Electric Field of a Uniformly Charged Ring

Objectives

In Chapter 15, we found an analytical expression for the electric field along the axis of a uniformly charged ring. For a general observation location, we have to compute the electric field numerically.
After completing this activity you should be able to:

- Create and use lists of many source charges and many observation locations in VPython
- Calculate and display the net electric field of a macroscopic uniformly charged object
- Discuss the effect of varying the number of point charges used to model a charged ring
- Compute the motion of a charged particle near a macroscopic uniformly charged object

1 Creating a List of Source Charges

Study the following code fragment.

```
from __future__ import division, print_function
from visual import *
scene.width = 1000
scene.height = 700
R = 0.08
Q = 50e-9
Nq = 25
dtheta = 2*pi/Nq
dQ = Q/Nq
## source charges
ring(pos=vector(0,0,0), radius=R, color=color.red, thickness=0.005)
sources = []
angles=arange(0,2*pi,dtheta)
for theta in angles:
    a = sphere(pos=vector(0, R*cos(theta),R*sin(theta)), radius=0.007, color=color.cyan, q=dQ)
    sources.append(a)
```

⇒ What is the radius of the ring?
⇒ What is the net charge of the ring?
⇒ What plane does the ring lie?
⇒ How many charges are used to model the ring?

2 Observation Locations Along the Axis

⇒ Add a list of observation locations `obs` that consists of 17 arrows located along the ring axis, extending a distance equal to twice the ring radius in both directions from the center of the ring. Set the arrow lengths to 0.01. One of these arrows should be located at the center of the ring. Note that there will be 16 spaces between the 17 arrows.
⇒ Run your program. Do you see 17 arrows in appropriate locations? You may have to zoom out slightly.

3 Electric Field

Here is a code template that compute the electric field at each observation location.

```
## outer loop picks observation location
for Ea in obs:
    Enet = vector(0,0,0)
    ## inner loop goes through all source charges
```

```
    for scharge in sources:
        ## add code to calculate contribution of this source charge
        ## add this to net field at this location
    Ea.axis = Enet*scalefactor
```

⇒ Complete the code to compute the electric field along the axis. Check if the results are consistent with the analytical expression for the field on the axis.

Check your work before continuing.

4 Off-Axis Observation Locations

We do not have an analytical expression for the electric field of a charged ring at locations not on the axis, so we'll need to compute these values.

⇒ Add a line of arrows perpendicular to the axis, a distance $(R/2)$ from the ring, where R is the radius of the ring. One possible scheme is to position the arrows from $z = -2R$ to $z = 2R$, in increments of $2R/3$, with $x = R/2$ and $y = 0$. You can experiment with adding arrows at other locations.

5 Motion of an Electron

⇒ Use a sufficient number of charged particles in your model of the ring to give a good approximation for the field of the ring.

⇒ Position the electron at a location on the axis of the ring, a distance of $2R$ from the center of the ring, with zero initial momentum. Have it leave a trail. Try a value of Δt of around 1×10^{-11} s, and `rate(500)`.

⇒ To be able to see the electron, you may need to comment out the code that creates arrows on the axis (or, alternatively, decrease the opacity of the arrows so they are partly transparent).

5.1 Optional: Using the Mouse to Position the Electron

You may allow the user to position the electron by clicking a location on the display. To wait for a mouse click and get its location, then use that location to create the electron, do the following :

```
mouseposition = scene.waitfor('click').pos     ## wait for mouse click, save the mouse position
electron = sphere(pos=mouseposition, ...)      ## use mouse position when creating electron
```