mt360_ur7_4_8_1 Boundary Conditions

Table of Contents

Title

```
Authors: Ulaby and Ravaioli
Book Edition: 7th
Problem #: 4.48
Last Updated: 2/20/2016
```

Description

With reference to Fig. 4-19, pg 206 in the book.

$$E_2 = \hat{x}3 - \hat{y}2 + \hat{z}(V/m), \epsilon_1 = 2\epsilon_0, \epsilon_2\ 18\epsilon_0$$

and the boundary has a surface charge density

$$\rho_s = 3.54 * 10^{-11}(C/m^2)$$

Exercise

1. Complete the function " electricFieldBoundaryConditions " in order to calculate E1, the angle E2 makes with the z-plane, and the angle E1 makes with the z-plane. This function is at the bottom of the Provided Code section. You do not need to alter any other code.

2. Play around with the interaction to answer the questions.

Questions

1. From part 1, what is E1, and what angle does E2 make with the z-plane?

2. From part 2, what affect does changing epsilon1, epsilon2, and rho_s have on E1? Why?

Useful Information

tangential components

The tangential component runs parallel to the boundary. In this exercise, the x and y components of E1 and E2 compose the tangential component.

$$E_t = \sqrt{E_x^2 + E_y^2}$$

Angle

The angle the electric field makes with the z-plane is determined by the tangential and normal component of E. In this exercise, the normal component is z.

$$\tan\theta = \frac{E_t}{E_n}$$

Boundary Conditions for the electric fields

$FieldComponent$	$AnyTwoMedia$
$TangentialE$	$E_{1t} = E_{2t}$
$TangentialD$	$\frac{D_{1t}}{\epsilon_1} = \frac{D_{2t}}{\epsilon_2}$
$NormalE$	$\epsilon_1 E_{1n} - \epsilon_2 E_{2n} = \rho_s$
$NormalD$	$D_{1n} - D_{2n} = \rho_s$

