LAB #2

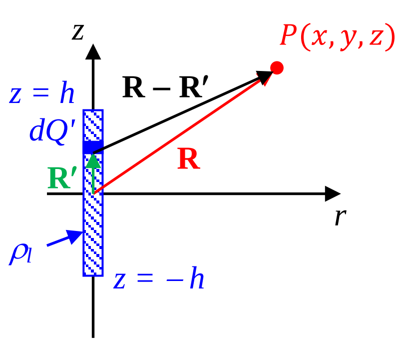
The purpose of Lab #2 is to get you to start writing MATLAB programs which give you numerical solutions to electrostatic problems that are already familiar to you from analytical treatment during the lectures and from your textbook. You will see that with relatively little programming effort very accurate numerical results can be obtained which are more general than the analytical formulas derived in your textbook. You will end up with very versatile MATLAB programs of your own. You will be able to solve problems which are difficult or impossible to solve analytically. We will expect that you are now reasonably comfortable with using MATLAB through your work in Labs #0 and #1.

INTRODUCTION
Numerical Integration with MATLAB

For many electrostatic and magnetostatic problems, we cannot arrive at an analytic solution due the difficulty of the integrations which may be involved. From your own experience, you may have noticed that we can easily find the electric field on the axis of a charged disk. However, for measurement points off the axis, the required integrations become much more challenging to solve. This is one example of a case in which we can turn to MATLAB to do the integrations for us and return the values of the fields at these off-axis points. This is the purpose of numerical integration.

As with most mathematical software packages, MATLAB includes a number of built-in functions that are able to calculate the value of a definite integral. Functions such as *quadl* or *trapz* use specialized techniques to quickly evaluate integrals. However, since the purpose of these labs is to give you the opportunity to continue to learn how to use MATLAB you will be writing you own functions to perform integral calculations. In addition, by writing these functions it will help you better understand the analytic process which you need to use to solve Coulomb's law problems. You may use the built-in functions to verify your own functions' results, but they cannot be the only method used to evaluate the integrals in these labs.

To introduce you to the idea of numeric integration we will consider the case of determining the electric field due to a finite line charge in free space at an arbitrary point $P(x, y, z)$. We will work in Cartesian coordinates, since then we will not have to be concerned with integrating unit vectors which may change with position.



The electric field intensity for this distribution can be found by using Coulomb's law:

$$\mathbf{dE} = \frac{dQ'}{4\pi\varepsilon_0|\mathbf{R} - \mathbf{R}'|^3}(\mathbf{R} - \mathbf{R}')$$

<u>Step 1)</u> Define dQ'

$$dQ' = \rho_l dz'$$

<u>Step 2)</u> Determine the position and distance vectors

$$\mathbf{R} = x\hat{\mathbf{x}} + y\hat{\mathbf{y}} + z\hat{\mathbf{z}}$$
$$\mathbf{R}' = z'\hat{\mathbf{z}}$$
$$\mathbf{R} - \mathbf{R}' = x\hat{\mathbf{x}} + y\hat{\mathbf{y}} + (z - z')\hat{\mathbf{z}}$$

Step 3) Simplify the integrand

$$\mathbf{dE} = \frac{\rho_l dz'}{4\pi\varepsilon_0[x^2 + y^2 + (z - z')^2]^{3/2}}[x\hat{\mathbf{x}} + y\hat{\mathbf{y}} + (z - z')\hat{\mathbf{z}}]$$

Step 4) Evaluate the integral

$$\mathbf{E}(x, y, z) = \int \mathbf{dE} = \int_{-h}^{h} \frac{\rho_l dz'}{4\pi\varepsilon_0[x^2 + y^2 + (z - z')^2]^{3/2}}[x\hat{\mathbf{x}} + y\hat{\mathbf{y}} + (z - z')\hat{\mathbf{z}}]$$

This integral of the vector **dE** leads to the three scalar integrals:

$$E_x(x, y, z) = \frac{\rho_l x}{4\pi\varepsilon_0}\int_{-h}^{h} \frac{dz'}{[x^2 + y^2 + (z - z')^2]^{3/2}}$$

$$E_y(x, y, z) = \frac{\rho_l y}{4\pi\varepsilon_0}\int_{-h}^{h} \frac{dz'}{[x^2 + y^2 + (z - z')^2]^{3/2}}$$

$$E_z(x, y, z) = \frac{\rho_l}{4\pi\varepsilon_0}\int_{-h}^{h} \frac{(z - z')dz'}{[x^2 + y^2 + (z - z')^2]^{3/2}}$$

These integrals are not impossible to evaluate and we could work towards an analytic solution to this problem. But it is a good example of how we can use MATLAB to finish the solution and evaluate the electric field at the general point $P(x,y,z)$.

