Computational exercise: The electric field of many point charges
PH 220

Objective: Write a computer code that reads in the locations and charges of several point charges, and then returns the components of the electric field at various points in space.

Background: When considering the dynamics of a charged particle in the presence of other charges, the first task is to find the electric field at any given point in space. For a single source charge, the magnitude of the electric field in Newtons is given by

$$F = kq/r^2$$

where q is the magnitude of the source's charge in Coulombs, r is the distance to the source charge in meters, and $k = 8.99 \times 10^9$ N m^2/C^2. The direction of the field is away from the source charge if it is positive, and toward the source charge if it is negative.

If multiple source charges are present, one finds the net electric field at any given point by finding the electric field vectors for each of the source charges independently, and then summing those individual electric field vectors.

How does one find the distance and direction between two points in Cartesian space? It's actually pretty straightforward. If the two points have coordinates (x_1, y_1, z_1) and (x_2, y_2, z_2), the vector drawn from the first point to the second point will have components

$$\vec{r} = (x_2 - x_1)\hat{i} + (y_2 - y_1)\hat{j} + (z_2 - z_1)\hat{k}$$

where $\hat{i}$, $\hat{j}$, and $\hat{k}$ are the usual Cartesian unit vectors. To find the length of this vector, we simply take the square root of the scalar product of the vector with itself, i.e.

$$r = \sqrt{\vec{r} \cdot \vec{r}} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

As for the direction, we can find a unit vector in the direction of the vector $\vec{r}$ by simply dividing $\vec{r}$ by its length r:

$$\hat{r} = \vec{r}/r = \left(\frac{x_2 - x_1}{r}\right)\hat{i} + \left(\frac{y_2 - y_1}{r}\right)\hat{j} + \left(\frac{z_2 - z_1}{r}\right)\hat{k}$$

Exercise requirements: Write a computer code that does the following:

1. Reads a file specifying the number of source charges, the charge and the coordinates of each of the source charges, the number of points at which to find the electric field, and the coordinates of those points. This input file will be formatted as follows. The first line

of the input will consist of a single integer specifying how many source charges there are. The next lines, one for each source charge, gives the charge in nC, the x-coordinate in meters, the y-coordinate in meters, and the z-coordinate in meters. The source charge lines are followed by another single integer specifying how many points we are finding the electric field for. This is followed by a line for each of those points, giving the x-coordinate in meters, the y-coordinate in meters, and the z-coordinate in meters. The following is a sample input file:

```
6
  -3.450   1.323  -0.458   0.498
   2.850  -1.685   2.254  -2.211
   0.330   3.844   0.003   1.112
  -5.440  -4.451  -3.239  -0.731
   3.690   3.332  -3.854  -2.074
   2.980   2.133  -2.298   3.287
4
   0.000   0.000   0.000
   1.000   0.000   2.000
   0.000   1.000   0.000
   0.000   1.000  -1.000
```

Reading and writing data from files is a very common operation in computational work, and one that you absolutely must learn. Unfortunately, the process for reading files varies from programming language to programming language, and consequently I cannot give any sort of universally applicable instruction here. If you do not already know how to do file I/O, I recommend that you turn to any of the many great tutorials found online.

2. Calculates the electric field components at each of the requested points.
3. Writes out the coordinates of the requested points, in order, and the resulting electric field components, formatted as follows:

```
  1: (  0.000,  0.000,  0.000) E=< 10.725, -5.089,  5.157> N/C
  2: (  1.000,  0.000,  2.000) E=<  0.087, -1.098,-11.816> N/C
  3: (  0.000,  1.000,  0.000) E=<  4.461, -6.044,  3.271> N/C
  4: (  0.000,  1.000, -1.000) E=<  3.873, -4.898,  4.815> N/C
```

Validation: You can verify whether your code is working correctly or not by feeding it the example input file above, and seeing whether it reproduces the example output.

Grading rubric: Your grade on this exercise, out of 50 points total, will be calculated based on the following criteria:

- Your code accurately finds the electric field for a test input file that I provide. This test file will include ten test coordinates, for a total of 30 electric field components. Each component is worth one point. (30 points)
- Your code is well organized and “pretty”. (3 points)
- The output is formatted as required. (2 points)
- You have included all of the following comments in your code:
    - A header that describes what the code does and how to use it. (5 points)
    - A description of each of the variables in your code. (5 points)
    - Frequent descriptions of what your code is doing, including any numerical methods that you are employing. (5 points)