mt360_ur7_8_1_1 Wave Reflection at Normal Incidence

Table of Contents

Title

```
Authors: Ulaby and Ravaioli
Book Edition: 7th
Problem #: Based On 8.3
Last Updated: 3/16/2016
```

Description

You have an electromagnetic wave propagating in the +z direction in medium 1 that comes into contact with another medium, medium 2. (See Figure 8-3 in your text books).

The properties of medium 1 are # It has an electric permittivity of 4. # It is made out of non-conducting material. # The relative magnetic permeability is 1.

The properties of medium 2 (except for the relative magnetic permeability which is 1) can be adjusted using the interactive plot. In addition, the frequency of the electromagnetic wave can be adjusted, and the time can be adjusted to create a phase up to 360 degrees.

The purpose of this assignment is to visually see how the electric permittivity and conductivity of medium two, and the frequency of the signal affect the wave reflected and transmitted at the boundary.

Exercise

1. Complete the functions getTau, and getGamma. The function getTau returns the transmission coefficient and the function getGamma returns the reflection coefficient.
2. Interact with the image as necessary/ desired in order to answer the questions below.
3. In the function 'sigmaScale' change the scale to 1, and interact with the image again in order to answer the questions below.

Questions

1. What affect does the relation between the electric permittivity of medium 1 and medium 2 have on the reflected and transmitted wave?

2. What affect does frequency of the wave have on the reflected and transmitted wave with sigma is 0 and when sigma is non-zero? why?
3. How can the magnitude of the transmitted wave be larger than the magnitude of the source wave?
4. Can the magnitude of the reflected wave be larger than the magnitude of the source wave?
5. When you change the scale to 1 in the function 'sigmaScale' what happens to the transmitted wave? Why?

Useful Information

Reflection Coefficient of a Wave at Normal Incidence to the Boundary

$$\Gamma = \frac{\eta_{c_2} - \eta_{c_1}}{\eta_{c_2} + \eta_{c_1}}$$

Transmission Coefficient of a Wave at Normal Incidence to the Boundary

$$\tau = \frac{2\eta c_2}{\eta_{c_2} + \eta_{c_1}}$$

Provided Code

```
function mt360_ur7_8_1_1_wave_reflection_at_normal_incidence()

fig = figure(1);
clf();
set(fig, 'Tag', 'fig')
var = guihandles(fig);
var.ax = axes('Parent',fig,'position',[0.1 0.5  0.8 0.4]);

var.f = 1;              % Frequency of signal in Hz
var.w = var.f*2*pi;     % Frequency of signal in rad/s

% Medium 1
var.Eo_i = 1;            % Magnitude of the incidence electric field,
 V/m
var.er_1 = 4;            % Relative electric permittivity of medium 1
var.uo_1 = 1;            % Relative magnetic permeability of medium 1
var.sigma_1 = 0;         % Conductivity of medium 1, S/m
var.epsilon_1 = var.er_1*P.Eo;% Electric permittivity of medium 1, F/m
var.mu_1 = var.uo_1*P.Uo;    % Magnetic permeability of medium 1, H/m

% Medium 2
var.er_2 = 4;            % Relative electric permittivity of medium 2
var.uo_2 = 1;            % Relative magnetic permeability of medium 2
var.mu_2 = var.uo_2*P.Uo;    % Magnetic permeability of medium 2, H/m
var.sigma_2 = 0;         % Conductivity of medium 2, S/m

var.t = 0;

% plot initial data
```