To do this we can write our own function in MATLAB. User-defined functions are what make MATLAB so powerful, as you can make use of all of its mathematical utilities and built-in functions to do an infinite variety of tasks. As you may see in later years, it can be used to define and run a control system, or be used in many different ways to process audio, visual, or other types of signals.

Just as in many other programming languages, a MATLAB function takes a set of input arguments and generates a set of output arguments based on the code contained in the function. As you know from Lab #1, a function cannot be entered directly at the command prompt, >>, rather, they must be created in the MATLAB editor and saved as an **.m** file.

As you have seen in Lab #1 a simple example, would be:

```
function [output] = polynomial(x)
output = 2*x.^2-5*x+1;
```

This function takes a vector `x`, and calculates a vector of the same size, `output`, which is the value of the polynomial $2x^2 - 5x + 1$ for each element of `x`. Note, there is not an “end function” command required for MATLAB functions.

Multiple inputs and outputs can also be used, as the following example illustrates:

```
function [Etot,Ex,Ey,Ez] = unknown(Q,R,theta,phi)

epsilon = 8.854e-12;
Etot = Q/(4*pi*epsilon*R^2);
Ex = Etot*sin(theta)*cos(phi);
Ey = Etot*sin(theta)*sin(phi);
Ez = Etot*cos(theta);
```

What do you think this function does?

To implement our numerical integration function we must "discretize" the line of charge and break it up into N pieces. This turns our integration into a Riemann sum, for example:

$$E_x(x,y,z) = \frac{\rho_l x}{4\pi\varepsilon_0}\int_{-h}^{h}\frac{dz'}{[x^2+y^2+(z-z')^2]^{3/2}} \quad \rightarrow \quad E_x(x,y,z) \approx \frac{\rho_l x}{4\pi\varepsilon_0}\sum_{k=1}^{N}\frac{\Delta z'}{[x^2+y^2+(z-z_k')^2]^{3/2}}$$

Therefore, one way to code this in MATLAB might be:

```
function [Etot,Ex,Ey,Ez]=lineofcharge(h,rhol,x,y,z,N)

epsilon=8.854e-12;

dz=2*h/N;  % Discretize the total line length of 2h into N pieces

zprime=linspace(-h,h,N); % The linspace command creates a vector that ranges
                         % from -h to h and has N elements with equal spacing

% Use a for loop to "walk" along the line
for k=1:length(zprime)
    % Evaluate the expression which is the same for each component of E
    integrand=dz/((x^2+y^2+(z-zprime(k))^2)^(3/2));

    % Evaluate the differential elements for each component of E, which
    % arise from that small part of the line, i.e., dQ at zprime(k).
    dEx(k)=integrand;
    dEy(k)=integrand;
    dEz(k)=(z-zprime(k))*integrand;
end

% Do the "integration" by summing up the differential pieces that result from
% each value of zprime.
Ex=((rhol*x)/(4*pi*epsilon))*sum(dEx);
Ey=((rhol*y)/(4*pi*epsilon))*sum(dEy);
Ez=(rhol/(4*pi*epsilon))*sum(dEz);
Etot=(Ex^2+Ey^2+Ez^2)^0.5;
```

A more efficient way to calculate the vectors, `dEx`, `dEy`, and `dEz`, would be to eliminate the `for` loop and use MATLAB dot operators. This means that the `for` loop structure would be replaced with these four lines:

