Physics 332: Using VPython to visualize fields

In Physics 232, you used VPython to visualize electric and magnetic fields due to different charge and current configurations. You've now learned to use quiver plots to visualize fields, and those a great for static fields, but not so handy for visualizing time-evolving ones, so we return to VPython.

Say you have an analytical expression for the field as a function of position and time, $\vec{E}(r,t)$, and you want to visualize how it evolves over time at a number of locations.

To visualize the field at just one place and time, you'd probably fist create an arrow of zero length,

```
Earrow = arrow(pos=vector(x,y,z), axis=vector(0,0,0), color=color.cyan)
```

And then inside a time loop update the length and direction of the arrow's axis with

```
while t < tmax:
    Earrow.axis = E (some function of x,y,z, and t)
    t = t + dt
```

If you want to illustrate the field at several locations, you'd generally do the same things, but take advantage of VPython's letting you define an array of Earrows.

First, you'd create an empty array

```
Earrows = []
```

Then you'd populate the array

```
for x in linspace(xmin, xmax, number_of_x_steps):
    for y in linspace(ymin, ymax, number_of_y_steps):
        for z in linspace(zmin, zmax, number_of_z_steps):
            Earrow = arrow(pos=vector(x,y,z), axis=vector(0,0,0),
                                                color=color.cyan)
            Earrows.append(Earrow)
```

The nested for…in… lines step x, y, and z's values across the range you've selected. The append operation takes the new arrow that you've created and adds it to an array of arrows.

The for…in… structure can be used slightly differently inside you're while loop so you step through updating each arrow's axis.

```
While t< tmax:
    for Earrow in Earrows:
        Earrow.axis = E(some function of Earrow.x, Earrow.y, Earrow.z, t)
    t = t + dt
```

Here’s a program to find the vector potential of a wire who’s current was suddenly turned on at t = 0 (Griffith’s Example 10.1)

```
from visual import *
# initialize arrows
Aarrows = []
for x in linspace(-4,4,10):
    for y in linspace(-4,4,10):
        Aarrow=arrow(pos=vector(x,y,0), axis=vector(0,0,0), color = color.cyan)
        Aarrows.append(Aarrow)

scene.autoscale = 0
c=3e8
t = 0
dt = 1e-11
tmax = 1e-7
while t < tmax:
    rate(500)
    for Aarrow in Aarrows:
        s = sqrt(Aarrow.x**2+Aarrow.y**2)
        if s < c*t:
            Aarrow.axis = log((c*t+sqrt((c*t)**2 -s**2))/s)*vector(0,0,1)
    t = t + dt
```