MATLAB EXERCISE 2.5 **Oblique boundary plane with nonzero surface charge.** Repeat the previous MATLAB exercise but for an oblique boundary plane with nonzero surface charge of density ρ_s on it. *(ME2_5.m on IR)*

SOLUTION:

```
%
%   Book: MATLAB-Based Electromagnetics (Pearson Prentice Hall)
%   Author: Branislav M. Notaros
%   Instructor Resources
%   (c) 2011
%
%   This MATLAB code or any part of it may be used only for
%   educational purposes associated with the book
%
%
%

% Dielectric - dielectric boundary conditions - Oblique boundary plane with
% nonzero surface charge

clear all;
close all;
EPS0 = 8.8542*10^(-12);

NORMALx = input('Enter x-component of the normal: ');
NORMALy = input('Enter y-component of the normal: ');
NORMALz = input('Enter z-component of the normal: ');
% Electric field vector
Ex1 = input('Enter x-component of E-field in medium 1, in V/m: ');
Ey1 = input('Enter y-component of E-field in medium 1, in V/m: ');
Ez1 = input('Enter z-component of E-field in medium 1, in V/m: ');

RHOS = input('Enter the surface charge density in pC/m^2: ');
RHOS = RHOS*10^(-12);
% Dielectrics
EPSR1 = input('Enter the relative permittivity of medium 1: ');
EPSR2 = input('Enter the relative permittivity of medium 2: ');

if (NORMALx~=0 || NORMALy~=0 || NORMALz~=0)
NORMALmag = sqrt(NORMALx^2 + NORMALy^2 + NORMALz^2);
NORMALx = NORMALx/NORMALmag;
NORMALy = NORMALy/NORMALmag;
NORMALz = NORMALz/NORMALmag;
NORMAL = [NORMALx, NORMALy, NORMALz];

Emag = sqrt(Ex1^2 + Ey1^2 + Ez1^2);
E1 = [Ex1,Ey1,Ez1];

alphaAngle = acos((dot(NORMAL,E1'))/Emag);
E1normal = Emag*cos(alphaAngle).*NORMAL;
E1tangential = E1 - E1normal;

E2normal = (E1normal.*EPSR1*EPS0 - RHOS.*NORMAL)/(EPSR2*EPS0);
E2tangential = E1tangential;
E2 = E2normal + E2tangential;

disp('E-field in medium 2, in V/m, is:');
```

```
fprintf('(%.3f)*ux',E2(1));
fprintf(' + (%.3f)*uy',E2(2));
fprintf(' + (%.3f)*uz\n',E2(3));

A = [E1(1),E1(2),E1(3),E2(1),E2(2),E2(3),NORMALx,NORMALy,NORMALz];
B = max(abs(A)) + 0.1;

    figure(1);
    if NORMALz ~= 0
        [x,y] = meshgrid(-B:2*B:B,-B:2*B:B);
        Bz = -1/NORMALz.*(NORMALx.*x + NORMALy.*y);
        h = surf(x,y,Bz);alpha(0.4);axis equal; hold on;
    elseif NORMALy ~= 0
        [x,z] = meshgrid(-B:2*B:B,-B:2*B:B);
        By = -1/NORMALy.*(NORMALx.*x + NORMALz.*z);
        h = surf (x,By,z);alpha(0.4);axis equal; hold on;
    elseif NORMALx ~= 0
        [y,z] = meshgrid(-B:2*B:B,-B:2*B:B);
        Bx = -1/NORMALx.*(NORMALy.*y + NORMALz.*z);
        h = surf (Bx,y,z);alpha(0.4);axis equal; hold on;
    end
    colormap(white);
    plot3(0,0,0,'ko','MarkerFaceColor','k'); hold on;
    quiver3(0,0,0,NORMALx, NORMALy, NORMALz,0,'r','LineWidth',2);
    text (NORMALx/2, NORMALy/2, NORMALz/2,'n');
    quiver3(0,0,0,E1(1),E1(2),E1(3),0,'b', 'LineWidth',2);
    text (E1(1)/2,E1(2)/2,E1(3)/2,'E1');
    quiver3(0,0,0,E2(1),E2(2),E2(3),0,'g','LineWidth',2);
    text (E2(1)/2,E2(2)/2,E2(3)/2,'E2');
    xlabel('x [m]');
    ylabel('y [m]');
    zlabel('z [m]');
else
    disp('Error - normal cannot be zero');
end
```