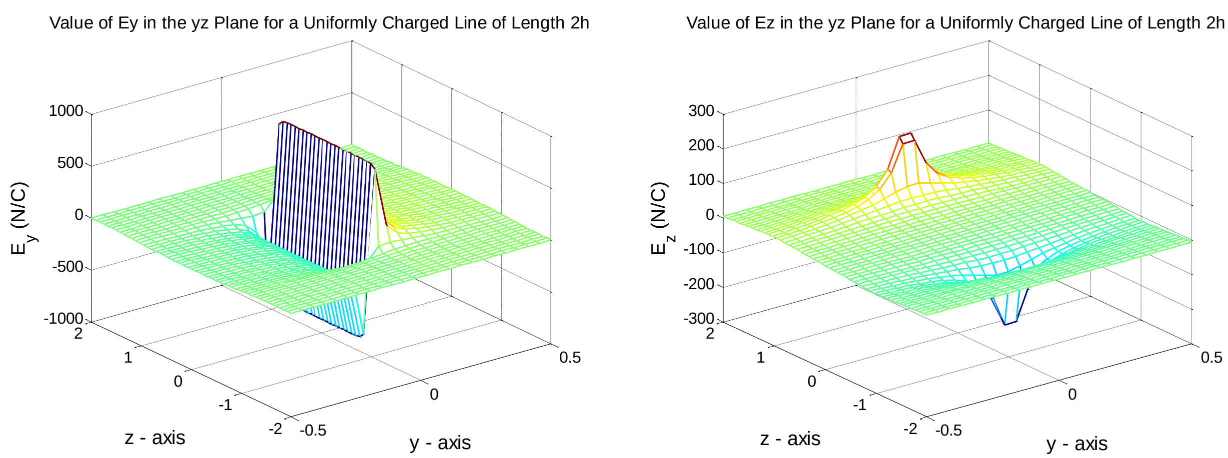
```
integrand=dz./((x^2+y^2+(z-zprime).^2).^(3/2));
dEx=integrand;
dEy=integrand;
dEz=(z-zprime).*integrand;
```

Observe that the vectors `integrand`, `dEx`, `dEy`, and `dEz`, will all be the same size as `zprime`. Indeed, each element of these vectors relates to the corresponding element of `zprime`, i.e., that point on the charged line. These represent the small amount of field created in each of these three coordinates that is created by that small piece of charge (`dQ'`) at `zprime`.

We have decided to have N be one of our input variables so that we can control the accuracy and computation time of the function. As N is increased the accuracy should increase, but so will the time it takes for the function to run.

Now that we have the ability to find the electric field from a line charge at any point, we can use this to visualize the field distribution for a line charge. For example, we can plot the strength of the E_y and E_z components in the *yz*-plane by running the following function with values of h = 1 m, *rhol* = 1e-9 C/m, N = 500. The resulting plots are shown below.

```
function [Etot,Ex,Ey,Ez]=plotlineofcharge(h,rhol,N)

x=0;  % We will focus on the fields in the yz plane
y=linspace(-0.5,0.5,24);  % These define the y and z values of the
z=linspace(-2,2,50);      % points at which we want to find the fields.

% These for loops calculate the field components at each point of interest
% in the yz plane.  As a result we have matrices that are created, with
% the rows corresponding to fixed z values, and the columns corresponding
% to fixed y values.
for e=1:length(z)
    for m=1:length(y)
        [Etot(e,m),Ex(e,m),Ey(e,m),Ez(e,m)]=lineofcharge(h,rhol,0,y(m),z(e),N);
    end
end

[Y,Z]=meshgrid(y,z); % Create the required arrays for the 3D plots

figure % Create a figure for the view of the Ey component of the field
mesh(Y,Z,Ey);
xlabel('y - axis');
ylabel('z - axis');
zlabel('E_y (N/C)');
title('Value of Ey in the yz Plane for a Uniformly Charged Line of Length 2h');
grid on;

figure % Create a figure for the view of the Ez component of the field
mesh(Y,Z,Ez);
xlabel('y - axis');
ylabel('z - axis');
zlabel('E_z (N/C)');
title('Value of Ez in the yz Plane for a Uniformly Charged Line of Length 2h');
grid on;
```



From these plots we can see that the field decreases quite rapidly ($\propto 1/r^2$) as one moves away from the line charge. Notice that in defining y and z, we avoided the exact points where the line charge existed, i.e., $y = 0$. This way we avoided the singularity that would have existed in the calculations, i.e., $1/0$ when all three coordinates were zero.

