoritmo

```

Do fre=0, maxfre
  Do r=0, nk
    Suma(:)=0 # un arreglo con np lugares
    Do n=0, maxtime
      Do j=1, np
        Read(unidad donde están los datos,*) i, tiempo, p, campo
        Suma(p)=suma(p)+campo*exp(-im*2*pi*tiempo*fre/maxtime)
      End do
    End do
  End do
End do

```

```

    End do
    Do j=1, np
        Write(unidad donde se guardan los datos,*) i, fre, j, real(suma(j)*conjg(suma(j)))
    End do
End do
End do
End do

```

Para calcular las resonancias se hará un cambio al algoritmo que compara los picos de resonancia, esto pues se deben comparar todos los picos para todos los puntos.

Primero se realizará un algoritmo para la lectura de los datos

```

Do fre=0, maxfre
    Do i=0,nk
        Do p=1, np
            Read(unidad donde estan los datos,*) j, r, n, campovsW(fre+1,i+1,p)
        End do
    End do
End do

```

Es importante recordar que se están guardando los datos en cada paso del proceso, esto para luego realizar análisis que puedan ser útiles. En este caso los datos se están guardado en un archivo tipo texto, los valores leídos de los campos se guardan en un arreglo `campovsW(fre+1,i+1,p)`, donde se especifica cada uno de los valores mostrados y corresponden con la transformada inversa de Fourier.

Ahora, para ubicar los picos se realiza un análisis para cada punto así

```

Do p=1, np
    Algoritmo de comparación de los picos en el
    espectro de frecuencias.
End do

```

Solo que se modifica el archivo guardado de la comparación así: `p, i, j`. Donde `p` es el punto, `i` es el vector de onda y `j` el valor de la frecuencia en unidades de `df`.

Para hacer la comparación para cada vector de onda `i`, se deben comparar los diversos puntos `p` y su respectivo valor de frecuencia `j`, esto es si para dos puntos distintos `p1` y `p2` un pico de resonancia es el valor `j`, entonces se guarda dicho valor, y se sigue el mismo proceso para todos los valores y todos los puntos.

La implementación algorítmica se realizará de la siguiente manera

```

Io=0
Do while (io>=0)
  Read(unidad con los datos, *,iostat=io) p, i, j
  If (io==0) then
    Arreglo(i+1,p,j+1)=1
  End if
End do

```

Es importante destacar que este tipo de estructura es para un archivo para el cual no hay la información sobre si está creado o no, esto se logra con el comando iostat.

Aquí la información se guarda en un arreglo que tiene un tamaño $A(nk, np, nf)$ que en un inicio tiene todos sus valores iguales a cero. Luego, si se encontró alguna resonancia para un punto en particular, se cambia dicho valor a 1 para hacer la comparación. La comparación de los picos de resonancia para todos los puntos escogidos en el análisis espectral se puede realizar con un algoritmo como

```

Do i=1, nk+1
  Do j=1, maxfre+1
    Bandera=.true.
    Do p=1, np
      If [A(i,p,j)==0] then
        Bandera=.false.
        p=np+1
      End if
    End do
    If (bandera.equ..true.) then
      Write(unidad donde se guardan las frecuencias para un i dado, *) i-1,j-1
    End do
  End do
End do

```

En esta rutina si hay un cero para un valor i,p,j dado, entonces no es una frecuencia propia de sistema, por lo tanto hay que salir del bucle de comparación.

Implementación FDTD y análisis DFT

De acuerdo a la estructura algorítmica, los archivos creados son usados para realizar el análisis fdt después de simulado, de tal forma que estos datos puedan ser usados en otro tipo de procesos de señales.

Ahora, de acuerdo con otros investigadores del método, se recomienda que el análisis de los datos se realice conforme se van obteniendo los mismos, así
De esta forma, se generan los datos que serán analizados al final. En este caso, los datos

Programa FDTD

Estructura FDTD: actualización de los campos



Análisis DFT para cada frecuencia: esto pues FDTD es un método que varía el tiempo a medida que actualiza los campos.

Escritura de archivo con el análisis

Fin programa

entregados contienen la información de los puntos estudiados, así como el vector de onda y su variación con respecto de la frecuencia, es decir, para cada valor de $\vec{k}$ se analizan p puntos, esta información para cada punto p debe tener la variación del campo con la frecuencia, $\vec{D}(t) \rightarrow \vec{D}(\omega)$, donde su dependencia se verá de la siguiente manera
Los picos mostrados corresponden a máximos locales que pueden ser interpretados como

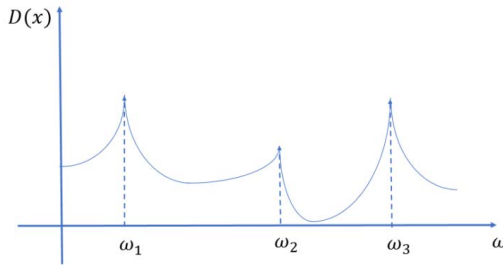


Figura 7

Diagrama espectral de una señal periódica en el espectro de frecuencias.

Fuente: Elaboración propia

modos resonantes o propios del sistema estudiado.

Así, al comparar los picos en el diagrama espectral, se puede encontrar las frecuencias propias del sistema. La implementación del DFT en el núcleo FDTD se puede realizar de la siguiente manera
Al finalizar este cálculo se debe guardar la información en un archivo, esto se realiza de la

Suma(:, :, :) = 0

Do r=0, nk

Do n=0, maxtime

Nucleo
FDTD

```

Do j=1, np
  Do fre=0, maxfre
    Suma(r+1, fre+1, j)=suma(r+1,fre+1,j)+exp(-im*2*pi*fre*n/maxfre)*campo(j)
  End do
End do
# parte del algoritmo en donde se extrae la información de los puntos analizados.

```

```

End do
End do

```

siguiente forma:

Algunos resultados preliminares en la implementación mostraron una reducción significativa

```

Do fre=0, maxfre
  Do i=0, nk
    Do p=1, np
      Write(unidad archivo, *) i, fre, p, real (suma(i+1,fre+1,p)*conjg(suma(i+1,fre+1,p))
    End do
  End do
End do

```

del tiempo de cómputo, tanto así que se pudieron estudiar alrededor de 30 puntos para un maxtime=20000 y nk=15, en menos de 30 minutos.

Sin embargo, subsiste el problema de la escogencia de los puntos para el análisis dado el tipo de fuente. Esto pues los resultados no son independientes de la región en la grilla de simulación estudiada, aunque se deben considerar puntos con baja simetría.

CAPÍTULO 3:

Método FDTD aplicado a sistemas dispersivos

Resumen del capítulo

En este capítulo se desarrolla el formalismo FDTD para el caso en el cual las respuestas materiales son dispersivas, es decir, dependen de la frecuencia de la radiación electromagnética incidente. Se trabaja especialmente el caso de materiales metálicos. Así mismo, se desarrolla el formalismo para el caso de los metamateriales que son conjuntos de circuitos resonantes los cuales presentan una respuesta efectiva para sus funciones dieléctrica y magnética que no se halla en la naturaleza.

Respuesta material dispersiva dieléctrica arbitraria

En este apartado se analizará como aplicar el método fdtD a materiales que cambian sus propiedades ópticas en función de la frecuencia de la radiación incidente (metales, medios polares, metamateriales, entre otros) (Kittel, 1966). Para esto se tomará como referencia al trabajo mostrado en el libro de Sullivan (2000), en el cual se realiza el estudio en el caso de disposiciones materiales 1D, con respuestas dieléctricas dispersivas de la forma

$$\begin{aligned}\epsilon(\omega) &= \epsilon_r + \frac{\sigma}{i\omega\epsilon_0}, \\ \epsilon(\omega) &= \epsilon_r + \frac{\sigma}{i\omega\epsilon_0} + \frac{x}{1 + i\omega t_0}, \\ \epsilon(\omega) &= 1 + \frac{\omega_p^2}{\omega(i\omega_e - \omega)} \\ \epsilon(\omega) &= \epsilon_r + \frac{\epsilon_1}{1 + 2i\delta_o\left(\frac{\omega}{\omega_0}\right) - \left(\frac{\omega}{\omega_0}\right)^2}\end{aligned}\tag{43}$$

Con el uso de:

- La transformada inversa de Fourier.
- Una ecuación diferencial auxiliar.
- La transformada Z.

Todas estas técnicas se basan en la descripción de la respuesta en frecuencia usando la transformada inversa de Fourier. En este caso solo se desarrollará el formalismo para la respuesta dieléctrica, siendo una deducción similar el caso de la respuesta magnética.

Considérese la relación

$$\vec{D}(\vec{r}, \omega) = \epsilon(\vec{r}, \omega)\vec{E}(\vec{r}, \omega)\tag{44}$$

En su descripción temporal se tiene que

$$\vec{D}(\vec{r}, t) = \int_0^t dt' \phi(\vec{r}, t - t') \vec{E}(\vec{r}, t') \quad (45)$$

Donde

$$\phi(\vec{r}, t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \epsilon(\vec{r}, \omega) \exp(-i\omega t) \quad (46)$$

Con la condición causal

$$\phi(\vec{r}, t) = 0, \quad t < 0 \quad (47)$$

Los métodos descritos no tienen presente los cristales fotónicos, en donde cada uno de los materiales utilizados pueden tener diferentes respuestas a la radiación incidente, para hacer un análisis de este tipo hay que tener en cuenta la deducción realizada en el artículo de Sakoda *et al.* (2001), en el que los autores el cálculo de $\vec{D}(\vec{r}, t)$ para dos tipos de materiales: uno con respuesta dispersiva y otro sin respuesta dispersiva.

Para aplicarlo al problema elegido se puede ver que

$$\epsilon(\vec{r}, \omega) = \epsilon_1 + (\epsilon_2(\omega) - \epsilon_1) s(\vec{r}) \quad (48)$$

Donde

$$s(\vec{r}) = \begin{cases} 1, & \text{en las barras} \\ 0, & \text{fuera de las barras} \end{cases} \quad (49)$$

En ese sentido la ecuación (46) quedaría como

$$\begin{aligned} \phi(\vec{r}, t) = & \frac{1}{2\pi} \epsilon_1 (1 - s(\vec{r})) \int_{-\infty}^{\infty} d\omega \exp \exp(-i\omega t) \\ & + \frac{1}{2\pi} s(\vec{r}) \int_{-\infty}^{\infty} d\omega \epsilon_2(\omega) \exp(-i\omega t) \end{aligned} \quad (50)$$

Como se sabe

$$\delta(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \exp \exp(-i\omega t) \quad (51)$$

Entonces, la ecuación (45) tomará la forma

$$\begin{aligned} \vec{D}(\vec{r}, t) = & \int_0^t dt' \{ \epsilon_1 (1 - s(\vec{r})) \delta(t - t') + s(\vec{r}) \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \epsilon_2(\omega) \\ & \exp \exp(-i\omega(t - t')) \} \vec{E}(\vec{r}, t') \\ = & \epsilon_1 (1 - s(\vec{r})) \vec{E}(\vec{r}, t) + s(\vec{r}) \int_0^t dt' \{ \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \epsilon_2(\omega) \\ & \exp \exp(-i\omega(t - t')) \} \vec{E}(\vec{r}, t') \end{aligned} \quad (52)$$

Si se conoce la transformada de Fourier inversa para una forma específica de 2 se puede escribir que

$$\vec{D}(\vec{r}, t) = \epsilon_1(1 - s(\vec{r}))\vec{E}(\vec{r}, t) + s(\vec{r}) \int_0^t dt' g(t - t') \vec{E}(\vec{r}, t') \quad (53)$$

Donde

$$g(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \epsilon_2(\omega) \exp \exp(-i\omega t) \quad (54)$$

En el caso de no conocer la forma de $g(t)$ se puede intentar resolver la ecuación (54) aproximando la integral de manera discreta, es decir

$$\begin{aligned} g(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \epsilon_2(\omega) \exp \exp(-i\omega t) \rightarrow g(n) \\ &= \frac{1}{2\pi} \sum_{r=-Rmin}^{Rmax} 2\pi \Delta f \epsilon_2(2\pi r \Delta f) \exp(-i2\pi r \Delta f n \Delta t) \end{aligned} \quad (55)$$

donde se ha realizado la asignación

- $t \rightarrow n\Delta t$
- $\omega = 2\pi f \rightarrow \Delta\omega = 2\pi\Delta f$
- $f \rightarrow r\Delta f$

Siendo n y r valores enteros correspondientes a la discretización de los intervalos de tiempo y frecuencia respectivamente.

