

Exercise 1: Visualize the field from a dipole

(Made by Sigurd Sørli Rustad)

In this exercise we are going to visualize the field from a magnetic dipole with a streamplot. The field from a dipole is given by the equation

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \left(\frac{3\mathbf{r}(\mathbf{m} \cdot \mathbf{r})}{r^5} - \frac{\mathbf{m}}{r^3} \right) \quad (1)$$

Make a function that takes magnetic moment \mathbf{m} , its location and the position \mathbf{r} you want to evaluate the field. The output should be the resulting magnetic field. You only need to do it in 2D. Make a streamplot to visualize the field.

Extra challenge: Try to vectorize your code. If you do it properly it will be compatible with both 2D and 3D. NumPy has several good packages that you can use to vectorize your code. The ones used in the solution are `tensordot` to dotproduct along an axis and `linalg.norm` to find the length of several vectors along an axis.

Solution.

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.constants as const
mu0 = const.mu_0
def dipole(m, r, r0):
    """
    This function take the magnetic dipole m, with position r0,
    and returnsthe magnetic field in postion r.
    """
    #np.subtract allows r0 to be a list and not an array
    R = np.subtract(np.transpose(r), r0).T
    #Finding the length of the vector
    norm_R = np.linalg.norm(R, axis=0)
    #Tensordot allows us to dotproduct along an axis
    m_dot_R = np.tensordot(m, R, axes=1)
    #calculatig the magnetic field
    B = 3*m_dot_R*R/norm_R**5 - np.tensordot(m, 1/norm_R**3, axes=0)
    #multiplying with the physical constant
    B *= mu0/(4*np.pi)
    return B

xa = np.linspace(-1, 1)
ya = np.linspace(-1, 1)
Bx, By = dipole(m = [0, 0.1], r = np.meshgrid(xa, ya),\
    r0=[-0.2, 0.1])
plt.streamplot(xa, ya, Bx, By)
plt.show()
```