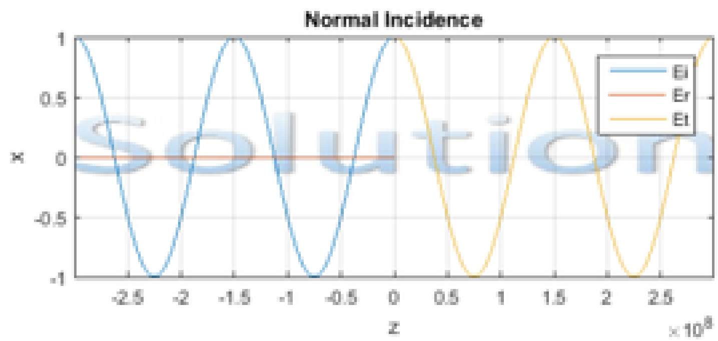
```
var.h = plot([1],[1],[1],[1],[1],[1]);

grid on
var = updatePlot(var,P);
title('Normal Incidence')
ylabel('x')
xlabel('z')
legend('Ei','Er','Et')
guidata(fig,var);

% User interactive controls
er_2_ctrl = uicontrol('Parent',fig,'Style','slider',...
        'Position',[100,20,380,20],'value',var.er_2,...
        'min',0.2,'max',80,'Callback',{@update_er_2,P},...
        'Tag','slider1');
sigma_2_ctrl = uicontrol('Parent',fig,'Style','slider',...
        'Position',[100,40,380,20],'value',var.sigma_2,...
        'min',0,'max',10,'Callback',{@update_sigma_2,P});
freq_ctrl = uicontrol('Parent',fig,'Style','slider',...
         'Position',[100,60,380,20],'value',var.f,...
         'min',1,'max',10e3,'Callback',{@update_freq,P});
t_ctrl = uicontrol('Parent',fig,'Style','slider',...
          'Position',[100,80,380,20],'value',var.t,...
          'min',0,'max',10,'Callback',{@update_t,P});

% controls to display slider names
er_2_txt = uicontrol('Parent',fig,'Style','text',...
       'Position',[0,20,100,20],'String','er_2');
sigma_2_txt = uicontrol('Parent',fig,'Style','text',...
       'Position',[0,40,100,20],'String','sigma_2 *scale, S/m');
freq_txt = uicontrol('Parent',fig,'Style','text',...
        'Position',[0,60,100,20],'String','freq, Hz');
t_txt = uicontrol('Parent',fig,'Style','text',...
        'Position',[0,80,100,20],'String','t*2pi/w, s');

% controls to display slider values
er_2_disp = uicontrol('Parent',fig,'Style','text',...
        'Position',[480,16,40,15],'String',...
        num2str(var.er_2),'Tag','er_2_disp');
sigma_2_disp = uicontrol('Parent',fig,'Style','text',...
        'Position',[480,37,40,15],'String',...
        num2str(var.sigma_2),'Tag','sigma_2_disp');
freq_disp = uicontrol('Parent',fig,'Style','text',...
         'Position',[480,59,40,15],'String',...
         num2str(var.f),'Tag','freq_disp');
t_disp = uicontrol('Parent',fig,'Style','text',...
         'Position',[480,80,40,15],'String',...
         num2str(var.t),'Tag','t_disp');
end

% Callback function when Ex_mag slider changes
function update_er_2(hObject,callbackdata,P)
 handle = findobj('Tag','fig');
 var = guidata(handle);
 var.er_2= (hObject.Value);
```

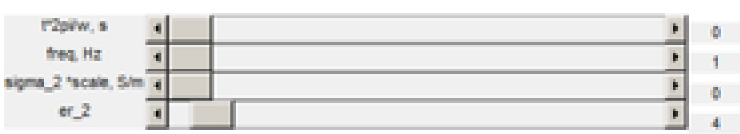
```
    var = updatePlot(var,P);
    guidata(handle, var);
    er_2_disp = findobj('Tag','er_2_disp');
    set(er_2_disp,'String',num2str(hObject.Value));
end

% Callback function when Ey_mag slider changes
function update_sigma_2(hObject,callbackdata,P)
  handle = findobj('Tag','fig');
  var = guidata(handle);
  var.sigma_2 = (hObject.Value);
  var.sigma_2 = ... % Conductivity of medium 2, S/m
      sigmaScale(var.sigma_2);
  var = updatePlot(var,P);
  guidata(handle, var);
  sigma_2_disp = findobj('Tag','sigma_2_disp');
  set(sigma_2_disp,'String',num2str(hObject.Value));
end

% Callback function when Ex_phase slider changes
function update_freq(hObject,callbackdata,P)
  handle = findobj('Tag','fig');
  var = guidata(handle);
  var.f = (hObject.Value);
  var = updatePlot(var,P);
  guidata(handle, var);
  freq_disp = findobj('Tag','freq_disp');
  set(freq_disp,'String',num2str(hObject.Value));
end

% Callback function when Ey_phase slider changes
function update_t(hObject,callbackdata,P)
  handle = findobj('Tag','fig');
  var = guidata(handle);
  var.t = (hObject.Value);
  var.t = var.t*2*pi/var.w; % time, s
  var = updatePlot(var,P);
  guidata(handle, var);
  t_disp = findobj('Tag','t_disp');
  set(t_disp,'String',num2str(hObject.Value));
end

function var = updatePlot(var,P)

var.w = 2*pi*var.f;% Frequency of wave, rads/s

% Medium 2
var.epsilon_2 = ...% Electric permittivity of medium 2, F/m
  var.er_2*P.Eo;
var.eta_c_2 = ... % Intrinsic impedance of medium 2, Ohms
  sqrt(var.mu_2/(var.epsilon_2 + var.sigma_2/var.w));
var.k1 = ...      % Wave number of medium 2,
  var.w*sqrt(var.mu_2*var.epsilon_2);
var.alpha_2 = ... % Attenuation coefficient of transmitted wave
  getAlpha(var.w,var.mu_2,var.epsilon_2,var.sigma_2/var.w);
```

```
var.beta_2 = ...  % Phase constant of transmitted wave
  getBeta(var.w,var.mu_2,var.epsilon_2,var.sigma_2/var.w);

% Medium 1
var.eta_c_1 = ... % Characteristic impedance of medium 1, Ohms
  sqrt(var.mu_1/(var.epsilon_1 + var.sigma_1/var.w));
var.k2 = ...      % Wave number of medium 2,
  var.w*sqrt(var.mu_1*var.epsilon_1);
var.alpha_1 = ... % Attenuation coefficient of transmitted wave
  getAlpha(var.w,var.mu_1,var.epsilon_1,var.sigma_1/var.w);
var.beta_1 = ...  % Phase constant of transmitted wave
  getBeta(var.w,var.mu_1,var.epsilon_1,var.sigma_1/var.w);

var.tau = getTau(var.eta_c_2,var.eta_c_1);   % transmission
 coefficient
var.gamma = getGamma(var.eta_c_2,var.eta_c_1); % reflection
 coefficient
var.Eo_r = ...
  var.gamma*var.Eo_i; % Magnitude of the reflected electric field, V/m
var.Eo_t = ...
  var.tau*var.Eo_i;   % Magnitude of the transmitted electric field,
 V/m

var.z2 = ... % Distance array, m
  0:pi/(var.beta_2*100):4*pi/var.beta_2;
var.z1 = ... % Distance array, m
  -4*pi/var.beta_1:pi/(var.beta_1*100):0;
var.Ei = ... % Incident Electric field propagating in the +z direction
  var.Eo_i*cos(var.w*var.t -
 var.beta_1*var.z1).*exp(var.alpha_1*var.z1);
var.Er = ... % Incident Electric field propagating in the +z direction
  var.Eo_r*cos(var.w*var.t -
 var.beta_1*var.z1).*exp(var.alpha_1*var.z1);
var.Et = ... % Incident Electric field propagating in the +z direction
  var.Eo_t*cos(var.w*var.t - var.beta_2*var.z2).*exp(-
var.alpha_2*var.z2);

% Plot the electric fields

% SI.solution_image(?,?)
set(var.h(1,1), 'Xdata',var.z1,'Ydata',var.Ei)
set(var.h(2,1), 'Xdata',var.z1,'Ydata',var.Er)
set(var.h(3,1), 'Xdata',var.z2,'Ydata',var.Et)
ylim([-max([var.Eo_i,var.Eo_t]),max([var.Eo_i,var.Eo_t])])
xlim([-max(abs(var.z1)),max(var.z2)])
end

function alpha =  getAlpha(w,mu,epsilon,epsilon_doublePrime)
  alpha =  w*((mu*epsilon/2)*(sqrt(1+(epsilon_doublePrime/
epsilon)^2)-1))^0.5;
end

function beta = getBeta(w,mu,epsilon,epsilon_doublePrime)
  beta = w*((mu*epsilon/2)*(sqrt(1+(epsilon_doublePrime/
epsilon)^2)+1))^0.5;
```

```
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%     Students only need to modify the functions below
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% returns the transmission coefficient
function tau =  getTau(eta_c_2,eta_c_1)
  tau =  2*eta_c_2/(eta_c_2+eta_c_1);
end

% returns the reflection coefficient
function gamma =  getGamma(eta_c_2,eta_c_1)
  gamma = (eta_c_2-eta_c_1)/(eta_c_2+eta_c_1);
end

function sigma =  sigmaScale(sigma)
  scale = 1e-10;
  sigma =  sigma * scale;
end
```

Solution





Published with MATLAB® R2015a