Lab: Electric Field of a Uniformly Charged Rod

The goal of this VPython problem is to write a program to calculate and display the electric field of a uniformly charged rod at locations not in the plane bisecting the rod (in other words, at locations where the analytical formula does not apply). The rod has a length 4 m, and a total charge of 9e-8 C. You will divide the rod into a number of pieces, which you will approximate by point charges; then apply the superposition principle to get the net field at the observation locations.

Planning: Do sections 1-5 on a whiteboard

1) Diagram: Consider a rod of length 4 m, oriented along the x-axis, with the center of the rod at the origin. The rod is positively charged. Initially you will divide into 8 equal length segments. Initially the observation location will be $< -1, 0.4, 0 >$ m. Draw axes, the rod, the segments, the observation location, and what you expect the net electric field to look like at this observation location.

In your program, you will write a loop to "step" through the rod piece by piece, starting at the left end and moving to the right. For each piece, you will find $\Delta\vec{E}$, the contribution of that piece to the net field at the observation location. The following questions will help you create algebraic expressions for important quantities to use in the program.

2) You will start by approximating the rod as 8 point charges. What is the length of each "point-like" segment of the rod?

Now write a general symbolic expression for the calculation you just did. If the length of the rod is *L*, and the number of pieces is *N*, what is a symbolic expression for Δx, the length of one piece, in terms of *L* and *N*?

3) What is the amount of charge on each of the 8 pieces?

Now write a general symbolic expression for the calculation you just did. If the total charge of the rod is *Q,* and the number of pieces is *N*, what is a symbolic expression for ΔQ, the amount of charge on one piece, in terms of *Q* and *N*?

4) What is the position vector of the *center* of the leftmost piece of the rod (relative to the origin)? Call this piece #1.

Now write a general symbolic expression for the calculation you just did: If the length of the rod is *L*, and the length of one piece is Δx, what is the position of the center of the leftmost piece of the rod? Your answer should be a vector (relative to the origin). Hint: what is the x-coordinate of the left end of the rod?

5) What is the position vector of the center of piece #2?

What quantity did you have to add to the x-coordinate of piece #1 to get the x-coordinate of piece #2?

What quantity, expressed symbolically, will you add to the x-coordinate of the current source location to get the x-coordinate of the next piece?

CHECKPOINT VP1: Have an instructor check your diagram and your plan, for credit.

6) ***Program organization. Your program will have 5 sections*:**

```
# constants
# initial values
# first loop to create objects (sphere) representing the pieces of the rod
# second loop to calculate net electric field at observation location,
      # due to all the point-like pieces of the rod
# finally, create appropriately scaled arrow to visualize net electric field
      # at observation location, print value of net field
```

(Note: This is a simple way to organize the program; if you are a skilled programmer, you are free to use more advanced features of Python to do this more elegantly).

6.1) # *constants*

Get the shell from the files page on the class website. The program starts with the usual two lines (remember the second line has 4 underscores). Replace ??? with the appropriate values.

```
from visual import *
from __future__ import division

# constants
L = ???                 ## see problem statement
N = 8                   ## to begin with; you will change it later
Q = ???                 ## see problem statement
oofpez = 9e9
scalefactor = 1.0       ## not the correct value; you will change it later
deltax = ???            ## use your symbolic expression from above
deltaQ = ???            ## use your symbolic expression from above
```

6.2) # *initial values*

Give the variable x the initial value of the x-coordinate of the center of piece 1, using your symbolic expression from above. This is the x-component of the position vector for the location of the center of piece 1.

```
x = ???
```

6.3) # *first loop to create spheres representing the pieces of the rod*

Now write a loop to create a red sphere, with radius deltax/2, at the center of each piece of the rod. You should end up with N red spheres in a line on the x-axis.

```
while x < ??? :
    sphere(pos = ??? ....)
    # now add the appropriate constant to x to get the position of
            # the next sphere
```

RUN THE PROGRAM. Does it display 8 spheres in a line on the x-axis?

SELF-CHECK: Change the value of N to 15. Does your program display 15 spheres, some to the left and some to the right of the origin?

CHECKPOINT VP2: Have an instructor check your running program, for credit.

6.4) # *second loop to calculate the net electric field at a single observation location*

In the # constants section, change N back to 8.

After the first loop that creates and displays the spheres, ***un-indent your code***.

- Give a value of < −1, 0.4, 0 > to "obslocation".
  ```
  Obslocation = ???
  ```
- Create a vector named Enet that starts off with zero values for each component:
  ```
  Enet = vector(0, 0, 0)
  ```
- Set "x" back to the value it had initially before you created the spheres (just copy the line you used before).
- Now write a second loop to do the following:
  1. loop through the locations of the source charges (the "point charges" represented by the spheres)
  2. for each point charge, calculate deltaE at the observation location due only to that "point charge" (***Note: because the spheres don't have names, you can't use*** sphere.pos ***here; to calculate*** r ***you must first create a vector that is the position of the center of the piece of interest. You might call the vector*** sourcelocation)
  3. add this contribution deltaE to the net field Enet:
     ```
     Enet = Enet + deltaE
     ```

6.5) # *displaying an arrow*

After the second loop, ***un-indent your code***.

- Print the value of Enet. Also print the magnitude of Enet.
- As a check, you could temporarily put the observation location on a perpendicular to the center of the rod and compare with the analytical solution for the rod, given in the textbook for that special case.
- Create an orange (color.orange) arrow at the observation location to represent Enet. Note, you will need to figure out a scale factor. Look at the printed value of Enet to figure this out.
-

SELF-CHECK: What is the value of Enet when N = 8? When N = 15? When N = 25? When N = 50? Do your results make sense?

CHECKPOINT VP3: Have an instructor check your running program, for credit.

7) ***Make sure you understand the organization and functioning of your program, because***:
When you submit your program in WebAssign, you will be asked to change some or all of the following quantities, and report on the results:

- Number of pieces into which the rod is divided
- Observation location
- Amount of charge on the rod
- Length of the rod
-

TURN IN YOUR PROGRAM TO WEBASSIGN

8) *Extra credit (optional*)
After saving your original program, make a copy of it, and give it a different name. Modify your program to calculate and display the electric field at 10 different locations along the rod, with the same y-value as the initial observation location.

To do this you can "nest" loops like this:

```
xobs =                 ## initial value
xobsfinal =            ## final value
delta_xobs =           ## increment
while xobs < xobsfinal:
    # copy your entire loop to calculate E and indent it here
    # you need to start with a zero E vector each time
            # calculate field, create arrow
    xobs = xobs + delta_xobs ## new observation location
```

You will be able to turn in this program in WebAssign too.