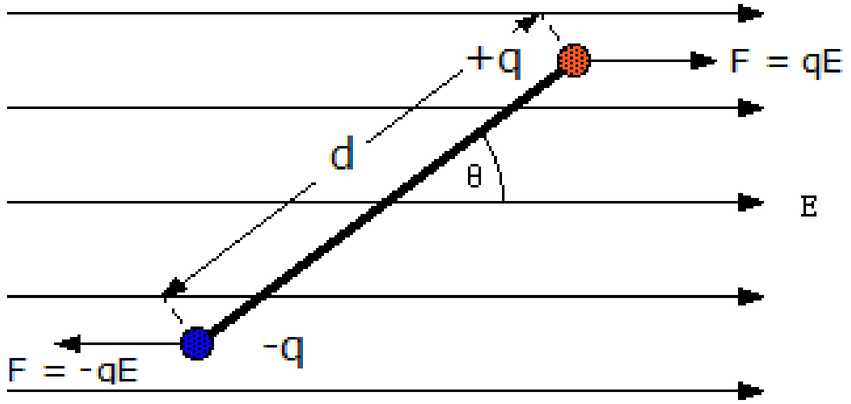
Exercise 1: Dipole in a uniform electric field

(Made by: Sigurd Sørlie Rustad)
In this exercise we will take a look at an electric dipole in a uniform electric field. The distance between them is d (see figure under).



a) What is the net force on the dipole?

Solution. $\Sigma \mathbf{F} = 0$

b) Find the torque in terms of the electric moment $\mathbf{p}$ and the electric field $\mathbf{E}$.

Hint. The formula for torque is $\boldsymbol{\tau} = \mathbf{r} \times \mathbf{F}$, where $\mathbf{r}$ is the the vector going from the point of rotation to where the force $\mathbf{F}$ is applied.

Solution. Using the formula for torque and some cross product rules we get

$$\boldsymbol{\tau} = \mathbf{r} \times \mathbf{F} \ = \left(\frac{\mathbf{d}}{2} \times q\mathbf{E}\right) + \left(-\frac{\mathbf{d}}{2} \times (-q\mathbf{E})\right) = q\mathbf{d} \times \mathbf{E} = \mathbf{p} \times \mathbf{E} \tag{1}$$

c) For what angles θ do we have stable and unstable equilibrium? And when do we have maximum torque?

Solution. Stable equilibrium: $\theta = 0$, Unstable equilibrium: $\theta = \pi$, maximum torque: $\theta = \frac{\pi}{2} \quad \wedge \quad \theta = \frac{3\pi}{2}$

d) Show that for small values of θ you get a simple harmonic motion. Here you need to use the small angle approximation ($sin(\theta) \approx \theta$) and $\boldsymbol{\tau} = I\boldsymbol{\alpha}$, where I is the moment of inertia and $\boldsymbol{\alpha}$ is the angular acceleration.

Hint. A simple harmonic oscillator has the differential equation

$$m\frac{d^2x}{dt^2} = -kx \tag{2}$$

Where m and k are constant and x describes the harmonic motion. The solution is

$$x(t) = Acos(\omega t + \phi) \qquad \omega = \sqrt{\frac{k}{m}} \tag{3}$$

A and ϕ are constant.

Solution. Looking at the torque:

$$\boldsymbol{\tau} = \mathbf{p} \times \mathbf{E} \implies \tau = -|\mathbf{p}||\mathbf{E}|sin(\theta) = I\alpha \tag{4}$$

Using the small angle approcimation we get

$$I\alpha \approx -|\mathbf{p}||\mathbf{E}|\theta \tag{5}$$

This gives us the differential equation

$$I\frac{d^2\theta}{dt^2} = -|\mathbf{p}||\mathbf{E}|\theta \tag{6}$$

This is an equation for a simple harmonic oscillator.

e) What is the angular velocity?

Solution.

$$\omega = \sqrt{\frac{|\mathbf{p}||\mathbf{E}|}{I}} \tag{7}$$

f) Find the formula for θ.

