

Proposal 4: Simulate a plain wave using FDFD

- 1) Design And Optimization Of Nano-optical Elements By Coupling Fabrication To Optical Behavior. Capítulo 3
- 2) Veronis G. (2016) Finite-Difference Frequency-Domain Technique. In: Bhushan B. (eds) Encyclopedia of Nanotechnology. Springer, Dordrecht. https://doi.org/10.1007/978-94-017-9780-1_16

Para más información mirar repositorio en <https://github.com/JuanjoNeuro/EntregaProyecto>

Métodos Computacionales en Física No Lineal - Máster FisyMat

Isaac José Sánchez Díaz y Juan José Jiménez Fernández - Abril 2021



Algoritmo

```
%Valibles físicas del modelo (Uso U.N.)
e0const=1/4*pi;
mu0const=4*pi;
%Sigma=58 eV es la conductividad del hierro
sigmaEconst=0;

%Relación de optimizacion de PAPER PRINCIPAL
sigmaHconst=sigmaEconst*(e0const/mu0const);
%sigmaHconst=0;

%Frecuencia en estas unidades
%(3eV es la frecuencia del rojo)
lambda=2.5;
k=lambda/(2*pi);

%Subdivisión espacial
increX=lambda/20;

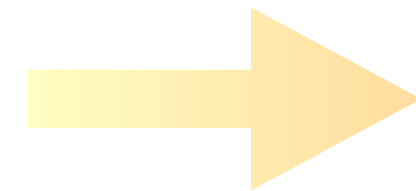
%Frecuencia según la relación de dispersión
w=(2/increX)*sin(k*increX/2);

%Defino el número de subdivisiones espaciales
dimeX=1000;

%% Inicializo los campos que voy a evolucionar
%Defino el campo electrico
E=zeros(1,dimeX)';
H=zeros(1,dimeX)';
%Defino las corrientes

%Fuentes reales
%J=ones(1,dimeX)';
J=gaussiana(1,0.5,dimeX,0.01)';
%J=barrera(dimeX,round(0.30*dimeX),round(0.70*dimeX))';

%Defino en cada punto las constantes
e0=e0const*ones(1,dimeX)';
mu0=mu0const*ones(1,dimeX)';
sigmaE=sigmaEconst*ones(1,dimeX)';
sigmaH=sigmaHconst*ones(1,dimeX)';
```



```
%Defino la matriz que inicializa el sistema
M=zeros(dimeX);

%Según las definiciones la subdivisiones nos lleva a que:
for i=1:dimeX
    for j=1:dimeX
        if i==j
            M(i,j)=-2/(increX)^2 + w^2*e0(i)*mu0(i) - ...
                sqrt(-1)*w*mu0(i)*sigmaE(i);
        elseif ((j==i-1)||j==i+1))
            M(i,j)=(increX)^(-2);
        end
    end
end

%Diagonalizo y obtengo los modos
[vectores,autovalores]=eig(M);

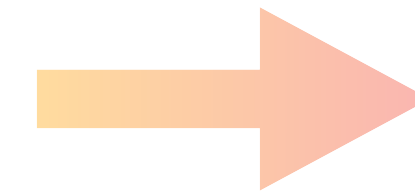
%Campo eléctrico obtenido
E=M^(-1)*sqrt(-1)*w*(J.*mu0);

%Defino el campo con condiciones PEC
CampoE=[E;0];

%Creo el magnético
for i=1:dimeX
    H(i)=CampoE(i+1)-CampoE(i);
    H(i)=H(i)/increX;
end

%Defino el campo magnético
H=H./(sqrt(-1)*w*mu0+sigmaH);

%Defino el número de subdivisiones espaciales
Espacio=1:dimeX;
```



```
%Ploteo de los campos
figure(1)
subplot(2,2,1)
plot(Espacio,real(E),'.-b')
hold on
title('Campo Eléctrico Real')
xlim([min(Espacio) max(Espacio)])
hold off

subplot(2,2,2)
plot(Espacio,imag(E),'.-b')
hold on
title('Campo Eléctrico Imaginario')
xlim([min(Espacio) max(Espacio)])
hold off

subplot(2,2,3)
plot(Espacio,real(H),'.-r')
hold on
title('Campo Magnético Real')
xlim([min(Espacio) max(Espacio)])
hold off

subplot(2,2,4)
plot(Espacio,imag(H),'.-r')
hold on
title('Campo Magnético Imaginario')
xlim([min(Espacio) max(Espacio)])
hold off

%Ploteo de los campos
figure(2)
plot(Espacio,J,'.-k')
hold on
title('Fuente de corriente')
xlim([min(Espacio) max(Espacio)])
hold off
```

Definimos las constantes que va a usar nuestro modelo.

Se modelizan las fuentes de corriente y el valor de las variables en cada punto

Utilizamos la matriz sustrato del sistema
updateamos los campos eléctrico y magnético a cada punto

Ploteamos los resultados en gráficas en formato imagen y animaciones de vídeo con la evolución de los campos usando distintos parámetros

Resultados

