Elizabeth Champagne

PHYS 331 H

Computational Project

Electric Field of a Charged Sphere with a small hole on the surface.

1. Consider a spherical shell of radius R centered on the origin of coordinates. The sphere is uniformly charged, with total charge Q, except for the region where θ ≤ 1.00˚. Consider field point on the positive z-axis. Determine 𝐸 ⃗ as a function of z.
2. Write a computer program to do the integral. Evaluate the electric field at points z = 0.01nR, where n is an integer. Do this from n = 0 to n = 500. Provide a printout of your code.
3. On the same graph, graph the function you obtained in part a) and the points you obtained in part b). Do not connect the points with a curve. Do both calculations give the same answer? If the graph is too messy, you may want to subdivide it into several graphs, or you may want to try just plotting a fraction of the points you calculated.
4. Comment on your calculation.
5. The standard equation for electric field is

(I will use to refer to spherical coordinates, and to refer to distances between charges).

This applies to two point charges, but because I am working with a continuous charge distribution I will use

In my system, the origin is at the center of the shell. I have source coordinates and field coordinates . Since I am only finding the electric field along the z-axis, I know that

and since all source coordinates are located a distance R from the origin I know that

The distance from the field coordinates to the source coordinates can be written:

Or, more usefully,

Because I know that the magnitudes of both and , as well as the angle between them, I can use the Law of Cosines to rewrite this as

Likewise, the unit vector can be written

Since the spherical unit vectors are not constant, I will eventually need to convert to Cartesian coordinates. However, it makes the numerator much more manageable if I do so now (the denominator can be rewritten in terms of the Law of Cosines again).

Given that

I now know that

Since I want to integrate with respect to space, rather than dQ, I will define the surface charge density . I will substitute σdA for dQ, and solve for A later to find the value of the surface charge density. I know that the spherical surface area element , where .

The full integral, with limits, is:

Pulling constants out:

Usually, I would integrate with respect to before but I know that the azimuthal symmetry means that there is no electric field component in the or directions. Thus, I know that integrating with respect to first means that I can cancel my and components, since both cosine and sine are 0 when evaluated from to . I now have:

With the use of Wolfram Alpha and my TI-89, this integral evaluates to

Simplifying the cosines above:

The only thing left is to solve for σ. This requires simply solving a surface area integral for .

Some of the R’s cancel:

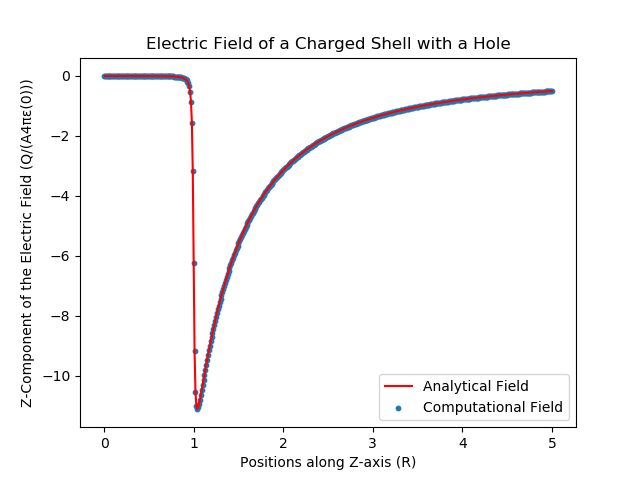
1. My program, ElectricField.py, and its output are appended at the end of this packet.

I originally wrote the code in C#, because I like the syntax better. However, I have never graphed in C# before, but I had some experience with Python’s MatPlotLib and NumPy, so to save myself some trouble I redid the code in Python so I could graph it. (The downside is that Python took considerably longer to do the calculations, though to be fair I made it calculate 1 million tiny vectors for each step on the z-axis.)

To simplify the output, my program computes the field in “units” of , where Q is an arbitrary charge, and A is a function of the arbitrary radius R (but A , due to the hole in the shell). Each line of output represents one 0.01R step along the z-axis. The output code is in Cartesian coordinates, formatted <x,y,z>. The computer produced extremely small values for the x- and y-components that can be dismissed as rounding error.

1. In order to graph my analytical solution on the same axes, I had to separate out my “units” of . My formula for graphing was:

The final graph:



(The code for this, ElectricFieldGrapher.py, is also appended. I separated the graphing program from the integrating program for the sake of not running unnecessary computations to redo the graph).

1. While I was still solving the problem, I came up with two checks to determine whether my final answer was reasonable.
2. The first is that at large z, the electric field should approximate the electric field of a point charge.
3. The second is that the strongest electric field should be near z = 1R. As z →1R, the magnitude of the electric field should increase from the near-zero field inside the shell. Once z passes R, the electric field will weaken as the distance from the charge grows.

The second test is easy to confirm by looking at my graph in part c.

The first test requires the limit of for z >> R.

So the second check works as well.