MATLAB EXERCISE 1.13 **Vector numerical integration and field visualization using `quiver`.** MATLAB function `quiver` is used for visualization of field vectors in space. Input data are coordinates of nodes in a mesh in a Cartesian coordinate system and intensities of field components at the nodes. Implement `quiver` to visualize the electric field distribution due to a uniform straight line charge of finite length a and total charge Q placed along the x-axis and centered at the coordinate origin. The electric field vector at each node of the mesh should be computed by vector numerical integration of elementary fields due to equivalent point charges along the line representing short segments into which the line is subdivided. With such an integration (superposition) procedure, this MATLAB program may be applicable, with minor modifications, to many similar and more complex charge distributions, where the analytical expression for electric field components is not available or is difficult to find. *(ME1_13.m on IR)*

SOLUTION:

Fig.S1.8 shows a visualization of the resultant field distribution.

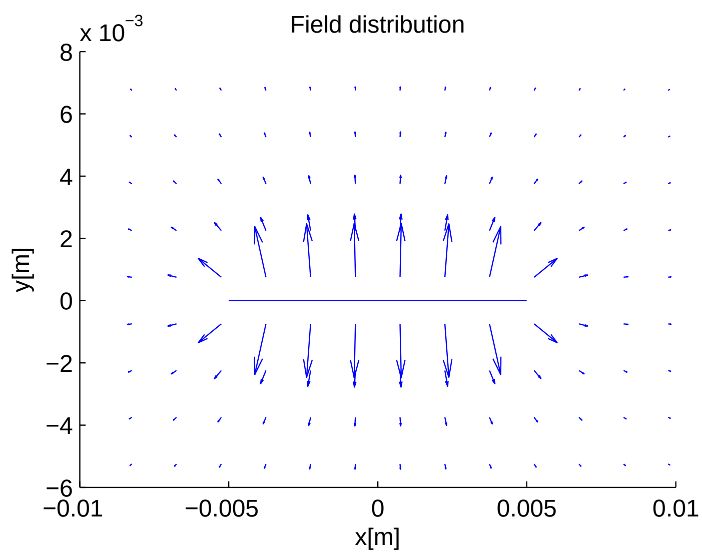

Figure S1.8 Visualization of the electric field distribution due to a uniform straight line charge of finite length ($a = 1$ cm and $Q = 1$ nC) using MATLAB function `quiver`, where the electric field vector at mesh nodes is previously computed, in MATLAB, as a sum of elementary fields due to equivalent point charges modeling the line, as explained in detail in the TUTORIAL (in the book); for MATLAB Exercise 1.13.

```
%
%   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
%
%
%


%   Vector numerical integration and field visualization using quiver


clear all;
close all;

EPS0 = 8.8542 * 10^(-12);
a = input('Dimension of a line at the origin in cm: ');
a = a*10^(-2);
Q = input('Total charge in nC: ');
Q = Q *10^(-9);
rho = Q/a ;
dl = 0.001*a;
xline = -a/2 : dl : a/2;
yline = zeros (1,length (xline));

v = -0.825*a: 0.15*a : 0.975*a;
u = -0.525*a: 0.15*a : 0.675*a;
[x,y] = meshgrid(v,u);
[M,N] = size(x);

for i = 1 : M
    for j = 1 : N
        for t = 1 : length(xline)
            r(i,j,t,:) = [x(i,j) - xline(t) y(i,j) - yline(t)];
            rabs(i,j,t) = vectorMag(r(i,j,t,:));
            runit(i,j,t,:) = r(i,j,t,:)/rabs(i,j,t);
            E(i,j,t,:) = rho*dl*runit(i,j,t,:)/(4*pi*EPS0*rabs(i,j,t)^2);
        end;
        Etotx(i,j) = sum(E(i,j,:,1));
        Etoty(i,j) = sum(E(i,j,:,2));
    end;
end;
figure(1);
line([-a/2 a/2],[0 0]);
hold on;
quiver(v,u,Etotx,Etoty);
hold off;
title('Field distribution');
```

```
xlabel('x[m]');
ylabel('y[m]');
```