Computing Electric Field Along the Axis of a Charged Ring With a Computer

This program computes the net electric field due to a uniformly charged ring of radius R and charge Q at a given point in space.

In [3]:
```
from __future__ import division, print_function
from vpython import *
from math import *
```

In [4]:
```
scene=canvas(title="Electric field due to uniformly charged ring")
scene.background=color.white

R=0.02 #radius of ring in m
Q=1e-9 #charge of ring in C
N=4 #number of unique pieces
oofpez=9e9 #1/(4pi epsilon_0) in N m^2/C^2
dq=Q/N #charge of a piece

#draw the objects
myring=ring(pos=vector(0,0,0), radius=R, axis=vector(0,0,1), color=co
lor.blue, thickness=0.02*R)
zaxis=cylinder(pos=-2*R*myring.axis, radius=0.015*R, axis=4*R*myring.
axis, color=color.black)
point=sphere(pos=R*myring.axis, color=color.red, radius=5*zaxis.radiu
s)


points =[]
rings = []


calc_points = input("How many points would you like to use to calcula
te the electric field?")
num_rings = input("How many rings would you like to use?")
spacing = input("How far apart would you like to rings to be? Choose
 a number between 0.01 and 0.05")


for i in range(calc_points):

    x = 0.01*sin(i*2*pi/calc_points)
    y = 0.01*cos(i*2*pi/calc_points)

    points.append(sphere(pos=vector(x,y,0.01), color=color.red, radiu
s=3*zaxis.radius))


charge =[]
```

```
for i in range(num_rings):

    loc = i - (num_rings)//2
    rings.append(ring(pos=vector(0,0,spacing*loc), radius=R, axis=myr
ing.axis, color=color.blue, thickness=0.02*R))
    charge.append(Q)



for a_point in points:

     for i in range(len(rings)):
        aring = rings[i] # look at one ring

        dq = charge[i]/N #charge of a piece
        Enet = vector(0,0,0)
        dtheta=2*pi/N #theta increment for our loop
        theta=dtheta/2 #initial theta for first piece of loop
        Ering = vector(0,0,0) #net electric field for single ring

        rpoint = a_point.pos


        scale=1.2*mag(rpoint)/8000 #used to scale the arrows represen
ting E-field

        while theta<2*pi:
            rpiece=R*vector(cos(theta),sin(theta),0) #location of pie
ce
            r=rpoint-rpiece #vector from piece to point in space
            rmag=mag(r) #magnitude of r
            rhat=norm(r) #unit vector for r
            dE=oofpez*dq/rmag/rmag*rhat #Electric field due to piece
 at rpoint
            Enet=Enet+dE #net electric field of the first one up to t
his one
            #particle=sphere(pos=rpiece, radius=point.radius, color=c
olor.yellow) #draw a particle at center of piece
            dEvector=arrow(pos=rpoint, axis=scale*dE/5, color=color.m
agenta, shaftwidth=point.radius/5)
            theta=theta+dtheta

       # print("The net electric field = ",Enet, "N/C")
        Evector=arrow(pos=rpoint, axis=scale*Enet, color=color.orange
, shaftwidth=point.radius/2)
```

```
How many points would you like to use to calculate the electric fiel
d?20
How many rings would you like to use?20
How far apart would you like to rings to be? Choose a number between
0.01 and 0.050.01
```

In [ ]:

I could not get Latex to work with Jupyter Notebook for the life of me so I had to screenshot my visual outputs.