Debido a que en este caso $g(t)$ ha sido aproximada con una discretización, el valor de $g(t)$ para un t dado dependerá del tamaño de Δf y Δt . En el caso de la simulación FDTD, Δt corresponde al paso temporal de la simulación, mientras que Δf puede ser escogido de forma arbitraria para realizar la integración y no necesariamente es el surgido de la relación $\Delta f = \frac{1}{maxtime\Delta t}$.

Es importante recordar además que el valor $Rmax$ es aquel para el cual la integral en su forma discreta converge al verdadero valor de $g(t)$ para un Δf escogido. Este método es el usado en el caso de no conocer la transformada de Fourier para $\epsilon(\omega)$.

A continuación, se planteará una relación de recurrencia para $\vec{D}$, a partir de la discretización de la ecuación (52). No está de más mencionar que es conveniente buscar una forma explícita para la función $g(t)$.

Como

$$\vec{D}(\vec{r}, t) = \epsilon_1(1 - s(\vec{r}))\vec{E}(\vec{r}, t) + s(\vec{r}) \int_0^t dt' \left\{ \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \epsilon_2(\omega) \exp \exp(-i\omega(t - t')) \right\} \vec{E}(\vec{r}, t')$$

entonces se puede escribir que

$$\vec{D}(\vec{r}, t) = \vec{T}(\vec{r}, t) + \frac{s(\vec{r})}{2\pi} \int_{-\infty}^{\infty} d\omega \epsilon_2(\omega) \exp \exp(-i\omega t) \vec{M}(\vec{r}, t, \omega) \quad (56)$$

Donde

$$\vec{T}(\vec{r}, t) = \epsilon_1(1 - s(\vec{r}))\vec{E}(\vec{r}, t) \quad (57)$$

$$\vec{M}(\vec{r}, t, \omega) = \int_0^t (i\omega t') \vec{E}(\vec{r}, t') \quad (58)$$

Como se observa $\vec{M}$ es una función que corresponde al peso de $\vec{E}$ para una frecuencia dada cuando se tiene una evolución temporal entre $[0, t]$. Hay que recordar que esto no corresponde a una transformada de Fourier pues no se ha tomado un intervalo en el cual el campo $\vec{E}$ sea periódico.

Entonces, para discretizar $\vec{M}$ se tiene que:

$$\vec{M}(\vec{r}, t, \omega) = \int_0^t (i\omega t') \vec{E}(\vec{r}, t') \rightarrow \vec{M}^n(\omega) = \sum_{j=0}^n \exp \exp(i\omega j \Delta t) \vec{E}^j \Delta t \quad (59)$$

Donde la discretización se realiza para cada punto de la grilla (i, j) , por tal motivo solo se tendrá en cuenta la variación temporal para los siguiente. De la ecuación (59) se tiene que

$$\begin{aligned} \vec{M}^n(\omega) &= \exp \exp(i\omega n \Delta t) \vec{E}^n \Delta t + \sum_{j=0}^{n-1} \exp \exp(i\omega j \Delta t) \vec{E}^j \Delta t \\ &= \exp \exp(i\omega n \Delta t) \vec{E}^n \Delta t + \vec{M}^{n-1}(\omega) \end{aligned} \quad (60)$$

Como se puede ver es una relación recurrente. Así pues, al discretizar la ecuación (56), se encuentra que

$$\vec{D}^n = \vec{T}^n + \frac{s}{2\pi} \sum_{r=-Rmax}^{Rmax} \epsilon_2(2\pi r \Delta f) \exp \exp(-i2\pi r \Delta f n \Delta t) \{ \exp \exp(i2\pi r \Delta f n \Delta t) \vec{E}^n \Delta t + \vec{M}^{n-1}(2\pi r \Delta f) \} (2\pi r \Delta f) \quad (61)$$

Si se reordenan algunos términos esta expresión puede escribirse como

$$\begin{aligned} \vec{D}^n &= \epsilon_1(1 - s) \vec{E}^n + s \Delta t F(Rmax) \vec{E}^n \\ &\quad + s \sum_{r=-Rmax}^{Rmax} \epsilon_2(2\pi r \Delta f) \exp(-i2\pi r \Delta f n \Delta t) \vec{M}^{n-1}(2\pi r \Delta f) r \Delta f \end{aligned} \quad (62)$$

Donde

$$F(Rmax) = \sum_{-Rmax}^{Rmax} \epsilon_2(2\pi r \Delta f) r \Delta f \quad (63)$$

Entonces, al resolver la ecuación (62) para $\vec{E}^n$ se tiene que

$$\vec{E}^n = \frac{\vec{D}^n - s \sum_{r=-Rmax}^{Rmax} \epsilon_2(2\pi r \Delta f) \exp(-i2\pi r \Delta f n \Delta t) \vec{M}^{n-1}(2\pi r \Delta f) r \Delta f}{\epsilon_1(1-s) + s \Delta t F(Rmax)} \quad (64)$$

Las ecuaciones que añaden a las rutinas del cálculo del método fdtf para actualizar los campos a partir del conocimiento del campo $\vec{D}$, son: (60), (63) y (64). Como ya se mencionó estas ecuaciones se deben evaluar para cada punto de la grilla de simulación.

De forma análoga es posible tener una representación para los campos $\vec{H}$ y $\vec{B}$ en el caso de una respuesta dispersiva $\mu(\omega)$.

Uno de los problemas con esta representación está en las posibles indeterminaciones de la función $\epsilon_2(\omega)$, pues para valores donde ella se indetermina, sucederá también en los campos E y D. El análisis de estas indeterminaciones debe hacerse usando técnicas propias del cálculo y en ocasiones de variables complejas.

Como ya se mencionó, estas dificultades se pueden resolver fácilmente si se conoce el valor de la función $g(t)$, ecuación (55), pues la implementación de las ecuaciones (60), (63) y (64) depende del valor de Rmax para que las funciones converjan. Asimismo, hay que recordar que el paso en frecuencias es diferente al usado en el caso del análisis del espectro al aplicar la transformada inversa de Fourier de los campos. Debe tenerse en cuenta que el campo $\vec{M}$ se debe describir para cada valor de frecuencias, así como para cada punto de la grilla y su evolución temporal.

Método FDTD al considerar la dispersión tipo Drude, materiales metálicos

Considérese la función dieléctrica (Sakoda et al., 2001)

$$\epsilon_2(\omega) = \epsilon_\infty \left(1 - \frac{\omega_p^2}{\omega(\omega + i\gamma)}\right) \quad (65)$$

Para la cual su transformada de Fourier está descrita como

$$g(t) = \epsilon_\infty \delta(t) + \frac{\epsilon_\infty \omega_p^2}{\gamma} (1 - \exp(-\gamma t)) \quad (66)$$

Con este resultado se puede remplazar la función $g(t)$ en la expresión para $\vec{D}(\vec{r}, t)$ y discretizar. Añadiendo así la sustitución para $\vec{E}(\vec{r}, t)$ en la rutina para el calculo del método FDTD.

Como se sabe de las ecuaciones (53) y (57) es posible escribir que

$$\vec{D}(\vec{r}, t) = \vec{T}(\vec{r}, t) + s(t) \int_0^t g(t-t') \vec{E}(\vec{r}, t') dt' \quad (67)$$

Así pues, al remplazar la relación (66) toma la forma

$$\begin{aligned}\vec{D}(\vec{r}, t) = & \vec{T}(\vec{r}, t) + s(\vec{r})\epsilon_{\infty}\vec{E}(\vec{r}, t) \\ & + \frac{s(\vec{r})\epsilon_{\infty}\omega_p^2}{\gamma} \left[\int_0^t \vec{E}(\vec{r}, t') dt' \right. \\ & \left. - \int_0^t \exp(-\gamma(t-t'))\vec{E}(\vec{r}, t') dt' \right]\end{aligned}\quad (68)$$

Si se discretizan estas últimas integrales se tiene que

$$\begin{aligned}\vec{I}_1(t) = \int_0^t \vec{E}(\vec{r}, t') dt' \rightarrow \vec{I}_1^n &= \sum_{i=0}^n \vec{E}^i \Delta t = \vec{E}^n \Delta t + \sum_{i=0}^{n-1} \vec{E}^i \Delta t \\ &= \vec{E}^n \Delta t + \vec{I}_1^{n-1}\end{aligned}\quad (69)$$

De forma análoga se puede discretizar

$$\int_0^t \exp(-\gamma(t-t'))\vec{E}(\vec{r}, t') dt' = \exp(-\gamma t) \int_0^t \exp(\gamma t')\vec{E}(\vec{r}, t') dt'$$

donde se define y discretiza que

$$\begin{aligned}\vec{I}_2(t) = \int_0^t \exp \exp(\gamma t') \vec{E}(\vec{r}, t') dt' \rightarrow \vec{I}_2^n &= \sum_{i=0}^n \exp(\gamma i \Delta t) \vec{E}^i \Delta t \\ &= \exp(\gamma n \Delta t) \vec{E}^n \Delta t + \vec{I}_2^{n-1}\end{aligned}\quad (70)$$

Así pues, se tiene que la expresión (68) en su forma discreta será

$$\vec{D}^n = \epsilon_1(1-s)\vec{E}^n + s\epsilon_{\infty}\vec{E}^n + \frac{s\epsilon_{\infty}\omega_p^2}{\gamma} [\vec{I}_1^{n-1} + \exp \exp(-\gamma n \Delta t) \vec{I}_2^{n-1}] \quad (71)$$

Al resolver para $\vec{E}^n$ se obtiene

$$\vec{E}^n = \frac{\vec{D}^n - \frac{s\epsilon_{\infty}\omega_p^2}{\gamma} [\vec{I}_1^{n-1} + \exp \exp(-\gamma n \Delta t) \vec{I}_2^{n-1}]}{\epsilon_1(1-s) + s\epsilon_{\infty}} \quad (72)$$

Donde además hay que imponer las condiciones $\vec{I}_1^0 = 0$ y $\vec{I}_2^0 = 0$ a las relaciones (69) y (70) respectivamente.

Es importante recordar que cada una de las integrales auxiliares, así como los campos y la función s , son evaluadas para lugares de la grilla de simulación, es decir, son evaluadas en la posición $\vec{r} \rightarrow (i, j)$

Además, todos los campos evaluados toman el valor de 0, pues al inicio no hay energía en el sistema simulado. Por otra parte, la notación utilizada es vectorial por lo que en algunos casos se deberán definir las variaciones para cada componente de los campos siguiendo las relaciones antes encontradas.

