Standard E3-1

I can identify what charge should be included in the electric field contribution at a specified point

An electric field is produced by a charged source. Should an electric field exist, and is measured an any arbitrary point, then the charge of the source can be identified by the following equation:

$$\vec{E} = \frac{q}{4\pi\epsilon_0 |r|^2} * \hat{r}$$

$$q = \frac{4\pi\epsilon_0 |r|^2}{\vec{E}} * \frac{1}{\hat{r}}$$

Where the electric field is known, and the specific point's location, or distance from the source, is also known.

Scenario:

An electric field of <56234, 1236431, 0.77642>C/m is measured at a point <0.04, 0.22230, 0.0002>m away from the source. What is the charge of this source?

In [6]:
```python
import numpy as np
import scipy.constants as const

# Constants
e0= const.epsilon_0
e = const.elementary_charge
k = 1/(4*np.pi*e0)
k_inverse = 4*np.pi*e0

def vectorize(final, initial):
    # Input variables are not vectorized. This function puts them in vector format,
    # The distance between the points, with the first entry as the final location.
    # Then will calculate the magnitude and unit vector for that distance
    # Returns the distance vector, magnitude, and unit vector

    f = np.array([final])
    i = np.array([initial])
    distance = f - i
    magnitude = np.linalg.norm(distance)
    hat = distance/magnitude
    return distance, magnitude, hat

def e_calc(charge, rmag, rhat):
    # Calculate electric field, EQN: (kq/|r|^2)*rhat

    e_field = (k*charge/rmag**2)*rhat
    return e_field
```

```
def charge_calc(e_field, rmag, rhat):
    # Use the same e field equation, but rearranged to solve for the charge

    charge = (k_inverse * rmag**2/e_field)*(1/rhat)
    return charge


# Inputting the coordinates of each point of interest.
origin = (0,0,0)                                # Origin Location
specific_point = [0.04, 0.22230, -0.0002]       # Specific Point distancce from cha
e_field = [56234, -1236431, 0.77642]            # Measured E field at specific poin


# Creating the r vectors
r_vec, r_mag, r_hat = vectorize(specific_point, origin)

# Calculating the charge
charge = charge_calc(e_field, r_mag, r_hat)

print(charge)
```

```
[[ 5.70001784e-16 -4.66472274e-18 -8.25673742e-09]]
```

In [ ]: