

## 3 Cartographie d'un champ électrostatique

### Script à compléter

#### Fichiers Python

Script à compléter  
Fiche d'accompagnement  
[hatier-clic.fr/pc1200](http://hatier-clic.fr/pc1200)

L'objectif de cette activité est de cartographier un champ électrostatique créé par une particule chargée puis par deux particules chargées. On ne cherche pas à construire les lignes de champ, mais uniquement à tracer les directions et sens du champ électrostatique en des positions régulièrement choisies.

#### Prérequis théoriques

- L'expression vectorielle du champ électrostatique créé par une particule chargée, ainsi que l'utilisation de vecteurs unitaire pour le décrire.
- Le lien entre le champ électrostatique en un point et la force ressentie par une particule chargée placée en ce point.

#### Apports théoriques supplémentaires de l'activité

- Le principe de superposition est ici utilisé, sans être nommé. S'il est nécessaire à cette activité, il ne constitue pas une attente en termes d'acquisition.

La question 1a nécessite d'observer les sens des vecteurs de la cartographie, pour déterminer le signe de la charge portée par la particule qui engendre ce champ électrostatique.

La question 1b doit être abordée en se remémorant la formule mathématique donnant la norme du champ électrostatique créé par une particule chargée. On doit alors réfléchir sur les différentes échelles de distances que l'on pourrait rencontrer et ainsi sur la norme du champ électrostatique que l'on obtiendrait en faisant les calculs manuellement. On comprendra alors l'intérêt de ne pas représenter des vecteurs dont la longueur serait proportionnelle à la norme du champ électrostatique en ce point.

La question 1c doit être abordée en se remémorant le lien entre la force subie par une particule chargée en un point dans un champ électrostatique et le champ électrostatique en ce point. Il faut répondre en utilisant une formulation permettant de donner une réponse globale indépendamment de la position choisie pour le raisonnement.

La question 2a nécessite les modifications du programme suivantes (en suivant le modèle du traitement fait pour la génération du premier champ électrostatique) :

- la création des constantes associées au deuxième champ électrostatique (sa charge «  $q_2$  » et ses coordonnées «  $x_2$  » et «  $y_2$  ») (lignes 31 à 34) ;
- la création d'une matrice «  $r_2$  » correspondant à la distance séparant le point courant et de la source de ce deuxième champ électrostatique (lignes 38 et 39) ;
- la création d'une matrice «  $E_2$  » correspondant à la coordonnée radiale de ce deuxième champ électrostatique (lignes 43 et 44) ;
- la création de deux matrices «  $u_{x_2}$  » et «  $u_{y_2}$  » correspondant aux coordonnées du vecteur unitaire portant le deuxième champ électrostatique au point courant (lignes 49 à 51) ;
- la création de deux matrices «  $E_{x_2}$  » et «  $E_{y_2}$  » correspondant aux coordonnées du deuxième champ électrostatique au point courant (lignes 56 à 58).

Ensuite il reste à exécuter le programme pour obtenir la carte du champ électrostatique total ainsi obtenu.

La question 2b doit être abordée en se remémorant une nouvelle fois le lien entre la force subie par une particule chargée en un point dans un champ électrostatique et le

champ électrostatique en ce point. Il faut caractériser cette force au point spécifié par l'énoncé en observant le champ électrostatique produit en ce point sur la carte de champ obtenue.

La question 2c nécessite de suivre la même démarche qu'à la question précédente.