En el caso que no exista absorción en el material, esto es en la ecuación (65) $\gamma = 0$, la transformada de Fourier de la función dieléctrica tomará la forma

$$g(t) = \epsilon_2 \delta(t) + \epsilon_2 \omega_p^2 t \quad (73)$$

Entonces la ecuación (67) será

$$\begin{aligned} \vec{D}(\vec{r}, t) = & \vec{T}(\vec{r}, t) + s(\vec{r})\epsilon_2 \vec{E}(\vec{r}, t) \\ & + s(\vec{r})\epsilon_2 \omega_p^2 \left[t \int_0^t \vec{E}(\vec{r}, t') dt' - \int_0^t t' \vec{E}(\vec{r}, t') dt' \right] \end{aligned} \quad (74)$$

Cuya discretización se puede realizar con la relación (69) y con la siguiente relación

$$\vec{I}_2(t) = \int_0^t t' \vec{E}(\vec{r}, t') dt' \rightarrow \vec{I}_2^n = \sum_{i=0}^n i \Delta t \vec{E}^i \Delta t = n \Delta t^2 \vec{E}^n + \vec{I}_2^{n-1} \quad (75)$$

Así, la forma discreta de la relación (74) será

$$\vec{D}^n = \epsilon_1(1-s)\vec{E}^n + s\epsilon_2 \vec{E}^n + s\epsilon_2 \omega_p^2 [n \Delta t \vec{I}_1^{n-1} - \vec{I}_2^{n-1}] \quad (76)$$

Resolviendo para $\vec{E}^n$ se tiene que

$$\vec{E}^n = \frac{\vec{D}^n - s\epsilon_2 \omega_p^2 [n \Delta t \vec{I}_1^{n-1} - \vec{I}_2^{n-1}]}{\epsilon_1(1-s) + s\epsilon_2} \quad (77)$$

Otra forma de ver este tipo de discretización es considerar la relación

$$\int_0^t t' \vec{E}(\vec{r}, t') dt' \rightarrow \Delta t^2 \sum_{i=0}^n i \vec{E}^i = \Delta t^2 \vec{\rho}^n = \Delta t^2 [n \vec{E}^n + \vec{\rho}^{n-1}]$$

Donde

$$\vec{\rho}^n = \sum_{i=0}^n i \vec{E}^i$$

Con esta expresión se tiene que

$$\vec{D}^n = \epsilon_1(1-s)\vec{E}^n + s\epsilon_2 \vec{E}^n + s\epsilon_2 \omega_p^2 [n \Delta t \vec{I}_1^{n-1} - \Delta t^2 \vec{\rho}^{n-1}]$$

y al resolver para $\vec{E}^n$ se tendrá

$$\vec{E}^n = \frac{\vec{D}^n - s\epsilon_2 \omega_p^2 [n \Delta t \vec{I}_1^{n-1} - \Delta t^2 \vec{\rho}^{n-1}]}{\epsilon_1(1-s) + s\epsilon_2}$$

Método FDTD aplicado a sistemas metamateriales

En este caso se considera una respuesta eléctrica y magnética dispersiva para los materiales que constituyen el cristal fotónico. En particular, se considerará la respuesta dispersiva utilizada en el artículo de Reyes *et al.* (2009):

$$\epsilon_2(\omega) = \epsilon_2 - \frac{\omega_e^2}{\omega^2}, \quad \mu_2(\omega) = \mu_2 - \frac{\omega_m^2}{\omega^2} \quad (78)$$

Debido a la respuesta dispersiva para los materiales, ecuación (45), es posible escribir para los campos eléctrico y magnético lo siguiente

$$\vec{D}(\vec{r}, t) = \int_0^t dt' \phi_e(\vec{r}, t - t') \vec{E}(\vec{r}, t') \quad (79)$$

$$\vec{B}(\vec{r}, t) = \int_0^t dt' \phi_m(\vec{r}, t - t') \vec{H}(\vec{r}, t') \quad (80)$$

donde ϕ_e y ϕ_m son las transformadas de Fourier para la respuesta eléctrica y magnética respectivamente. Debido a que la permitividad eléctrica y la permeabilidad magnética pueden escribirse como

$$\epsilon(\vec{r}, \omega) = \epsilon_1 + (\epsilon_2(\omega) - \epsilon_1)s(\vec{r}) \quad (81)$$

$$\mu(\vec{r}, \omega) = \mu_1 + (\mu_2(\omega) - \mu_1)s(\vec{r}) \quad (82)$$

donde $s(\vec{r})$ esta dada como en la ecuación (49). Como ya se mostró para el tipo de respuesta dispersiva se obtendrán transformadas de Fourier similares a las mostradas en la ecuación (73) tanto para la respuesta eléctrica como para la magnética, por tal motivo habrá una deducción similar a la que dirigió a la ecuación (77) para la discretización de los campos. Por ello en el núcleo fdtd se puede escribir que

$$\vec{E}^n = \frac{\vec{D}^n - s\epsilon_2\omega_e^2[n\Delta t \vec{I}_{e1}^{n-1} - \vec{I}_{e2}^{n-1}]}{\epsilon_1(1-s) + s\epsilon_2} \quad (83)$$

Con

$$\begin{aligned} \vec{I}_{e1}^n &= \vec{E}^n \Delta t + \vec{I}_{e1}^{n-1} \\ \vec{I}_{e2}^n &= n\Delta t^2 \vec{E}^n + \vec{I}_{e2}^{n-1} \end{aligned} \quad (84)$$

para lo cual se cumple $\vec{I}_{e1}^0 = 0$ y $\vec{I}_{e2}^0 = 0$. De forma análoga se tiene que

$$\vec{H}^n = \frac{\vec{B}^n - s\mu_2\omega_m^2[n\Delta t \vec{I}_{m1}^{n-1} - \vec{I}_{m2}^{n-1}]}{\mu_1(1-s) + s\mu_2} \quad (85)$$

Donde

$$\begin{aligned} \vec{I}_{m1}^n &= \vec{H}^n \Delta t + \vec{I}_{m1}^{n-1} \\ \vec{I}_{m2}^n &= n\Delta t^2 \vec{H}^n + \vec{I}_{m2}^{n-1} \end{aligned} \quad (86)$$

para lo cual se cumple $\vec{I}_{m1}^0 = 0$ y $\vec{I}_{m2}^0 = 0$

Es importante recordar que todos los campos al igual que las integrales y la función s son evaluados en puntos de la grilla.

CAPÍTULO 4:

Método FDTD utilizando un esquema de derivadas, otra deducción

Resumen del capítulo

En este capítulo mostraremos otra forma de expresar las ecuaciones usadas en el núcleo de evolución temporal ftdt, en el cual se considera un conjunto de funciones derivables. Este método tiene una ventaja sobre el desarrollado anteriormente y es que las ecuaciones no dependen explícitamente de la frecuencia de la radiación incidente.

Formulación

Considérese un sistema no magnético compuesto por barras de un material que posee respuesta dispersiva. En el caso de utilizar el vector polarización $\vec{P}$ (Kittel, 1966; Raman y Fan, 2010; Sanshui Xiao, 2002) se puede escribir que

$$\vec{D} = \epsilon_0 \vec{E} + \vec{P} \quad (87)$$

Si al considerar que el sistema presenta una respuesta efectiva que puede escribirse como

$$\vec{D} = \epsilon_0 \epsilon_r(\omega) \vec{E} \quad (88)$$

entonces, se puede escribir para el vector polarización

$$\vec{P} = -\epsilon_0 \vec{E} + \epsilon_0 \epsilon_r(\omega) \vec{E} \quad (89)$$

Por otro lado, en el caso de considerar una variación temporal se podrá escribir que

$$\vec{P}(t) = -\epsilon_0 \vec{E}(t) + \epsilon_0 \int_0^t g(t-t') \vec{E}(t') dt' \quad (90)$$

siendo la función

$$g(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \epsilon_r(\omega) \exp(-i\omega t) d\omega \quad (91)$$

con $g(t)=0$ cuando $t < 0$.

Como se puede ver el vector polarización podrá definirse si se conoce la variación temporal representada por la función $g(t)$.

Con el uso del vector polarización es posible escribir las ecuaciones de Maxwell, ecuaciones (6) como

$$\nabla \times \vec{E} = -\mu_0 \frac{\partial \vec{H}}{\partial t}, \quad \nabla \times \vec{H} = \epsilon_0 \frac{\partial \vec{E}}{\partial t} + \vec{P}_t \quad (92)$$

Donde $\vec{P}_t = \frac{\partial \vec{P}}{\partial t}$, función que depende exclusivamente de la forma de $g(t)$.

Al usar la ecuación (90) se puede escribir que

$$\vec{P}_t = -\epsilon_0 \frac{\partial \vec{E}}{\partial t} + \epsilon_0 \frac{\partial}{\partial t} \left[\int_0^t g(t-t') \vec{E}(t') dt' \right] \quad (93)$$

Pero de acuerdo con la fórmula de Leibniz se puede escribir que, si una función

$$F(x) = \int_{g(x)}^{h(x)} f(t, x) dt \rightarrow F'(x) = \int_{g(x)}^{h(x)} \frac{\partial f(t, x)}{\partial x} dt + f(h(x), x)h'(x) - f(g(x), x)g'(x)$$

de forma que

$$\vec{P}_t = -\epsilon_0 \frac{\partial \vec{E}}{\partial t} + \epsilon_0 \left[\int_0^t g'(t-t') \vec{E}(t') dt' + g(0) \vec{E}(t) \right] \quad (94)$$

donde

$$g'(t-t') = \frac{dg(t-t')}{dt} \quad (95)$$

en caso de ser $g(t)$ una función continua y con valor $g(0)=0$, es posible reducir la expresión anterior a

$$\vec{P}_t = -\epsilon_0 \frac{\partial \vec{E}}{\partial t} + \epsilon_0 \vec{R}(t) \quad (96)$$

donde

$$\vec{R}(t) = \int_0^t g'(t-t') \vec{E}(t') dt' \quad (97)$$

Se puede expresar esta última relación en términos de otras funciones vectoriales que dependerán de derivadas de ordenes superiores en la función $g(t)$. Es por esta razón que es necesario conocer la forma explícita de la función $g(t)$, pues el proceso de remplazo puede continuar si las derivadas de ordenes superior de $g(t)$ no se reducen.

Pero, ¿cuál es el objetivo de este análisis?, se busca expresar $\vec{P}_t$ como un sistema de ecuaciones que se pueda integrar al esquema fdtd, en el cual no existen integrales explícitas. Como se demostró, el uso del vector polarización se puede aplicar a un esquema de diferencias finitas al discretizar la integración para el vector $\vec{P}$. Esto es posible de hacer en el caso de formas específicas para la respuesta material.

En el caso de una respuesta dispersiva tipo Drude sin absorción

$$\epsilon_r(\omega) = 1 - \frac{\omega_p^2}{\omega^2} \quad (98)$$

se tiene que

$$\begin{aligned}\vec{P}_t &= -\epsilon_0 \frac{\partial \vec{E}}{\partial t} + \epsilon_0 \frac{\partial}{\partial t} \left[\int_0^t [\delta(t-t') + \omega_p^2(t-t')] \vec{E}(t') dt' \right] \\ &= \epsilon_0 \omega_p^2 \frac{\partial}{\partial t} \left[\int_0^t (t-t') \vec{E}(t') dt' \right]\end{aligned}\quad (99)$$

donde se ha hecho uso de la relación $g(t) = \delta(t) + t\omega_p^2$.

Por lo tanto, al aplicar la fórmula de Leibnitz a la última integral se tiene que

$$\vec{P}_t = \epsilon_0 \omega_p^2 \int_0^t \vec{E}(t') dt' \quad (100)$$

pero de esta última relación se puede hacer

$$\frac{\partial \vec{P}_t}{\partial t} = \epsilon_0 \omega_p^2 \frac{\partial}{\partial t} \int_0^t \vec{E}(t') dt' = \epsilon_0 \omega_p^2 \vec{E}(t) \quad (101)$$

Es decir, para esta forma específica de la función respuesta se tiene que el sistema de ecuaciones a discretizar en un esquema ftdt serán las relaciones (92) y (101)

La ventaja de esta propuesta es que no existe una dependencia explícita con para ninguno de los campos.

Ahora, si se aplica este esquema a un cristal fotónico sin tomar en cuenta el vector polarización y suponiendo una respuesta dispersiva para la permitividad dieléctrica, la función para este sistema será de la forma de la ecuación (48)

$$\epsilon(\omega) = \epsilon_1 + (\epsilon_2(\omega) - \epsilon_1)s(\vec{r})$$

Así, la relación (74) se puede escribir como

$$\vec{D}(\vec{r}, t) = \epsilon_1(1 - s(\vec{r}))\vec{E}(\vec{r}, t) + \epsilon_2 s(\vec{r}) \left\{ \vec{E}(\vec{r}, t) + \omega_p^2 \int_0^t (t-t') \vec{E}(\vec{r}, t') dt' \right\} \quad (102)$$

Si se define

$$\vec{R}(\vec{r}, t) = \int_0^t (t-t') \vec{E}(\vec{r}, t') dt' \quad (103)$$

Entonces se tendrá que

$$\vec{E}(\vec{r}, t) = \frac{\vec{D}(\vec{r}, t) - \epsilon_2 s(\vec{r}) \omega_p^2 \vec{R}(\vec{r}, t)}{\epsilon_1(1 - s(\vec{r})) + \epsilon_2 s(\vec{r})} \quad (104)$$

Ahora, para el campo $\vec{R}(\vec{r}, t)$ se puede mostrar que

$$\frac{\partial \vec{R}(\vec{r}, t)}{\partial t} = \int_0^t \vec{E}(\vec{r}, t') dt' = \vec{I}(\vec{r}, t) \quad (105)$$

Y, además

$$\frac{\partial \vec{I}(\vec{r}, t)}{\partial t} = \vec{E}(t) \quad (106)$$

En este esquema se discretizan las ecuaciones de Maxwell, ecuaciones (6), junto con las ecuaciones (104), (105) y (106).

La principal dificultad para implementación de este esquema consiste en decidir cuáles son los tiempos para la actualización de los campos, hay que recordar que en el esquema ftdt los campos se actualizan en tiempos diferentes.

Para aplicar correctamente el método ftdt se puede proponer actualizar los campos de la siguiente manera:

- a. Al actualizar primero el campo $\vec{B}$.

$$\begin{aligned} \nabla \times \vec{E} &= \frac{-1}{c} \frac{\partial \vec{B}}{\partial t} \rightarrow t = n \rightarrow \left(\vec{B}^{n+\frac{1}{2}} - \vec{B}^{n-\frac{1}{2}} \right) \propto \vec{E}^n \\ \frac{\partial \vec{I}}{\partial t} &= \vec{E} \rightarrow t = n \rightarrow \left(\vec{I}^{n+\frac{1}{2}} - \vec{I}^{n-\frac{1}{2}} \right) \propto \vec{E}^n \\ \frac{\partial \vec{R}}{\partial t} &= \vec{I} \rightarrow t = n + \frac{1}{2} \rightarrow \left(\vec{R}^{n+1} - \vec{R}^n \right) \propto \vec{I}^{n+\frac{1}{2}} \\ \nabla \times \vec{H} &= \frac{1}{c} \frac{\partial \vec{D}}{\partial t} \rightarrow t = n + \frac{1}{2} \rightarrow \left(\vec{D}^{n+1} - \vec{D}^n \right) \propto \vec{H}^{n+\frac{1}{2}} \\ \vec{E} &= \frac{\vec{D} - \epsilon_2 s \omega_p^2 \vec{R}}{\epsilon_1 (1-s) + \epsilon_2 s} \rightarrow t = n + 1 \rightarrow \vec{E}^{n+1} \propto \vec{R}^{n+1}, \vec{D}^{n+1} \end{aligned}$$

- b. Luego actualizar primer el campo $\vec{D}$.

$$\begin{aligned} \frac{\partial \vec{I}}{\partial t} &= \vec{E} \rightarrow t = n - \frac{1}{2} \rightarrow \left(\vec{I}^n - \vec{I}^{n-1} \right) \propto \vec{E}^{n-\frac{1}{2}} \\ \frac{\partial \vec{R}}{\partial t} &= \vec{I} \rightarrow t = n \rightarrow \left(\vec{R}^{n+\frac{1}{2}} - \vec{R}^{n-\frac{1}{2}} \right) \propto \vec{I}^n \\ \nabla \times \vec{H} &= \frac{1}{c} \frac{\partial \vec{D}}{\partial t} \rightarrow t = n \rightarrow \left(\vec{D}^{n+\frac{1}{2}} - \vec{D}^{n-\frac{1}{2}} \right) \propto \vec{H}^n \\ \vec{E} &= \frac{\vec{D} - \epsilon_2 s \omega_p^2 \vec{R}}{\epsilon_1 (1-s) + \epsilon_2 s} \rightarrow t = n + \frac{1}{2} \rightarrow \vec{E}^{n+\frac{1}{2}} \propto \vec{R}^{n+\frac{1}{2}}, \vec{D}^{n+\frac{1}{2}} \\ \nabla \times \vec{E} &= \frac{-1}{c} \frac{\partial \vec{B}}{\partial t} \rightarrow t = n + \frac{1}{2} \rightarrow \left(\vec{B}^{n+1} - \vec{B}^n \right) \propto \vec{E}^{n+\frac{1}{2}} \end{aligned}$$

Como se observa las dos propuestas son similares, solo se diferencian en el campo que se actualiza primero. Por otra parte, se puede observar que los campos se actualizan siguiendo la secuencia

$$n \rightarrow n + 1, \quad n - \frac{1}{2} \rightarrow n + \frac{1}{2}$$

Además, las variaciones en el espacio se realizan de acuerdo con las relaciones de los campos en la celda de Yee.

Además, es posible mostrar que si se considera el vector polarización y se lo aplica a un cristal fotónico con barras que presentan dispersión tipo Drude se tiene que

$$\vec{P}(\vec{r}, t) = \epsilon_0 h(\vec{r}) \vec{E}(\vec{r}, t) + \epsilon_0 \epsilon_2 s(\vec{r}) \omega_p^2 \int_0^t (t - t') \vec{E}(\vec{r}, t') dt' \quad (107)$$

Con

$$h(\vec{r}) = \epsilon_1 (1 - s(\vec{r})) + s(\vec{r}) \epsilon_2 - 1 \quad (108)$$

de donde se puede escribir que

$$\begin{aligned} \frac{\partial \vec{P}(\vec{r}, t)}{\partial t} &= \epsilon_0 h(\vec{r}) \frac{\partial \vec{E}(\vec{r}, t)}{\partial t} + \epsilon_0 \epsilon_2 s(\vec{r}) \omega_p^2 \frac{\partial}{\partial t} \left[\int_0^t (t - t') \vec{E}(\vec{r}, t') dt' \right] \\ &= \epsilon_0 h(\vec{r}) \frac{\partial \vec{E}(\vec{r}, t)}{\partial t} + \epsilon_0 \epsilon_2 s(\vec{r}) \omega_p^2 \vec{I}(\vec{r}, t) \end{aligned} \quad (109)$$

donde

$$\vec{I}(\vec{r}, t) = \int_0^t \vec{E}(\vec{r}, t') dt'$$

Nótese que para $\vec{I}(\vec{r}, t)$ se cumple de nuevo la relación (106). Así, se tiene que de la ecuación (92) se puede escribir

$$\frac{\partial \vec{E}(\vec{r}, t)}{\partial t} = \frac{1}{\epsilon_0} \left[\nabla \times \vec{H}(\vec{r}, t) - \frac{\partial \vec{P}(\vec{r}, t)}{\partial t} \right] \quad (110)$$

Al reemplazar la relación (109) se tiene que

$$\frac{\partial \vec{E}(\vec{r}, t)}{\partial t} = \frac{\nabla \times \vec{H}(\vec{r}, t) - \epsilon_0 \epsilon_2 s(\vec{r}) \omega_p^2 \vec{I}(\vec{r}, t)}{\epsilon_0 \epsilon_1 (1 - s(\vec{r})) + \epsilon_0 \epsilon_2 s(\vec{r})} \quad (111)$$

Bajo este esquema, las ecuaciones a discretizar en el formalismo FDTD serán (106), (111) y la relación

$$\frac{\partial \vec{H}(\vec{r}, t)}{\partial t} = -\frac{1}{\mu_0} \nabla \times \vec{E}(\vec{r}, t)$$

Como puede verse, en este esquema solo son necesarias tres ecuaciones para la evolución temporal en el método FDTD. De forma análoga a lo discutido antes se debe escoger como se actualizarán los campos, para tal caso se plantean las siguientes propuestas

- a. Al actualizar primero el campo $\vec{H}$

$$\begin{aligned}\frac{\partial \vec{H}}{\partial t} &= -\frac{1}{\mu_0} \nabla \times \vec{E} \rightarrow t = n \rightarrow \left(\vec{H}^{n+\frac{1}{2}} - \vec{H}^{n-\frac{1}{2}} \right) \propto \vec{E}^n \\ \frac{\partial \vec{I}}{\partial t} &= \vec{E} \rightarrow t = n \rightarrow \left(\vec{I}^{n+\frac{1}{2}} - \vec{I}^{n-\frac{1}{2}} \right) \propto \vec{E}^n \\ \frac{\partial \vec{E}}{\partial t} &= \frac{\nabla \times \vec{H} - \epsilon_0 \epsilon_2 s \omega_p^2 \vec{I}}{\epsilon_0 (\epsilon_1 (1-s) + \epsilon_2 s)} \rightarrow t = n + \frac{1}{2} \rightarrow \left(\vec{E}^{n+1} - \vec{E}^n \right) \propto \vec{H}^{n+\frac{1}{2}}, \vec{I}^{n+\frac{1}{2}}\end{aligned}$$

- b. Al actualizar primero el campo $\vec{E}$

$$\begin{aligned}\frac{\partial \vec{E}}{\partial t} &= \frac{\nabla \times \vec{H} - \epsilon_0 \epsilon_2 s \omega_p^2 \vec{I}}{\epsilon_0 (\epsilon_1 (1-s) + \epsilon_2 s)} \rightarrow t = n \rightarrow \left(\vec{E}^{n+\frac{1}{2}} - \vec{E}^{n-\frac{1}{2}} \right) \propto \vec{H}^n, \vec{I}^n \\ \frac{\partial \vec{H}}{\partial t} &= -\frac{1}{\mu_0} \nabla \times \vec{E} \rightarrow t = n + \frac{1}{2} \rightarrow \left(\vec{H}^{n+1} - \vec{H}^n \right) \propto \vec{E}^{n+\frac{1}{2}} \\ \frac{\partial \vec{I}}{\partial t} &= \vec{E} \rightarrow t = n + \frac{1}{2} \rightarrow \left(\vec{I}^{n+1} - \vec{I}^n \right) \propto \vec{E}^{n+\frac{1}{2}}\end{aligned}$$

ANEXOS

Algoritmos desarrollados

En este apartado se presentan los diferentes algoritmos desarrollados para que los estudiantes los usen y modifiquen a gusto. Como ya fue indicado el lenguaje corresponde a Fortran, por lo cual se necesitará un compilador que permita su ejecución. Se recomienda utilizar el compilador de distribución libre para Windows, así como un ambiente de trabajo como el del proyecto Force-Fortran, el cual puede ser hallado en la web (<http://force.lepsch.com/p/download.html>). Otra forma de trabajar los algoritmos puede hacerse con la utilidad Cygwin la cual emula una consola de Linux en Windows, pueden descargarse de sus librerías los correspondientes compiladores para su facilidad de uso (<https://www.cygwin.com/>). Por último, si se cuenta con distribuciones de Linux, es posible instalar un compilador de Fortran si se usa el administrador de paquetes.

En principio, estos algoritmos están propuestos para el problema discutido en el documento. No obstante, la variación en la geometría y en el tipo de materiales considerados pueden ser aplicados fácilmente con las consideraciones presentadas en el documento.

Consideraciones para la implementación algorítmica

Para realizar la implementación numérica del método fdtd se debe tener presente lo siguiente:

- En la implementación numérica se tienen dos tipos diferentes de pulsos que es la manera de dar energía al sistema, una de ellas descrita en el artículo de Ashutosh y Pradip (2012), la cual es una fuente variable y la otra es la descrita en el libro de Sullivan (2000, p. 50) ecuación 37, la cual es un pulso gaussiano.
- El lugar del pulso para la radiación de energía al sistema es arbitrario. Cabe enunciar que en la literatura se hace énfasis en la necesidad de tener múltiples fuentes para que se existan la mayor cantidad de modos propios del sistema.
- El análisis de estos datos se debe realizar en puntos de la grilla diferentes a los puntos donde se colocan los pulsos.
- Tener presente que el a mayor número de puntos de la grilla de simulación mayor tiempo en la propagación de la información.
- La implementación de la dft supone que la señal utilizada es periódica.
- El análisis espectral puede hacerse para diferentes puntos y comparar los picos de resonancia para todos ellos.

Algoritmo FDTD aplicado al modo TE

En este algoritmo se aplica la evolución temporal para el caso de una grilla cuadrada, de barras paralelas de un material dieléctrico de sección transversal circular en aire, al considerar el modo te. Se analiza solo un punto de la grilla para realizar la transformada de Fourier y por ende el diagrama espectral. La información del cambio de la intensidad del campo estudiado con el tiempo está en el archivo fdtdTEvsT-g-x.dat que es entregado por el programa.

```

module varprogram
    real,parameter :: pi=3.1416
    real,parameter:: f=0.04 !factor de ocupación
    integer,parameter::ng=60!tamaño de la grilla en x(y), será número par, c=2.9979*(10.**8)
end module varprogram
program ftdTE
    use varprogram
    implicit none
    complex, dimension (:,:) , allocatable::Hz,Bz,Dy,Ey,Dx,Ex,px,I1x
    complex,dimension(:,:),allocatable::py,I1y
    real,dimension(:,:),allocatable::mu,ep
    real,dimension(:,:),allocatable::HzvsW
    real,dimension(2)::k,a1,a2,Rx,Ry
    complex::campo,suma,im !número imaginario
    integer :: maxtime,n,i,j,maxfre,fre,tiempo,io,PT,nk,r
    real::tao,Am,fr,pul,t0,e0,e1,e2,dt,dex,dely,c,a,spread,wp,s
    real::imp
    maxtime=20000! tiempo de la simulación
    maxfre=200! unidad máxima de frecuencia en unidades de df
    nk=30! número de particiones de los vectores k
    im=(0,1)
    a=0.01! parámetro de red en metros
    c=299792458.! velocidad de la luz en metros/segundo
    e0=8.854187*10**(-7)
    e1=1.
    e2=8.9
    wp=2.*pi*c/a! Frecuencia de plasma eléctrico

    t0=20
    spread=6.
    PT=int(t0+sqrt(2.*spread*log(10.**(10))))!tiempo máximo de duración del pulso para cambiar las
    condiciones iniciales

    allocate (ep(ng+1,ng+1))
    allocate (mu(ng+1,ng+1))
    allocate (Hz(ng+1,ng+1))
    allocate (Bz(ng+1,ng+1))
    allocate (Dy(ng+1,ng+1))
    allocate (Ey(ng+1,ng+1))
    allocate (Dx(ng+1,ng+1))
    allocate (Ex(ng+1,ng+1))
    a1(1)=a
    a1(2)=0.
    a2(1)=0.

```

```

a2(2)=a
Rx=a1
Ry=a2
delx=a/ng
dely=delx
dt=1./(2.*c*sqrt(1./(delx*delx)+1./(dely*dely)))
print*, "delx=dely=", dely, "dt=", dt, "df=", 1./(maxtime*dt), "wp=", wp, "
&fp=wp/(2*pi)=", wp/(2*pi), "rp=fp/df=", wp/(2*pi)/(1./(maxtime*dt))

open(unit=1, file="fdtdTEvsT-g-x.dat", status="unknown")
mu(:, :) = 1. !valor permeabilidad magnética, material no magnético mu=1.

do i=1, ng+1
  do j=1, ng+1
    if (((i-(ng/2))**2+(j-(ng/2))**2)).le.f*ng*ng then
      ep(i,j)=e2
      mu(i,j)=1.
    else
      ep(i,j)=e1
      mu(i,j)=1.
    end if
  end do
end do

!nucleo del mtodo FDTD
Print*, "calculo nucleo FDTD"
k(2)=0. !es el mismo para todos los puntos gamma-x
do r=0, nk

  k(1)=real(r)*(pi/(a*real(nk)))

  Hz(:, :) = 0. !condicion inicial para los campos...
  Bz(:, :) = 0.
  Dx(:, :) = 0.
  Dy(:, :) = 0.
  Ex(:, :) = 0.
  Ey(:, :) = 0.

  do n=0, maxtime

    do i=1, ng+1
      do j=2, ng+1
        Dx(i,j)=Dx(i,j)+c*dt*(hz(i,j)-hz(i,j-1))/dely
      end do
    end do
  end do

```

```

do i=1,ng+1
  Dx(i,1)=Dx(i,1)+c*dt*(hz(i,1)-exp(-im*dot_product(k,Ry))*hz(i,ng+1
&))/dely
end do

do i=2,ng+1
  do j=1,ng+1
    Dy(i,j)=Dy(i,j)-c*dt*(hz(i,j)-hz(i-1,j))/delx
  end do
end do
do j=1,ng+1
  Dy(1,j)=Dy(1,j)-c*dt*(hz(1,j)-exp(-im*dot_product(k,Rx))*Hz(ng+1,j
&))/delx
end do

Ex(:,)=Dx(:,)/ep(:,) !equivalencia entre los campos D y E
Ey(:,)=Dy(:,)/ep(:,) !equivalencia entre los campos D y E

do i=1,ng
  do j=1,ng
    Bz(i,j)=Bz(i,j)-c*dt*(((Ey(i+1,j)-Ey(i,j))/delx)-((Ex(i,j+1)-Ex(i,
&j))/dely))
  end do
end do
do j=1,ng
  Bz(ng+1,j)=Bz(ng+1,j)-c*dt*(((exp(im*dot_product(k,Rx))*Ey(1,j)-Ey
&(ng+1,j))/delx)-((Ex(ng+1,j+1)-Ex(ng+1,j))/dely))
end do

do i=1,ng
  Bz(i,ng+1)=Bz(i,ng+1)-c*dt*(((Ey(i+1,ng+1)-Ey(i,ng+1))/delx)-((exp
&(im*dot_product(k,Ry))*Ex(i,1)-Ex(i,ng+1))/dely))
end do

Bz(ng+1,ng+1)=Bz(ng+1,ng+1)-c*dt*(((exp(im*dot_product(k,Rx))*Ey(1
&,ng+1)-Ey(ng+1,ng+1))/delx)-((exp(im*dot_product(k,Ry))*Ex(ng+1,1)
&-Ex(ng+1,ng+1))/dely))

Hz(:,)=Bz(:,)/mu(:,) !equivalencia entre los campos B y H

if (n.le.PT) then
  pul=exp(-0.5*(real(t0-n)*real(t0-n))/spread) ! propuestas de pulsos, pulso 1
  !pul=Am*exp(im*2.*pi*fr*dt*n)*exp(-(n*dt-t0)/tao)*((n*dt-t0)/tao) ! pulso 2
  Bz(ng/2,ng/2)=pul !lugar donde se aplica el pulso
  Hz(ng/2,ng/2)=Bz(ng/2,ng/2)/mu(ng/2,ng/2)

```

```

end if
i=8 !lugar escogido para revisar su dependencia en el tiempo.
j=i
write(1,*)r,n,HZ(i,j)!con esta orden se guardan los datos del punto donde se hace el análisis
!print*,n,Dz(i,j)!con esta orden se puede ver como se transmite la información al lugar donde se
decida hacer el análisis
end do
end do

close(unit=1)

deallocate(Hz,Bz,Dx,Ex,Dy,Ey)

print*, "FIN DEL PROGRAMA"
end program

real function s(i,j) result(r)
use varprogram
implicit none
integer, intent(in)::i,j
!real::r
if (((i-(ng/2))**2+(j-(ng/2))**2)).le.f*ng*ng/pi then
    r=1.
else
    r=0.
end if
end function s

```

Algoritmo FDTD aplicado al modo TM

En este algoritmo se aplica la evolución temporal para el caso de una grilla cuadrada, de barras paralelas de sección transversal circular de un material dieléctrico en aire, al considerar el modo TM. Se analiza solo un punto de la grilla para realizar la transformada de Fourier y por ende el diagrama espectral. La información del cambio de la intensidad del campo estudiado con el tiempo está en el archivo fdtdTMvsT-g-x.dat que es entregado por el programa.

```

module varprogram
    real,parameter :: pi=3.1416
    real,parameter:: f=0.04 !factor de ocupación
    integer,parameter::ng=60!tamaño de la grilla en x(y), será número par, c=2.9979*(10.**8)
end module varprogram

program fdtdTM
implicit none

```

```

complex, dimension (:,:) , allocatable::Dz,Ez,By,Hy,Hx,Bx
real,dimension(:,:),allocatable::ep,mu
real,dimension(:,:),allocatable::DzvsW
real,dimension(2)::k,a1,a2,Rx,Ry
complex::campo,suma,im! número imaginario
integer :: ng,maxtime,n,i,j,maxfre,fre,tiempo,io,PT,nk,r
real::tao,Am,fr,pul,t0,f,pi,e0,e1,e2,dt,delx,dely,c,a,spread,wp
real::imp
maxtime=20000 ! tiempo de la simulación
ng=60! tamaño de la grilla en x(y), será número par
nk=30! número de particiones de los vectores k
f=0.04! factor de ocupación
pi=3.1416! valor de pi usado en esta simulación
im=(0,1)
a=0.01! parámetro de red en metros
c=299792458.! velocidad de la luz en metros/segundo
e0=8.854187*10**(-7)
e1=1.
e2=8.9
wp=2.*pi*c/a! Frecuencia de plasma eléctrico

```

```
t0=20
```

```
spread=6.
```

PT=int(t0+sqrt(2.*spread*log(10.**(10))))!tiempo máximo de duración del pulso para cambiar las condiciones iniciales

```

allocate (ep(ng+1,ng+1))
allocate (mu(ng+1,ng+1))
allocate (Dz(ng+1,ng+1))
allocate (Ez(ng+1,ng+1))
allocate (Hy(ng+1,ng+1))
allocate (By(ng+1,ng+1))
allocate (Hx(ng+1,ng+1))
allocate (Bx(ng+1,ng+1))

```

```
a1(1)=a
```

```
a1(2)=0.
```

```
a2(1)=0.
```

```
a2(2)=a
```

```
Rx=a1
```

```
Ry=a2
```

```
delx=a/ng
```

```
dely=delx
```

```
dt=1./(2.*c*sqrt(1./(delx*delx)+1./(dely*dely)))
```

```
print*,"delx=dely=",dely,"dt=",dt,"df=",1./(maxtime*dt)
```

```
maxfre=150 !unidad maxima de frecuencia en unidades de df
```

```
open(unit=1,file="fdtdTMvsT-g-x.dat",status="unknown")
```

```
do i=1,ng+1
```

```

do j=1,ng+1
  if (((i-(ng/2))**2+(j-(ng/2))**2)).le.f*ng*ng) then
    ep(i,j)=e2
    mu(i,j)=1.
  else
    ep(i,j)=e1
    mu(i,j)=1.
  end if
end do
end do
!núcleo del método FDTD
Print*, "calculo núcleo FDTD"
k(2)=0. !es el mismo para todos los puntos gamma-x
do r=0,nk

k(1)=real(r)*(pi/(a*real(nk)))

Dz(:,:)=0. !condicion inicial para los campos...
Ez(:,:)=0.
Hx(:,:)=0.
Hy(:,:)=0.
Bx(:,:)=0.
By(:,:)=0.
do n=0,maxtime

do i=2,ng+1
  do j=2,ng+1
    Dz(i,j)=Dz(i,j)+c*dt*(((Hy(i,j)-Hy(i-1,j))/delx)-((Hx(i,j)-Hx(i,j-
&1))/dely))
  end do
end do

do i=2,ng+1
  Dz(i,1)=Dz(i,1)+c*dt*(((Hy(i,1)-Hy(i-1,1))/delx)-((Hx(i,1)-exp(-im
&*dot_product(k,Ry))*Hx(i,ng+1))/dely))
end do

do j=2,ng+1
  Dz(1,j)=Dz(1,j)+c*dt*(((Hy(1,j)-exp(-im*dot_product(k,Rx))*Hy(ng+1
&,j))/delx)-((Hx(1,j)-Hx(1,j-1))/dely))
end do

Dz(1,1)=Dz(1,1)+c*dt*(((Hy(1,1)-exp(-im*dot_product(k,Rx))*Hy(ng+1
&,1))/delx)-((Hx(1,1)-exp(-im*dot_product(k,Ry))*Hx(1,ng+1))/dely))

Ez(:,:)=Dz(:,:)/ep(:,:) !equivalencia entre los campos E y D

```

```

do i=1,ng+1
  do j=1,ng
    Bx(i,j)=Bx(i,j)-c*dt*(Ez(i,j+1)-Ez(i,j))/dely
  end do
end do
do i=1,ng+1
  Bx(i,ng+1)=Bx(i,ng+1)-c*dt*(exp(im*dot_product(k,Ry))*Ez(i,1)-Ez(i
&,ng+1))/dely
end do

do i=1,ng
  do j=1,ng+1
    By(i,j)=By(i,j)+c*dt*(Ez(i+1,j)-Ez(i,j))/delx
  end do
end do
do j=1,ng+1
  By(ng+1,j)=By(ng+1,j)+c*dt*(exp(im*dot_product(k,Rx))*Ez(1,j)-Ez(n
&g+1,j))/delx
end do

Hx(:,:)=Bx(:,:)/mu(:,:) !equivalencia entre los campos B y H
Hy(:,:)=By(:,:)/mu(:,:)

if (n.le.PT) then
  pul=exp(-0.5*(real(t0-n)*real(t0-n))/spread)
  !pul=Am*exp(im*2.*pi*fr*dt*n)*exp(-((n*dt-t0)/tao)*((n*dt-t0)/tao))
  Dz(ng/2,ng/2)=pul
  Ez(ng/2,ng/2)=Dz(ng/2,ng/2)/ep(ng/2,ng/2)
end if
i=ng/2+10 !lugar donde se hace el análisis de la transformada de Fourier
j=i
write(1,*)r,n,Dz(i,j)!con esta orden se guardan los datos del punto donde se hace el análisis
!print*,n,Dz(i,j)!con esta orden se puede ver como se transmite la información
end do
end do

close(unit=1)

deallocate(Dz,Ez,Hx,Bx,Hy,By)

print*, "FIN DEL PROGRAMA"
end program

real function s(i,j) result(r)
use varprogram
implicit none

```

```

integer, intent(in)::i,j
!real::r
if (((i-(ng/2))**2+(j-(ng/2))**2)).le.f*ng*pi then
  r=1.
else
  r=0.
end if
end function s

```

Algoritmo FDTD para el modo TE al considerar dispersión tipo Drude

En este algoritmo se aplica la evolución temporal para el caso de una grilla cuadrada, de barras paralelas de sección transversal circular de un material metálico (dispersión tipo Drude) en aire, considerando el modo te. Se analiza solo un punto de la grilla para realizar la transformada de Fourier y por ende el diagrama espectral. Incluye el procedimiento para calcular el diagrama espectral.

```

module varprogram
  real,parameter :: pi=3.1416
  real,parameter:: f=0.001 !factor de ocupación
  integer,parameter::ng=60!tamaño de la grilla en x(y), será número par, c=2.9979*(10.**8)
end module varprogram
program ftdtTE
  use varprogram
  implicit none
  complex, dimension (:,:) , allocatable::Hz,Bz,Dy,Ey,Dx,Ex,px,Ilx
  complex,dimension(:,:),allocatable::py,Ily
  real,dimension(:,:),allocatable::mu
  real,dimension(:,:),allocatable::HzvsW
  real,dimension(2)::k,a1,a2,Rx,Ry
  complex::campo,suma,im! número imaginario
  integer :: maxtime,n,i,j,maxfre,fre,tiempo,io,PT,nk,r
  real::tao,Am,fr,pul,t0,e0,e1,e2,dt,dex,dely,c,a,spread,wp,s
  real::imp
  maxtime=20000!tiempo de la simulación
  maxfre=200! unidad maxima de frecuencia en unidades de df
  nk=15!número de particiones de los vectores k
  im=(0,1)
  a=0.01! parámetro de red en metros
  c=299792458.!velocidad de la luz en metros/segundo
  e0=8.854187*10**(-7)
  e1=1.
  e2=1.
  wp=2.*pi*c/a! Frecuencia de plasma eléctrico
  !t0=0.5*10.**(-12)
  !fr=10.**14

```

```

!tao=1./(8.*f)
!Am=1.
t0=20
spread=6.
PT=int(t0+sqrt(2.*spread*log(10.**t0)))!tiempo máximo de duración del pulso para cambiar las
condiciones iniciales
allocate (px(ng+1,ng+1))
allocate (py(ng+1,ng+1))
allocate (I1x(ng+1,ng+1))
allocate (I1y(ng+1,ng+1))
allocate (mu(ng+1,ng+1))
allocate (Hz(ng+1,ng+1))
allocate (Bz(ng+1,ng+1))
allocate (Dy(ng+1,ng+1))
allocate (Ey(ng+1,ng+1))
allocate (Dx(ng+1,ng+1))
allocate (Ex(ng+1,ng+1))
a1(1)=a
a1(2)=0.
a2(1)=0.
a2(2)=a
Rx=a1
Ry=a2
delx=a/ng
dely=delx
dt=1./(2.*c*sqrt(1./(delx*delx)+1./(dely*dely)))
print*, "delx=dely=", dely, "dt=", dt, "df=", 1./(maxtime*dt), "wp=", wp, "
&fp=wp/(2*pi)=", wp/(2*pi), "rp=fp/df=", wp/(2*pi)/(1./(maxtime*dt))

open(unit=1, file="fdtdTEvsT-g-x.dat", status="unknown")
mu(:,:)=1. !valor permeabilidad magnética, material no magnético

!núcleo del método FDTD
Print*, "calculo núcleo FDTD"
k(2)=0. !es el mismo para todos los puntos gamma-x
do r=0, nk
    k(1)=real(r)*(pi/(a*real(nk)))

    Hz(:,:)=0. !condicion inicial para los campos...
    Bz(:,:)=0.
    Dx(:,:)=0.
    Dy(:,:)=0.
    Ex(:,:)=0.
    Ey(:,:)=0.
    !campos auxiliares para añadir la dispersión tipo Drude, s es una función de s:s(i,j)
    px(:,:)=0.

```

```

I1x(:,:)=0.
py(:,:)=0.
I1y(:,:)=0.
do n=0,maxtime

do i=1,ng+1
  do j=2,ng+1
    Dx(i,j)=Dx(i,j)+c*dt*(hz(i,j)-hz(i,j-1))/dely
  end do
end do
do i=1,ng+1
  Dx(i,1)=Dx(i,1)+c*dt*(hz(i,1)-exp(-im*dot_product(k,Ry))*hz(i,ng+1
&))/dely
end do

do i=2,ng+1
  do j=1,ng+1
    Dy(i,j)=Dy(i,j)-c*dt*(hz(i,j)-hz(i-1,j))/delx
  end do
end do
do j=1,ng+1
  Dy(1,j)=Dy(1,j)-c*dt*(hz(1,j)-exp(-im*dot_product(k,Rx))*Hz(ng+1,j
&))/delx
end do

!núcleo de transformación para la dispersión tipo drude
Do i=1,ng+1
do j=1,ng+1
Ex(i,j)=(Dx(i,j)-s(i,j)*e2*wp*wp*(n*dt*I1x(i,j)-dt*dt*px(i,j)))/(e
&1*(1.-s(i,j))+s(i,j)*e2) !equivalencia entre los campos E y D
Ey(i,j)=(Dy(i,j)-s(i,j)*e2*wp*wp*(n*dt*I1y(i,j)-dt*dt*py(i,j)))/(e
&1*(1.-s(i,j))+s(i,j)*e2)
I1x(i,j)=Ex(i,j)*dt+I1x(i,j)
I1y(i,j)=Ey(i,j)*dt+I1y(i,j)
px(i,j)=n*Ex(i,j)+px(i,j)
py(i,j)=n*Ey(i,j)+py(i,j)
end do
end do

do i=1,ng
  do j=1,ng
    Bz(i,j)=Bz(i,j)-c*dt*(((Ey(i+1,j)-Ey(i,j))/delx)-((Ex(i,j+1)-Ex(i,
&j))/dely))
  end do
end do
do j=1,ng

```

```

Bz(ng+1,j)=Bz(ng+1,j)-c*dt*(((exp(im*dt*product(k,Rx))*Ey(1,j)-Ey
&(ng+1,j))/delx)-((Ex(ng+1,j+1)-Ex(ng+1,j))/dely))
end do

```

```

do i=1,ng
Bz(i,ng+1)=Bz(i,ng+1)-c*dt*(((Ey(i+1,ng+1)-Ey(i,ng+1))/delx)-((exp
&(im*dt*product(k,Ry))*Ex(i,1)-Ex(i,ng+1))/dely))
end do

