

On considère une distribution de deux charges ponctuelles $+q(-a, 0)$ et $-q(a, 0)$. On souhaite tracer un réseau de lignes de champ avec le cahier des charges suivants :

- ✓ Chaque point du tracé est distant d'une valeur notée *pas*
- ✓ Le point initial est pris à une distance *pas* de la charge $+q$, avec un angle θ par rapport à l'axe OX .
- ✓ On arrête le tracé lorsque le dernier point calculé se trouve à une distance inférieure à $1,5.pas$ de la charge $-q$

Voici le début du code Python permettant ce tracé :

```
# -*- coding: utf-8 -*-
import matplotlib.pyplot as plt
from numpy as np

# Constantes
global epsilon0 ,K,q,pas
epsilon0=10**(-9)/(36*np.pi)
K=1/(4*np.pi*epsilon0);
q=1.6*10**(-19)
pas=0.02

def champ_ponctuel(charge , source , point ) :
    E=[]
    distance=((point [0] - source [0] )**2+(point [1] - source [1] )**2)**(1/2)
    Ex=K*charge *(point [0] - source [0] ) / distance**3
    E.append(Ex)
    Ey=K*charge *(point [1] - source [1] ) / distance**3
    E.append(Ey)
    return E
```

1. Proposer une fonction $champ_total(point)$ admettant en argument une liste comportant les coordonnées du point où calculer le champ et retournant une liste dont le premier élément est la composante selon OX du champ, le deuxième la composante selon OY et le troisième la norme.
2. On note $\vec{dl} = \overrightarrow{M_k M_{k+1}}$ avec M_k et M_{k+1} deux points successifs pour le tracé de la ligne de champ. Comment relier $\vec{E}(M_k)$, \vec{dl} et $|\vec{E}(M_k)|$?
3. Proposer une fonction $ligne_champ(theta)$ retournant le tracé de la ligne de champ correspondante. On limitera par sécurité à 1000 le nombre d'itérations de la boucle utilisée.
4. Tracer un réseau de lignes de champ pour la distribution.