Lab #2 – The Electric Field of a Dipole

OBJECTIVES

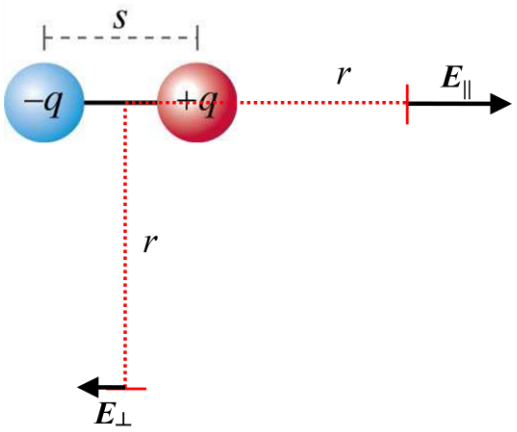
In this lab you will:

- Create a dipole using VPython
- Use vectors to map the Electric Field patterns of the dipole in 3D

From investigating the Electric Field of a dipole in lecture, you have learned very simple, yet powerful formulas for calculating the E-field of a dipole:

Along the *dipole axis*: $\left|\vec{E}_{||}\right| = \frac{1}{4\pi\epsilon_0}\frac{2qs}{r^3}$

And *perpendicular* to the dipole axis: $\left|\vec{E}_{\perp}\right| = \frac{1}{4\pi\epsilon_0}\frac{qs}{r^3}$



To calculate the Electric field at any location other than these two axes, you must use the *superposition principle*. Using VPython, you will create a program that allows easy calculation of the electric field at any point in space. Then, by having VPython display arrows representing the E-Field at multiple locations, you will be able to visualize the E-field pattern of an electric dipole.

1) Warm-Up Problem

Problem (1) A negatively charged *(-e)* atom and a positively charged *(+e)* atom are located at $\langle 0.5\times10^{-9}, 0, 0\rangle$ *m* and $\langle -0.5\times10^{-9}, 0, 0\rangle$ *m*, respectively. Choose the observation location to be $\langle 1.8\times10^{-9}, 1.0\times10^{-9}, 0\rangle$ *m*.

(a) Draw a diagram for the above situation. Include each charged particle, the observation location, the *relative position vectors* from each particle to the observation location, the individual Electric field vectors, and the *net* Electric field vector at the observation location. Label all the objects with descriptive names.

(b) Calculate the net Electric field at the observation location.

CHECKPOINT 1: Ask an instructor to check your work for credit.
You may proceed while you wait to be checked off

2) Electric Field at One Location

a) Open your code from Lab 1.

b) Save the code under a new name so you still will have access to your Lab 1 code.

You will now remove several lines and variables from your code in order to make it into a framework for this lab.

c) Under the "Create Objects" section, remove any additional objects so that only atom1 and atom2 remain. Move the code creating Earrow to the bottom of the program.

d) Towards the bottom of your code, eliminate the line that assigns Enet to be the arrow's axis.

e) Within your "Calculations" section, retain all calculation for E1 and E2 (the calculations pertaining to atom1 and atom2), but remove any additional calculations. Keep the calculation for Enet, but it should only contain E1 and E2.

Now you can now begin constructing the new program:

f) Change the locations of atom1 and atom2 $\langle 0.5 \times 10^{-9}, 0, 0\rangle$ *m* and $\langle -0.5 \times 10^{-9}, 0, 0\rangle$ *m*, respectively.

g) Give atom1 and atom2 a charge of –e and +e, respectively.

h) Make the negatively charged atom *blue* and the positively charge atom *red.*

i) Change the radius of each atom to be 1 Angstrom, 10^{-10} m (This is more reasonable for the size of actual atoms)

j) Change the observation location to $\langle 1.8 \times 10^{-9}, 1.0 \times 10^{-9}, 0\rangle$ *m* :

```
Obslocation = vector(1.8e-9,1.0e-9,0)
```

Since we have moved the arrow to the bottom of the screen, we now need to change how we calculate the relative position vectors r1 and r2 – so they only reference the observation location directly:

k) Change the lines that calculate the relative position vectors to:

```
r1=Obslocation-atom1.pos
r2=Obslocation-atom2.pos
```

l) In the line of code that creates the arrow, change the axis to Enet times your scale factor:

```
Earrow = arrow(pos = Obslocation, axis= Enet*scalefactor, color=color.orange)
```

Before you run your code, what value should the scale factor be set to?

Use your answer to part (b) of the Warm-Up problem and the sizes/positions of the objects within the system to calculate an appropriate scale factor.

m) Set the scale factor to the appropriate value. Run the code and draw what you see.

n) Do the diagram and the value for Enet agree with your solution to the Warm-Up problem?

CHECKPOINT 2: Ask an instructor to check your work for credit.
You may proceed while you wait to be checked off

3) Adding More Observation Locations

To see the Electric Field pattern of this dipole, you will extend your program to calculate the electric field at many different locations – all of which will lie on circles centered on the dipole. The circles will have a radius of 2x 10^{-9} m. Instead of copying and pasting code many times, you will place your calculations within a loop. During each iteration of the loop, VPython will create an arrow representing the Electric field at a different observation location. An easy and efficient method of moving the observation locations around a circle (of fixed radius) is to use polar coordinates.

a) In your "Set Initial Values" section, set the initial value of an angle, theta, to 0. Also, set the radius , r, to be 2x 10^{-9} m

```
theta = 0
r = 2e-9
```

b) Place the entirety of your "Calculations" section into a while loop that terminates when theta is 2π. Type the following before your calculations and *indent* all of your calculations:

```
while theta < 2*pi:
```

c) Since you are changing the observation location during each iteration, and since each calculation is dependent on the observation location, you must add the following line of code to your loop, just beneath the while statement:

```
Obslocation = vector(?,?,0)
```

Using theta and r, replace the question marks above with the appropriate polar values. VPython has the sine and cosine functions built-in, e.g. cos( ) and sin( ) are valid functions.

d) Add a print statement for the Observation location just above the print statement for Enet:

```
print "The observation location is",Obslocation,"m"
print "Enet =",Enet,"N/C"
```

e) At the very end of your loop, calculate the new value of theta with the following line:

```
theta = theta + pi/6
```

This will allow the loop to iterate 12 times before the value of theta meets or exceed 2π

f) Run your program.

It should calculate and display the electric field as arrows at 12 locations on a circle surrounding the dipole. If needed, adjust your scale factor.

g) Look at your display. Does it make sense? Do the magnitudes and directions of the 12 orange arrows make sense?

Compare the two vectors along the dipole axis with the vector directly above and the vector directly below the dipole center.

What is the relationship between their magnitudes?

CHECKPOINT 3: Ask an instructor to check your work for credit. You may proceed while you wait to be checked off

4) Adding more locations in a different plane

a) Currently, we have plotted the Electric field at 12 locations around a circle in the *xy* plane using polar coordinates. Find a way to plot the E-field vectors in at least one additional plane (e.g. *xz plane)*– again, along a circle of the same radius and centered on the dipole. *(Hint: you can simply copy and paste the while loop and make minor changes to the initial observation location. OR, you can place the entire loop within another while loop and use spherical coordinates)*

b) Run your program

CHECKPOINT 4: Ask an instructor to check your work for credit.