## Script à compléter

```

1  ### Création de cartes de champ électrostatique
2  ### créé par une ou deux particules chargées
3  ### (Seules la direction et le sens sont affichés)
4
5  import numpy as np
6  import matplotlib.pyplot as plt
7
8  ### Constantes du problème
9  xmin, ymin, xmax, ymax=-2, -2, 2.125, 2.125 # Taille de la zone en m
10 h=0.25 # Pas de la zone en m
11 K=8.99E9 # Constante de Coulomb en USI
12 e=1.60E-19 # Charge élémentaire en C
13
14 ### Coordonnées (x,y) des points de calcul
15 XX=np.arange(xmin, xmax, h)
16 YY=np.arange(ymin, ymax, h)
17 x, y=np.meshgrid(XX, YY)
18
19 ### Initialisation
20 Ex2=None
21
22 #####
23 ##### À compléter : question 2 #####
24 ### Champ créé par deux charges ###
25 #####
26
27 ### Particule 1
28 q1=-e # Charge électrique
29 x1=-1 # Abscisse en m
30 y1=0 # Ordonnée en m
31
32
33
34
35
36 ### Distance particule 1-point (x,y)
37 r1=((x-x1)**2+(y-y1)**2)**0.5
38
39
40
41 ### Coordonnée radiale champ E1
42 E1=K*q1/r1**2
43
44
45

```

### Modules

Le module numpy est nécessaire la création des matrices de coordonnées et indirectement la génération des matrices de champ.

Le module matplotlib permet de tracer les graphiques.

### Constantes du problème

On définit la cellule de travail, mais aussi les constantes de physique intervenant dans le programme.

On crée alors des matrices contenant les coordonnées des positions auxquelles les directions et les sens du champ électrostatique seront tracés.

### Partie à compléter

C'est dans cette partie que l'on ajoute la contribution de la deuxième particule source au programme.

Dans cette partie, les constantes de cette deuxième particule doivent être définies (charge et position). On doit ensuite créer les coordonnées du deuxième champ.

Pour plus de clarté, on commence par calculer les distances entre point courant et la position de la source. Puis on génère la coordonnée radiale du champ.

```

46  ### Coordonnées vecteur unitaire l
47  ux1=(x-x1)/r1
48  uy1=(y-y1)/r1
49
50
51
52
53  ### Coordonnées cartésiennes champ E1
54  Ex1=E1*ux1
55  Ey1=E1*uy1
56
57
58
59
60  #####
61  ### Fin de la partie à compléter ###
62  #####
63
64  ### Coordonnées du champ E
65  if Ex2 is None:
66      Ex=Ex1
67      Ey=Ey1
68  else:
69      Ex=Ex1+Ex2
70      Ey=Ey1+Ey2
71
72  ### Norme des vecteurs E
73  E=(Ex**2+Ey**2)**0.5
74
75  ### vecteur E normalisé
76  ### (pour avoir des flèches de taille identique)
77  Ex=Ex/E
78  Ey=Ey/E
79
80  ### Positions des différentes particules sources
81  if Ex2 is None:
82      X_particules=np.array([x1])
83      Y_particules=np.array([y1])
84  else:
85      X_particules=np.array([x1,x2])
86      Y_particules=np.array([y1,y2])
87
88  ### Tracé des vecteurs normalisés
89  plt.rcParams['axes.formatter.use_locale']=True
90  plt.figure(1, figsize=(5, 5))
91  plt.quiver(x,y,Ex,Ey)
92  plt.plot(X_particules,Y_particules,"r.",label="Particule(s)")
93  plt.legend(loc='upper right')
94  plt.xlabel("x (en m)")
95  plt.ylabel("y (en m)")
96  plt.title("Directions et sens du champ électrostatique")
97  plt.show()

```

### Partie à compléter (suite)

Enfin, après avoir généré les coordonnées du vecteur unitaire en tout point, on génère les coordonnées du champ électrostatique.

### Théorème de superposition

Dans cette partie du programme, on calcule le champ total en ajoutant les coordonnées de chaque champ électrostatique créé par chaque particule.

Si seul le premier champ est créé, alors le champ total correspond à ce seul champ E1. On calcule alors la norme en chaque point du champ total ainsi calculé.

### Champ normalisé

Pour des raisons de clarté d'affichage de la carte de champ, on trace des vecteurs champ total qui ont, en tout point, la même norme (une norme unité). On dit que le champ est normalisé.

### Tracé de la carte de champ

Après avoir défini des matrices contenant les coordonnées des sources de champ, on effectue le tracé des vecteurs champ électrostatique normalisé en tout point, et on ajoute au graphique les points correspondant aux positions des sources. L'instruction quiver permet de tracer les vecteurs.