```

```

Bz(ng+1,ng+1)=Bz(ng+1,ng+1)-c*dt*(((exp(im*dt*product(k,Rx))*Ey(1
&,ng+1)-Ey(ng+1,ng+1))/delx)-((exp(im*dt*product(k,Ry))*Ex(ng+1,1)
&-Ex(ng+1,ng+1))/dely))

```

```

Hz(:,:)=Bz(:,:)/mu(:,:) !equivalencia entre los campos B y H

```

```

if (n.le.PT) then
pul=exp(-0.5*(real(t0-n)*real(t0-n))/spread)
!pul=Am*exp(im*2.*pi*fr*dt*n)*exp(-(n*dt-t0)/tao)*((n*dt-t0)/tao))
Bz(ng/2,ng/2)=pul
Hz(ng/2,ng/2)=Bz(ng/2,ng/2)/mu(ng/2,ng/2)
end if

```

```

i=8 !lugar donde hacemos el análisis de la transformada de Fourier
j=i

```

```

write(1,*)r,n,HZ(i,j)!con esta orden se guardan los datos del punto donde se hace el análisis

```

```

!print*,n,Dz(i,j)!con esta orden se puede ver como se transmite la información al lugar donde se
decida hacer el analisis

```

```

end do

```

```

end do

```

```

close(unit=1)

```

```

deallocate(Hz,Bz,Dx,Ex,Dy,Ey,px,l1x,py,l1y)

```

```

print*,"calculo DFT para los datos del FDTD"

```

```

open(unit=2,file="fdtdTEvsT-g-x.dat",status="old")

```

```

open(unit=3,file="fdtdTEvsW-g-x.dat",status="unknown")

```

```

do fre=0,maxfre

```

```

!transformada discreta de Fourier, DFT, para los valores de Dz

```

```

do r=0,nk

```

```

suma=0.

```

```

do n=0,maxtime

```

```

read(2,*)i,tiempo,campo

```

```

    suma=suma+campo*exp(-im*2.*pi*tiempo*fre/maxtime)
end do
write(3,*)i,fre,real(suma*conjg(suma))
end do
rewind(2)!sirve para resetear el file, y volver a utilizarlo
!print*,fre,real(suma*conjg(suma))!para ver la transformada de Fourier

end do

close(unit=2)
close(unit=3)
!parte del programa para calcular los picos del espectro de frecuencias
print*, "calculo de las resonancias"

allocate (HzvsW(maxfre+1,nk+1))!arreglo para guardar el espectro de frecuencias
open(unit=3,file="fdtdTEvsW-g-x.dat",status="old")
open(unit=4,file="ResonanciasTE-g-x.dat",status="unknown")
do fre=0,maxfre
do i=0,nk
    read(3,*)j,r,HZvsW(fre+1,i+1)
end do
end do

do i=1,nk+1
do j=2,maxfre
    if ((HzvsW(j-1,i).lt.HzvsW(j,i)).and. (HzvsW(j+1,i).lt.HzvsW(j,i
&)))then
        write(4,*)i-1,j-1,real(j-1)/(maxtime*dt)/(10.**9)!valor de la frecuencia en Ghz
    end if
end do
if (HzvsW(1,i).gt.HzvsW(2,i)) then
    write(4,*)i-1,0,0.
end if
if (HzvsW(maxfre+1,i).gt.HzvsW(maxfre,i)) then
    write(4,*)i-1,maxfre,real(maxfre)/(maxtime*dt)/(10.**9)
end if

end do
close(unit=4)
close(unit=3)
print*, "FIN DEL PROGRAMA"
end program

real function s(i,j) result(r)
use varprogram
implicit none
integer, intent(in)::i,j

```

```

real::r
if (((i-(ng/2))**2+(j-(ng/2))**2)).le.f*ng*ng/pi then
  r=1.
else
  r=0.
end if
end function s

```

Algoritmo FDTD para el modo TE al considerar la dispersión tipo Drude y múltiples puntos

En este algoritmo se aplica la evolución temporal para el caso de una grilla cuadrada, de barras paralelas de sección transversal circular de un material metálico (dispersión tipo Drude) en aire, considerando el modo TE. Se analizan varios puntos de la grilla. Se usa junto con el algoritmo DFT mostrado más adelante para el análisis espectral.

```

module varprogram
  real,parameter :: pi=3.1416
  real,parameter:: f=0.125 !factor de ocupación
  real,parameter:: a=1!parametro de red en metros
  real,parameter:: c=299792458.!velocidad de la luz en metros/segundo
  real,parameter:: wp=2*pi*c/a! Frecuencia de plasma eléctrico
  real,parameter:: e0=8.854187*10**(-7)
  real,parameter:: e1=1. ! material que rodea las varillas
  real,parameter:: e2=8.9!material varillas
  integer,parameter::ng=60!tamaño de la grilla en x(y), será número par, c=2.9979*(10.**8)
  integer,parameter:: maxtime=20000!tiempo de la simulación
  integer,parameter:: nk=15!número de particiones de los vectores k
  integer,parameter:: np=4 !número de puntos para realizar el análisis
  complex,parameter:: im=(0,1)
end module varprogram
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
program ftdTE
use varprogram
implicit none
logical::bandera
complex, dimension (:,:) , allocatable::Hz,Bz,Dy,Ey,Dx,Ex,px,I1x
complex,dimension(:,:),allocatable::py,I1y
real,dimension(:,:),allocatable::mu
real,dimension(2)::k,a1,a2,Rx,Ry
complex::campo!numero imaginario
integer :: n,i,j,PT,r
real::tao,Am,fr,pul,t0,dt,dex,dely,spread,s
real::imp
!t0=0.5*10.**(-12)
!fr=10.**14

```

```

!tao=1./(8.*f)
!Am=1.
t0=20
spread=6.
PT=int(t0+sqrt(2.*spread*log(10.**t0)))!tiempo máximo de duración del pulso para cambiar las
condiciones iniciales
allocate (px(ng+1,ng+1))
allocate (py(ng+1,ng+1))
allocate (I1x(ng+1,ng+1))
allocate (I1y(ng+1,ng+1))
allocate (mu(ng+1,ng+1))
allocate (Hz(ng+1,ng+1))
allocate (Bz(ng+1,ng+1))
allocate (Dy(ng+1,ng+1))
allocate (Ey(ng+1,ng+1))
allocate (Dx(ng+1,ng+1))
allocate (Ex(ng+1,ng+1))
a1(1)=a
a1(2)=0.
a2(1)=0.
a2(2)=a
Rx=a1
Ry=a2
delx=a/ng
dely=delx
dt=1./(2.*c*sqrt(1./(delx*delx)+1./(dely*dely)))
print*, "delx=dely=", dely, "dt=", dt, "df=", 1./(maxtime*dt), "wp=", wp, "
&fp=wp/(2*pi)=", wp/(2*pi), "rp=fp/df=", wp/(2*pi)/(1./(maxtime*dt)), "
&valor de fn para normalizar las graficas=", c/a/(1./(maxtime*dt))

open(unit=1, file="fdtdTEvsT-g-x.dat", status="unknown")
mu(:, :)=1.

! núcleo del método fdtd,
Print*, "calculo nucleo FDTD"
k(2)=0.!es el mismo para todos los puntos gamma-x
do r=0,nk

k(1)=real(r)*(pi/(a*real(nk)))

Hz(:, :)=0.!condicion inicial para los campos...
Bz(:, :)=0.
Dx(:, :)=0.
Dy(:, :)=0.
Ex(:, :)=0.
Ey(:, :)=0.
!campos auxiliares para añadir la dispersión tipo Drude

```

```

px(:,:)=0.
I1x(:,:)=0.
py(:,:)=0.
I1y(:,:)=0.
do n=0,maxtime

do i=1,ng+1
  do j=2,ng+1
    Dx(i,j)=Dx(i,j)+c*dt*(hz(i,j)-hz(i,j-1))/dely
  end do
end do
do i=1,ng+1
  Dx(i,1)=Dx(i,1)+c*dt*(hz(i,1)-exp(-im*dot_product(k,Ry))*hz(i,ng+1
&))/dely
end do

do i=2,ng+1
  do j=1,ng+1
    Dy(i,j)=Dy(i,j)-c*dt*(hz(i,j)-hz(i-1,j))/delx
  end do
end do
do j=1,ng+1
  Dy(1,j)=Dy(1,j)-c*dt*(hz(1,j)-exp(-im*dot_product(k,Rx))*Hz(ng+1,j
&))/delx
end do

!núcleo de transformación para la dispersión tipo drude
Do i=1,ng+1
do j=1,ng+1
Ex(i,j)=(Dx(i,j)-s(i,j)*e2*wp*wp*(n*dt*I1x(i,j)-dt*dt*px(i,j)))/(e
&1*(1-s(i,j))+s(i,j)*e2) !equivalencia entre los campos E y D
Ey(i,j)=(Dy(i,j)-s(i,j)*e2*wp*wp*(n*dt*I1y(i,j)-dt*dt*py(i,j)))/(e
&1*(1-s(i,j))+s(i,j)*e2)
I1x(i,j)=Ex(i,j)*dt+I1x(i,j)
I1y(i,j)=Ey(i,j)*dt+I1y(i,j)
px(i,j)=n*Ex(i,j)+px(i,j)
py(i,j)=n*Ey(i,j)+py(i,j)
end do
end do

do i=1,ng
  do j=1,ng
    Bz(i,j)=Bz(i,j)-c*dt*(((Ey(i+1,j)-Ey(i,j))/delx)-((Ex(i,j+1)-Ex(i,
&j))/dely))
  end do
end do

```

```

do j=1,ng
  Bz(ng+1,j)=Bz(ng+1,j)-c*dt*(((exp(im*dt_product(k,Rx))*Ey(1,j)-Ey
&(ng+1,j))/delx)-((Ex(ng+1,j+1)-Ex(ng+1,j))/dely))
end do

do i=1,ng
  Bz(i,ng+1)=Bz(i,ng+1)-c*dt*(((Ey(i+1,ng+1)-Ey(i,ng+1))/delx)-((exp
&(im*dt_product(k,Ry))*Ex(i,1)-Ex(i,ng+1))/dely))
end do

Bz(ng+1,ng+1)=Bz(ng+1,ng+1)-c*dt*(((exp(im*dt_product(k,Rx))*Ey(1
&,ng+1)-Ey(ng+1,ng+1))/delx)-((exp(im*dt_product(k,Ry))*Ex(ng+1,1
&-Ex(ng+1,ng+1))/dely))

```

Hz(:,:)=Bz(:,:)/mu(:,:) !equivalencia entre los campos B y H

```

if (n.le.PT) then
  pul=exp(-0.5*(real(t0-n)*real(t0-n))/spread)
  !pul=Am*exp(im*2.*pi*fr*dt*n)*exp(-((n*dt-t0)/tao)*((n*dt-t0)/tao))

```

```

Bz(ng/2,ng/2-20)=pul
Hz(ng/2,ng/2-20)=Bz(ng/2,ng/2-20)/mu(ng/2,ng/2-20)
end if

```

! lugares donde se hace el análisis de la transformada de Fourier

```

do j=1,np
  write(1,*)r,n,j,HZ(2,2+10*j) !con esta orden se guardan los datos de los puntos donde se hace el
analisis
end do

end do
end do

close(unit=1)

deallocate(Hz,Bz,Dx,Ex,Dy,Ey,px,I1x,py,I1y)

print*, "FIN DEL PROGRAMA"
end program

real function s(i,j) result(r)
use varprogram
implicit none

```

```

real::r
if (((i-(ng/2))**2+(j-(ng/2))**2)).le.f*ng*ng/pi then
  r=1.
else
  r=0.
end if
end function s

```

Algoritmo DFT para comparar varios puntos

Algoritmo para el cálculo del diagrama espectral para varios puntos de la grilla. Se comparan los diferentes máximos de los puntos estudiados, para hallar los modos de propagación propios del sistema. Este programa se aplica a los puntos escogidos para la polarización TE, basados en la evolución temporal del algoritmo FDTD.

Se puede cambiar el nombre del archivo para considerar los puntos TM.

```

program DFT
use varprogram
implicit none
logical::bandera
complex,dimension(:),allocatable::suma
real,dimension(:,:,:),allocatable::CampovsW,Arreglo
complex::campo!numero imaginario
real::aux,delx,dely,dt,fre0,fref!variable auxiliar para correcta lectura de archivo
integer :: n,i,j,maxfre,fre,r,tiempo,p,io
maxfre=160 !unidad maxima de frecuencia en unidades de df
fre0=0
fref=80

delx=a/ng
dely=delx
dt=1./(2.*c*sqrt(1./(delx*delx)+1./(dely*dely)))
print*,"delx=dely=",dely,"dt=",dt,"df=",1./(maxtime*dt),"wp=",wp,"
&fp=wp/(2*pi)=",wp/(2*pi),"rp=fp/df=",wp/(2*pi)/(1./(maxtime*dt)),"
&valor de fn para normalizar las graficas=",c/a/(1./(maxtime*dt))
print*,
print*,"calculo DFT para los datos del FDTD"

open(unit=2,file="fdtdTEvsT-g-x.dat",status="old")
open(unit=3,file="fdtdTEvsW-g-x.dat",status="unknown")
allocate(suma(np))
do fre=0,maxfre

!transformada discreta de Fourier, DFT, para los valores de Dz

```

```

do r=0,nk
  suma(:)=0.
do n=0,maxtime
  do j=1,np
    read(2,*)i,tiempo,p,campo
    suma(p)=suma(p)+campo*exp(-im*2.*pi*(fre0+(fref-fre0)*real(fre)/re
&al(maxfre))*REAL(tiempo)/real(maxtime))
  end do
end do
do j=1,np
write(3,*)i,real(fre0+(fref-fre0)*real(fre)/real(maxfre)),j,real(s
&uma(j)*conjg(suma(j)))
end do
end do
rewind(2)!sirve para resetear el file, y volver a utilizarlo
!print*,fre,real(suma*conjg(suma))!para ver la transformada de Fourier

end do
deallocate(suma)
close(unit=2)
close(unit=3)
! parte del programa para calcular los picos del espectro de frecuencias
print*,"cálculo de las resonancias"

allocate (CampovsW(maxfre+1,nk+1,np))!arreglo para guardar el espectro de frecuencias
open(unit=3,file="fdtdTEvsW-g-x.dat",status="old")
open(unit=4,file="Res.puntos.TE-g-x.dat",status="unknown")
do fre=0,maxfre
do i=0,nk
  do p=1,np
    read(3,*)j,aux,n,CampovsW(fre+1,i+1,p)
  end do
end do
end do
do p=1,np
do i=1,nk+1
do j=2,maxfre
if (((CampovsW(j-1,i,p)).lt.CampovsW(j,i,p)).and. (CampovsW(j+1,i,p)
&.lt.CampovsW(j,i,p)))then
  write(4,*)p,i-1,j-1,real(fre0+(fref-fre0)*real(j-1)/real(maxfre))
  end if
end do
end do
end do
deallocate(CampovsW)

close(unit=4)

```

```

close(unit=3)

open(unit=4,file="Res.puntos.TE-g-x.dat",status="old")
open(unit=5,file="ResonanciasTE-g-x.dat",status="unknown")
print*,"comparacion de los datos obtenidos"
allocate(Arreglo(nk+1,np,maxfre+1))
Arreglo(:,,:)=0.
io=0
do while(io>=0)
  read(4,*,iostat=io)p,i,j,aux
  if(io==0)then
    arreglo(i+1,p,j+1)=1.
  end if
end do

do i=1,nk+1
  do j=1,maxfre+1
    bandera=.true.
    do p=1,np
      if (Arreglo(i,p,j)==0)then
        bandera=.false.
        p=np+1
      end if
    end do
    if (bandera.eqv..true.) then
      write(5,*)i-1,real(fre0+(fref-fre0)*real(j-1)/real(maxfre))
    end if
  end do
end do

close(unit=4)
close(unit=5)

print*,"FIN DEL PROGRAMA"
end program

```

REFERENCIAS

- Ashutosh, S. y Pradip, J. (2012). FDTD Analysis of the Dispersion Characteristics of the Metal PBG Structures. *Progress In Electromagnetics Research B*, 39, 71-88. DOI 10.2528/PIERB11120601
- Barut, A. O. (1980). *Electrodynamics and classical theory of fields and particles*. Dover Publications, Inc.
- Chouhan, N., Ameta, R., Meena, R., Mandawat, N. y Ghildiyal, R. (2016). Visible Light Harvesting Pt/CdS/Co-doped ZnO Nanorods Molecular Device for Hydrogen Generation. *International Journal of Hydrogen Energy*, 41, 2298-2306. <https://doi.org/10.1016/j.ijhydene.2015.11.019>
- Engheta, N., y Ziolkowski R. W. (Ed.) (2006). *Metamaterials Physics and Engineering Explorations*. John Wiley & Sons, Inc.
- He, J. y Cada, M. (1992). Combined Distributed Feedback and Fabry-Perot Structures with a PhaseMatching Layer for Optical Bistable Devices. *Applied Physics Letters*, 61, 2150. <https://doi.org/10.1063/1.108303>
- Huang, K., Bienstman, P., Joannopoulos, J., Nelson, K. y Fan, S. (2003). Phonon-polariton excitations in photonic crystals. *Physical Review B*, 68, 075209. <https://doi.org/10.1103/PhysRevB.68.075209>
- Jackson, J. D. (1998). *Classical Electrodynamics*. John Wiley & Sons, Inc.
- Joannopoulos, J., Johnson, S., Meade, R. y Winn J. (2008). *Photonic Crystals: Molding the Flow of Light*. Princeton University Press.
- Kittel, C. (1966). *Introduction to Solid States Physics*. Wiley.
- Kumar, A., Kumar V., Pandey, G., Sing, Kh. y Ojha S. (2013). Formation of gap soliton in negative Kerr nonlinear one-dimensional photonic crystal. *Optoelectronics and Advanced Materials Rapid Communications*, 7, 477-480. <https://oam-rc.inoe.ro/articles/formation-of-gapsoliton-in-negative-kerr-nonlinear-one-dimensional-photonic-crystal/>
- Kuzmiak, V., Maradudin, A. y McGurn A. (1994). Photonic band structures of two-dimensional systems containing metallic components. *Physical Review B*, 50, 16835. <https://doi.org/10.1103/PhysRevB.50.16835>
- Li, M., He, F., Liao, Q., Liu, J., Xu, L., Jiang, L., Song, Y. y Zhu, D.B. (2008). Ultrasensitive DNA Detection Using Photonic Crystals. *Angewandte Chemie*, 120, 7368-7372. <https://doi.org/10.1002/anie.200801998>
- Moreno, E., Erni, D. y Hafner, C (2002). Band structure computations of metallic photonic crystals with multiple method. *Physical Review B*, 65, 155120. <https://doi.org/10.1103/PhysRevB.65.155120>

- Raman, A. y Fan, S. (2010). Photonic Band Structure of Dispersive Metamaterials Formulated as a Hermitian Eigenvalue Problem. *Physical Review Letters*, 104, 087-401. <https://doi.org/10.1103/PhysRevLett.104.087401>
- Reyes, E., Mogilevsev, D., Cavalcanti, S., De Carvalho, C. y Oliveira, L. (2009). Plasmon polaritons in photonic superlattices containing. *Europhysics Letters*, 88, 24002. <https://doi.org/10.1209/0295-5075/88/24002>
- Sakoda, K., Chutinan, A., Noda, S., Mitsuyu, T. y Hirao, K. (2001). Photonic bands of metallic systems. I. Principle of calculation and accuracy. *Physical Review B*, 64, 045116. <https://doi.org/10.1103/PhysRevB.64.045116>
- Soto-Puebla, D., Xiao, M. y Ramos F. (2004). Optical properties of a dielectric-metallic superlattice: the complex photonic bands. *Physics Letters A*, 326, 273-280. <https://doi.org/10.1016/j.physleta.2004.03.070>
- Su, L., Krishna, S., Luo, J., Liu, J., Liu, X., Guo, J., Sum, T., Deng, R., Fan, H., Liu, X. y Tok, A. (2013). Photon Upconversion in Hetero-nanostructured Photoanodes for Enhanced Near-Infrared Light Harvesting. *Advanced Materials*, 25, 1603-1607. <https://doi.org/10.1002/adma.201204353>
- Sullivan, D. M. (2000). *Electromagnetic simulation using the FDTD method*. IEEE Press series on RF and microwave technology.
- Xiao, S. y Sailing, H. (2002). FDTD method for computing the off-plane band structure in a two-dimensional photonic crystal consisting of nearly free-electron metals. *Physica B: Condensed Matter*, 324, 403-408. [https://doi.org/10.1016/S0921-4526\(02\)01460-6](https://doi.org/10.1016/S0921-4526(02)01460-6)
- Yao, K., Xin, X.-K., Chueh, C.-C., Chen, K.-S., Xu, Y.-X. y Jen, A.K.-Y. (2015), Enhanced Light Harvesting by Integrating Synergetic Microcavity and Plasmonic Effects for High-Performance ITO-Free Flexible Polymer Solar Cells. *Advanced Functional Materials*, 25: 567-574. <https://doi.org/10.1002/adfm.201403297>
- Yee, K. (1966). Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media. *IEEE Trans. Antennas and Propagation*, 14, 302-307. <https://doi.org/10.1109/TAP.1966.1138693>
- Zohrabi, R. y Namdar, A. (2017). Perfect Tunable All-Optical Diode based on Periodic Photonic Crystal Grand Graded Structures. *Journal of Optical Communications*, 40, 187-193. <https://doi.org/10.1515/joc-2017-0080>



Semillero de investigación

EDCiBa

Semillero EDCiBa

Esta guía tiene el propósito de introducir a los estudiantes al estudio de las simulaciones numéricas, por medio de la construcción de algoritmos basados en el método de diferencias finitas, en el cual las funciones de interés evolucionan tanto en el tiempo como en el espacio. Se aplica el formalismo desarrollado a un problema específico de propagación de las ondas electromagnéticas en un cristal fotónico 2D, formado por barras paralelas de sección transversal circular dispuestas en un arreglo rectangular, contenidas en un material con diferente respuesta óptica. El texto describe cómo las propiedades ópticas del cristal fotónico se pueden estudiar al tener en cuenta dos polarizaciones principales, denominadas transversal eléctrica (te) y transversal magnética (tm), desarrolla el formalismo de las ecuaciones del electromagnetismo de Maxwell y presenta su discretización y aplicación al mundo computacional. Además, se describe el análisis espectral de las fluctuaciones de los campos electromagnéticos, para hallar los modos de propagación permitidos en el cristal.

