Exercise 1: Visualize electric field

Here we are going to visualize the electric field from several particles in 2D. We are going to do this in parts.

a) Consider a particle in (x_1, y_1) with charge Q, write down the general electric field in position (x, y).

Solution. The vector from the particle and the position you want to evaluate is given by $\mathbf{R} = [x - x_1, y - y_1]$. We can then write the general electric field:

$$\mathbf{E} = \frac{Q}{4\pi\epsilon_0\big((x-x_1)^2 + (y-y_1)^2\big)^{3/2}}[x - x_1, y - y_1] \tag{1}$$

b) Make a program that takes in position and charge from a particle, a point in space, and returns the electric field vector in that point. Make a vector arrow plot.

Hint 1. It can be a good idea to scale ϵ_0 so you get reasonable sizes.

Hint 2. Use quiver from matplotlib.pyplot to visualize the field.

Solution.

```
import numpy as np
import matplotlib.pyplot as plt
eps0 = 1 #scaled to get reasonable sizes
def Efield(pos, Q, x, y):
    """
    This function takes in the position and charge of a particle, the position
    you want to calculate the field, and returns the x- and y-coordinate of
    the field.
    """
    r_eval = np.array([x-pos[0], y-pos[1]])
    R_eval = np.linalg.norm(r_eval, axis=0)
    Field = Q/(4*np.pi*eps0*R_eval**3)*r_eval
    return Field[0], Field[1]
#The area we are going to be looking at
x = np.linspace(-10,10,20)
y = x
X, Y = np.meshgrid(x,y)
#The position and charge of the particle
pos1 = np.array([1, 0])
Q1 = 1
#Calculating the electric field
U, V = Efield(pos1, Q1, X, Y)
#Plotting
```

```
fig, ax = plt.subplots()
ax.quiver(X, Y, U, V)
ax.plot(pos1[0], pos1[1], 'or')
plt.show()
```

c) Expand the function you made to take in an arbitrary number of particles and then returns the resulting electric field. Make a vector arrow plot.

Solution.

```
eps0 = 1 #scaled to get reasonable sizes
def Efield_expanded(positions, charges, x, y):
    """
    This function takes in positions and charges, positions we want to evaluate,
    and returns the resulting electric field.
    It also returns a list with colors that corresponds to the charges of the particles.
    """
    Field = 0
    color = []
    for pos, Q in zip(positions, charges):
        r_eval = np.array([x-pos[0], y-pos[1]])
        R_eval = np.linalg.norm(r_eval, axis=0)
        Field += Q/(4*np.pi*eps0*R_eval**3)*r_eval
        if Q > 0:
            color.append('r')
        else:
            color.append('b')
    return Field[0], Field[1], color
#Defining positions and charges
pos1 = np.array([5, 5])
Q1 = 1
pos2 = np.array([0, 0])
Q2 = -1
pos3 = np.array([-2, -2])
Q3 = -3
positions = [pos1, pos2, pos3]
charges = [Q1, Q2, Q3]
#The area we are going to be looking at
x = np.linspace(-10,10,40)
y = x
X, Y = np.meshgrid(x, y)
#calculating the electric field
U, V, colors = Efield_expanded(positions, charges, X, Y)
#plotting
fig, ax = plt.subplots()
for col, pos in zip(colors, positions):
```

```
    ax.plot(pos[0], pos[1], 'o', color=col)
ax.quiver(X, Y, U, V)
plt.show()
```