Provided Code

```
function boundary_conditions()
param

% Parameters

var.P.Eo = P.Eo;              % Electric permittivity of free space
var.E2.x = 3;                 % x component, m
var.E2.y = -2;                % y component, m
var.E2.z = 2;                 % z component, m
var.E2.epsilon = 18*P.Eo;     % Electric permittivity, F/m
var.E2.Et = 0;                % Tangential component, m
var.E2.theta = 0;             % Angle vector E2 makes with the z-plane,
                            % degrees

% These parameters are zero because they have not been computed yet.
var.E1.x = 0;                 % x component, m
var.E1.y = 0;                 % y component, m
var.E1.z = 0;                 % z component, m
var.E1.epsilon = 2*P.Eo;      % Electric permittivity, F/m
var.E1.Et = 0;                % Tangential component, m
var.E1.theta = 0;             % Angle vector E1 makes with the z
                            % plane, degrees

var.rho_s = 3.54e-11;         % Surface charge density at
                            % boundary, C/m^2
```

```
% Call the function at the beginning in order to initialize data
[var.E1,var.E2] = ...
electricFieldBoundaryConditions(var.E2,var.E1,var.rho_s);

epsilon1 = 2;
epsilon2 = 18;
rho = 3.45;

% The code below is used to create the interactive gui

f = figure(1),clf();
var.ax = axes('Parent',f,'position',[0.1 0.5  0.8 0.4]);
var.ax.YLim = ([-4,22]);

% plot initial data
h = plot([0,var.E1.Et],[0,var.E1.z],-[0,var.E2.Et],-[0,var.E2.z]);
legend('E1','E2','location', 'NorthWest')
title('Boundary Conditions')
var.txt1 = text(1,0,['E1 theta = ', num2str(var.E1.theta),'
 degrees']);
var.txt2 = text(1,-2,['E2 theta = ', num2str(var.E2.theta),'
 degrees']);

% User interactive controls
e1_ctrl = uicontrol('Parent',f,'Style','slider',...
        'Position',[60,20,420,20],'value',epsilon1,...
        'min',2,'max',20,'Callback',{@update_e1,h},...
        'Tag','slider1');
e1_ctrl.UserData = var;
e2_ctrl = uicontrol('Parent',f,'Style','slider',...
        'Position',[60,40,420,20],'value',epsilon2,'min',2,...
        'max',20,'Callback',{@update_e2,h});
rho_ctrl = uicontrol('Parent',f,'Style','slider',...
        'Position',[60,60,420,20],'value',rho,...
        'min',0,'max',20,'Callback',{@update_rho_s,h});

% controls to display slider names
e1_txt = uicontrol('Parent',f,'Style','text',...
       'Position',[0,20,60,20],'String','epsilon1');
e2_txt = uicontrol('Parent',f,'Style','text',...
       'Position',[0,40,60,20],'String','epsilon2');

        rho_txt = uicontrol('Parent',f,'Style','text',...
                  'Position',[0,60,60,20],'String','rho');

% controls to display slider values
e1_disp = uicontrol('Parent',f,'Style','text',...
        'Position',[480,16,20,15],'String',...
        num2str(e1_ctrl.Value),'Tag','e1_disp');
e2_disp = uicontrol('Parent',f,'Style','text',...
        'Position',[480,37,20,15],'String',...
        num2str(e2_ctrl.Value),'Tag','e2_disp');
rho_disp = uicontrol('Parent',f,'Style','text',...
         'Position',[480,59,80,15],'String',...
         [num2str(rho_ctrl.Value),' x10^-11'],'Tag','rho_disp');
```

```
var.ax.YLim = ([-4,22]);
end

% Callback function when epsilon1 slider changes
function update_e1(hObject,callbackdata,h)
 var = hObject.UserData;
 var.E1.epsilon = (hObject.Value)*var.P.Eo;
 [var.E1,var.E2] =  electricFieldBoundaryConditions(...
                    var.E2,var.E1,var.rho_s);
 set(h(1,1), 'YData',[0,var.E1.z]);
 set(var.txt1, 'String',['E1 theta = ',...
                num2str(var.E1.theta),' degrees'])
 set(var.txt2, 'String',['E2 theta = ',...
                num2str(var.E2.theta),' degrees'])
 hObject.UserData = var;
 var.ax.YLim = ([-4,22]);
 e1_disp = findobj('Tag','e1_disp');
 set(e1_disp,'String',num2str(hObject.Value));
end

% Callback function when epsilon2 slider changes
function update_e2(hObject,callbackdata,h)
  e1_h = findobj('Tag','slider1');
  var = e1_h.UserData;
  var.E2.epsilon = (hObject.Value)*var.P.Eo;
  [var.E1,var.E2] =  electricFieldBoundaryConditions(...
                     var.E2,var.E1,var.rho_s);
  set(h(1,1), 'YData',[0,var.E1.z]);
  set(var.txt1, 'String',['E1 theta = ',...
                 num2str(var.E1.theta),' degrees'])
  set(var.txt2, 'String',['E2 theta = ',...
                 num2str(var.E2.theta),' degrees'])
  guidata(hObject, var);
  var.ax.YLim = ([-4,22]);
  e2_disp = findobj('Tag','e2_disp');
  set(e2_disp,'String',num2str(hObject.Value));
end

% Callback function when rho slider changes
function update_rho_s(hObject,callbackdata,h)
  e1_h = findobj('Tag','slider1');
 var = e1_h.UserData;
  var.rho_s = (hObject.Value)*1e-11;
  [var.E1,var.E2] =  electricFieldBoundaryConditions(...
                     var.E2,var.E1,var.rho_s);
  set(h(1,1), 'YData',[0,var.E1.z]);
  set(var.txt1, 'String',['E1 theta = ',...
                 num2str(var.E1.theta),' degrees'])
  set(var.txt2, 'String',['E2 theta = ',...
                 num2str(var.E2.theta),' degrees'])
  guidata(hObject, var);
  var.ax.YLim = ([-4,22]);
  rho_disp = findobj('Tag','rho_disp');
  set(rho_disp,'String',[num2str(hObject.Value),' x10^-11']);
```

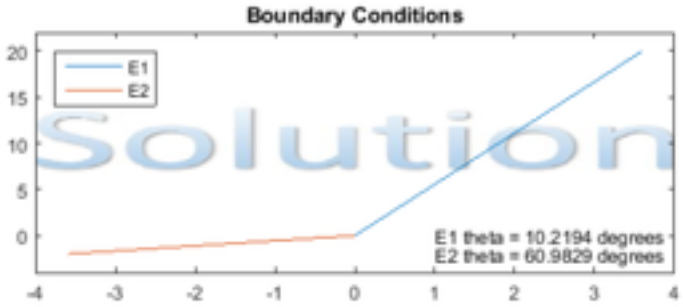
```
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%          Students only need to alter the function below
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% This function needs to be completed by the user.
function [E1,E2] =  electricFieldBoundaryConditions(E2,E1,rho_s)

E2.Et =% INSERT CODE HERE              % Tangential component of E2,m
E2.theta = % INSERT CODE HERE          % The angle E2 makes
                                     % with the z-axis, degrees
E1.x = % INSERT CODE HERE              % X-component of E1, m
E1.y = % INSERT CODE HERE              % Y-component of E1, m
E1.Et = E2.Et;                         % Tangential component of E1, m
E1.z = % INSERT CODE HERE              % The angle E1 makes
                                     % with the z-axis, degrees
E1.theta = atan(E1.Et/E1.z)*180/pi;    % The angle E1 makes
                                     % with the z-axis, degrees
end
```

Solution



Published with MATLAB® R2015a