We can also plot these fields in another way, by using 2D graphs. For example, we can add the following commands to the end of the function `plotlineofcharge` in order to plot how the y and z components of **E** change with respect to y when $z = 0$, i.e., in the plane which bisects the line of charge:

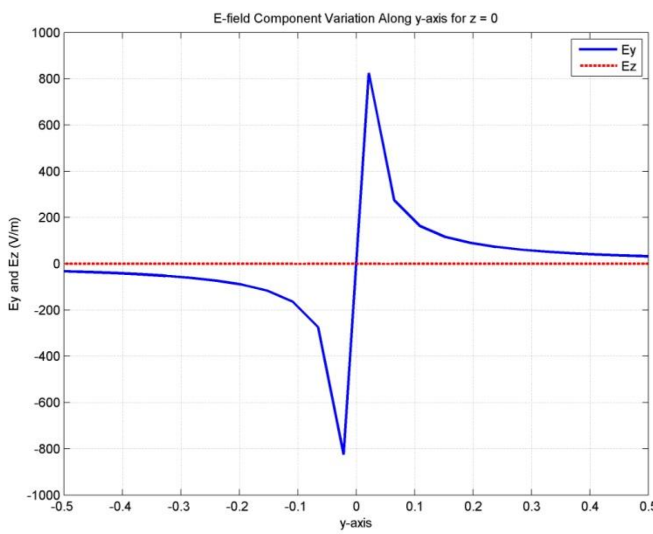
```
figure;
plot(y,Ey(26,:),y,Ez(26,:),'r--');
```

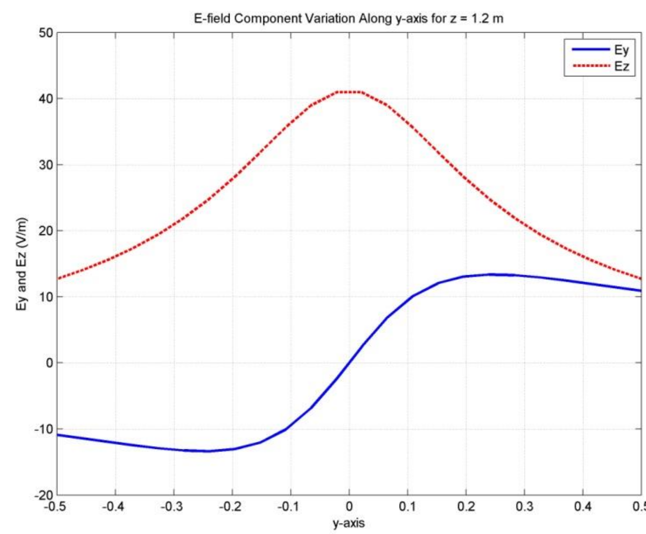
Note, the 26th entry of the `z` vector is 0, meaning `z(26)` = 0. This represents the position of the bisecting plane. This command results in the plot shown below on the left. The appropriate labels and titles have been added, and the function `legend` was used to specify which trace related to E_y and which was related to E_z.

Alternatively, we could plot the variation of these components with y just above the tip of the line charge [`z(41)` = 1.2] by adding the following commands to the function:

```
figure;
plot(y,Ey(41,:),y,Ez(41,:),'r--');
```

This gives the right-hand plot shown below.





You should verify that these plots make sense, meaning that they correspond with the behavior of the fields that you would predict from theory using superposition and Coulomb's law concepts.

The use of MATLAB to evaluate integrals which cannot be done by hand can be extremely useful in visualizing the fields from complex charge and current distributions. During this lab and the next lab, you will get experience in writing your own functions to determine the electric field from a ring of charge, and the electric field inside and outside a charged sphere.

Lab Section (circle): PRA01 PRA02 PRA03 PRA04 PRA05 PRA06

Last Name: _______________ First Name: _____________ Student #: _____________

PREPARATION - Individual

1) Read through these Lab #2 notes carefully, pay particular attention to the above introduction on numerical integration.

2) Review the solution to the electric field at a point $P(0, 0, z)$ above a uniformly charged circular ring of charge.

3) Use Coulomb's Law to set up the solution to the electric field at an *arbitrary point* $P(x, y, z)$ of a uniformly charged, circular ring of charge centered about the origin and lying in the xy plane. Do not attempt to do the integrations! As has been done above for the finite-line charge, just come up with the three integrations you will have to do for the E_x, E_y, and E_z components of the total electric field at $P(x, y, z)$. Write your solution in the box below:

4) Create a MATLAB function called *ringofcharge.m*, that carries out the following tasks:
   a) Takes as inputs the radius of the ring (a), the uniform charge density (ρ_ℓ), the point location (x, y, and z), and the number of steps for the integration (N).
   b) Numerically evaluates the three one-dimensional integrations required to find the electric field components E_x, E_y, and E_z of the circular ring of charge at a point $P(x, y, z)$. You cannot use any of the built-in MATLAB functions for doing this integration, such as *quadl* or *trapz*. These can be used to verify your own algorithm's result, but cannot be used in your *ringofcharge* function.
   c) Calculates the magnitude of the total electric field at this point.

This means that your function should be able to return the values for E_x, E_y, E_z and E_{tot} at any point, $P(x, y, z)$, for any uniformly charged ring centered about the z – axis and lying in the xy-plane. Your result from part 3) above should help, as well as the sample function `lineofcharge.m` shown in the introduction. Each student must submit their ***working*** code (i.e., debugged) by ***handwriting*** it in the box below. ***It is essential that you debug this code before you come to the lab, otherwise you will not finish the lab.*** You should keep an electronic copy of your code which you can run in the lab.
***Make sure this portion of the lab is your own work*!**

5) Answer the questions below.

Electric Fields Along the *z* – axis

i) What is the expression for the electric field at any point along the axis of a ring, which is uniformly charged ring (ρ_ℓ), has a radius a, lies in the xy plane and is centered about the origin (meaning the ring axis is the z-axis)?

ii) How could you make use of the expression written in part i) above, to verify your *ringofcharge.m* algorithm and code? Come up with a testing plan which includes at least two test cases that you will use in the lab session to make sure that your code is running properly. As a hint, one test case has been given to you.

Testing Plan:

Testing Cases:

Test Case #1: $a = 1$ cm, $\rho_l = -1$ nC, $x = 0$ m, $y = 0$ m, and $z = 2$ m
$\hat{\mathbf{E}}_{\text{Theory}} = -0.141\hat{\mathbf{z}}$

Test Case #2:

Use these two test cases to help you debug your code, and verify that it is working correctly.

Testing Results:

Test Case #1: $a = 1$ cm, $\rho_l = -1$ nC, $x = 0$ m, $y = 0$ m, and $z = 2$ m
$\hat{\mathbf{E}}_{\text{Theory}} = -0.141\hat{\mathbf{z}}$, $\hat{\mathbf{E}}_{\text{Matlab}} =$

Test Case #2:

Electric Fields Along the *y* – axis

For a uniformly charged ring lying in the xy plane and centered about the origin, what component(s) of the electric field will be non-zero for points along the y – axis? Explain your reasoning.

IN-LAB WORK - Group

Evaluation and Visualization of the Electric Field for a Charged Ring

1. Validating Your *ringofcharge.m* Function

To ensure that your function is properly calculating the electric field of a charged ring, do the following two exercises.

1.1. Electric Fields Along the z-axis

Now consider the case in which a uniformly-charged ring lies in the xy plane and is centered about the z-axis. It has a radius of $a = 0.5$ m and a total charge of $Q = 3$ mC.

Write another short function, *plotringcharge.m*, that calls your *ringofcharge.m* function to calculate the electric field components at points along the z – axis between $z = -3$ m and $z = 3$ m. Use $N = 500$ for your calculations.

- This function should plot the calculated values of E_z along the z – axis and compare this to the theoretical values (E_{theory}), determined from the expression stated in your preparation (part 5).
- Plot the theoretical values (E_{theory}) with a blue line, and the calculated values with red circles.
- Properly label your axes, and include a legend in your figure (see *help legend*).
- You should have at least 100 data points, but no more than 400. If these plots do not correspond, then you will have to correct your *ringofcharge* function.

1.2. Electric Fields Along the y – axis

Modify your *plotringofcharge* function to calculate and plot the electric field components E_x, E_y, and E_z, along the y – axis from 0.1 m to 2 m. Again, the ring has a radius of $a = 0.5$ m and a total charge of $Q = 3$ mC.

- Save this as a new function, as you will be asked to use the original *plotringofcharge* function again.
- Use an N value of 500.
- Verify that the correct component(s) are zero (or very close to zero), as you predicted in your preparation. Why are these components not exactly zero?

Before you leave, make sure to discuss the results of this section with your TA.

2. Electric Fields for Non-uniformly Charged Rings, General Points, and Asymmetric Charge Distributions

2.1. Fields due to Non-uniform Charge Distributions

Modify your *ringofcharge* function so that you can calculate electric fields for non-uniform charge densities.

- Then use your *plotringofcharge* function to plot E_x, E_y, and E_z along the z – axis, from $z = -3$ m to $z = 3$ m for the non-uniform charge distribution given to you by your TA.
- Use the values of $a = 0.5$ m and $N = 500$. For this charge distribution describe how the fields are different from those of the uniformly charged ring.

____

____

Before you leave, make sure to discuss the results of this section with your TA.