Solution. We are looking for a solution like

$$\theta(t) = Acos(\omega t + \phi) \tag{8}$$

In order for energy to be conserved $A = \theta_0$ where θ_0 is the maximum value of θ. ω is known and ϕ depends on the initial conditions, lets say the θ peaks at $t = 0$, then $\phi = 0$. The angle θ can therefore be described by the equation

$$\theta(t) = \theta_0 cos\left(\sqrt{\frac{|\mathbf{p}||\mathbf{E}|}{I}}t\right) \tag{9}$$

g) Plot the angle over two periods.

Hint. The formula for moment of inertia for point mass

$$I = \sum_i m_i {r_i}^2 \tag{10}$$

m_i is the mass and r_i the distance from the point of rotation to the point mass.

Solution.

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.constants as const
theta0 = (10.0/180)*np.pi   # initial displacement in radians
q = const.e                 # charge on dipole in Coulombs
d = 1e-20                   # distance separating the charges (m)
m = const.m_e               # mass of a single charge (kg)
E = 1e-3                    # magnitude of the electric field
I = 2*m*(d/2)**2            # moment of inertia
p = q*d                     # electric moment
omega = np.sqrt(p*E/I)      # angular frequency
T = 2*np.pi/omega           # period
t = np.linspace(0, 2*T, 2000)  # time interval from t = 0 to t = 2T
theta = theta0*np.cos(omega*t) # solution for Simple harmonic motion
#Plot
plt.plot(t, theta)
plt.title("Simple Harmonic Motion - dipole in an Electric field")
plt.x\label("time (s)")
plt.y\label("angle (radians)")
plt.grid()
plt.show()
```

h) Find the numeric solution (do not use small angle approximation) and plot both results over four periods. Use SciPy's odeint package to solve the differential equation. The initial conditions are $q = e$, $d = 3 \times 10^{-9}$m, $m = 3 \times 10^{-5}$kg and $E = 1000$N/C.

Solution.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
theta = np.pi/6         # intial angle of displacement (degres)
q = 1.0e-19             # Charge on dipole in Coulombs
d = 3.0e-9              # distance separating the charges (m)
p = q*d                 # electric moment
m = 3.0e-5              # mass of a single charge (kg)
E = 1000                # magnitude of the electric field
```

```
I = 2*m*(d/2)**2        # moment of inertia
omega = np.sqrt(p*E/I)  #angle velocity
T = 2*np.pi/omega       # period
# number of time intervals
N = int(1e4)
# set the times to evaluate
t = np.linspace(0, 4*T, N)
# compute approximate theta values on the time intervals
Approx_Solution = theta*np.cos(omega*np.array(t))
def torque(x0, t, p, I, E):
    dfdt = np.zeros(2)
    dfdt[0] = x0[1]
    dfdt[1] = -p*E*np.sin(x0[0])/I
    return dfdt
# Set the intial condition [initial angle, initial velocity]
x0 = [theta, 0.0]
# numerically solve the real solution using odeint
sol2 = odeint(torque, x0, t, args=(p, I, E))
# Plot the two solutions
plt.plot(t, Approx_Solution)
plt.plot(t, sol2[:, 0])
plt.grid()
plt.x\label("time (s)")
plt.y\label("angle (rads)")
plt.show()
```

i) Expand the code over and plot the difference between the numeric and approximate solution over time. Find where the difference between them is above 0.05 rad. Plot red dots over the numeric solution where that is the case.

Solution.

```
# compute the difference between approx and actual on intervals
diff = np.abs(Approx_Solution-sol2[:,0])
# Plot the difference vs time
plt.plot(t, diff)
plt.x\label("time (s)")
plt.y\label("Difference")
plt.show()
#finding where the error is bigger than 0.05 rad
index = np.where(diff>0.05)
# Plot the red dots
plt.plot(t, sol2[:,0], \label='numeric solution')
plt.scatter(t[index], sol2[index,0], color='r', \label='error>0.01 rad')
plt.x\label("time (s)")
plt.y\label("angle (rads)")
plt.legend(loc=1)
plt.grid()
plt.show()
```