Électromagnétisme – Chapitre 2

TP : Tracé de surfaces équipotentielles et de lignes de champ

I. Description du problème

On souhaite réaliser le tracé de surfaces équipotentielles et de lignes de champ d'un système électrostatique en deux dimensions. Hormis quelques situations simples, le calcul du potentiel électrostatique $V(x,y)$ et du champ électrostatique $\overrightarrow{E}(x,y)$ en tout point de l'espace peut s'avérer fastidieux. Nous allons les calculer numériquement sous Python, afin de réaliser ensuite les tracés correspondants.

II. Détermination des surfaces équipotentielles et des lignes de champ

II.1. Maillage de l'espace

Nous allons réaliser un maillage de l'espace en deux dimensions en considérant une grille de taille $L \times L$ constituée de $N \times N$ points régulièrement espacés avec un pas spatial $\delta x = \dfrac{L}{N-1}$.

Le potentiel électrostatique $V(x,y)$ sera alors approximé par une matrice `numpy` $V[i,j]$ associée à des matrices `numpy` $X[i,j]$ et $Y[i,j]$ telles que :

- $X[i,j]$ renvoie l'abscisse x du point considéré ;
- $Y[i,j]$ renvoie l'ordonnée y du point considéré.

De même, le champ électrostatique $\overrightarrow{E}(x,y) = E_x(x,y)\,\vec{e}_x + E_y(x,y)\,\vec{e}_y$ sera approximé par deux matrices `numpy` $E_x[i,j]$ et $E_y[i,j]$ associées aux matrices $X[i,j]$ et $Y[i,j]$.

II.2. Potentiel électrostatique et surfaces équipotentielles

On rappelle que le potentiel électrostatique créé en tout point M de l'espace par une charge ponctuelle q située en A s'exprime :

$$V(M) = \frac{q}{4\,\pi\,\varepsilon_0\,AM} + V_\infty$$

avec $\varepsilon_0 = 8{,}854.10^{-12}\,\mathrm{F\cdot m^{-1}}$ la permittivité diélectrique du vide, et V_∞ une constante que l'on prendra nulle pour la suite. En appliquant le principe de superposition, on en déduit l'expression du potentiel électrostatique créé par un ensemble de charges q_k situées en A_k :

$$V(M) = \sum_k V_k(M) = \sum_k \frac{q_k}{4\,\pi\,\varepsilon_0\,A_kM}.$$

Le calcul du potentiel électrostatique en chaque point de la grille sera alors réalisé à l'aide de la formule :

$$V[i,j] = \sum_k \frac{q_k}{4\,\pi\,\varepsilon_0\,A_kM}$$

avec $A_kM = \sqrt{(X[i,j]-x_k)^2 + (Y[i,j]-y_k)^2}$.

La détermination de lignes continues de l'espace sur lesquelles le potentiel électrostatique est uniforme permet finalement le tracé de surfaces équipotentielles en deux dimensions.

II.3. Champ électrostatique et lignes de champ

Le champ électrostatique se déduit du potentiel électrostatique à l'aide de la relation :

$$\vec{E}(M) = -\overrightarrow{\text{grad}}\, V(M).$$

Il est ainsi possible de calculer le champ électrostatique en chaque point de la grille à l'aide de la méthode d'Euler :

$$\begin{cases} E_x[i,j] = -\dfrac{V[i,j+1] - V[i,j]}{\delta x} \\ \\ E_y[i,j] = -\dfrac{V[i+1,j] - V[i,j]}{\delta x} \end{cases}$$

comme l'abscisse est associée à la variable j et l'ordonnée est associée à la variable i. Cependant, cette méthode est contraignante à mettre en œuvre sur les bords de la grille, car le potentiel en ces points n'est pas fixé. On préfèrera alors utiliser la fonction `gradient` de la bibliothèque `numpy` qui gère automatiquement ces effets de bords.

La détermination de lignes tangentes en chacun de ses points au champ électrostatique permet finalement le tracé de lignes de champ.

III. Algorithme de tracé

III.1. Charge ponctuelle

On considère une charge ponctuelle $q = 1{,}6.10^{-19}\,\text{C}$ située en O. On souhaite observer les surfaces équipotentielles et les lignes de champ autour de cette charge dans un espace de coté $L = 1\,\text{nm}$ constitué de $N = 100\,\text{points}$.

Application :

En utilisant les bibliothèques `numpy as np` et `matplotlib.pyplot as plt`, écrivons un programme Python permettant de tracer les surfaces équipotentielles et les lignes de champ en réalisant les étapes suivantes :

- définir les paramètres ε_0, L, N et q ;
- créer les matrices $X[i,j]$ et $Y[i,j]$ à l'aide de la fonction `np.meshgrid` prenant en entrée deux listes correspondantes aux abscisses et ordonnées des points de la grille :

```
axeX = np.linspace(-L/2,L/2,N)
axeY = np.linspace(-L/2,L/2,N)
(X,Y) = np.meshgrid(axeX,axeY)
```

- créer la matrice $V[i,j]$, initialement remplie de 0, à l'aide de la fonction `np.zeros` :

```
V = np.zeros((N,N))
```

- calculer le potentiel $V[i,j]$ en chaque point de la grille ;
- calculer les champs $E_x[i,j]$ et $E_y[i,j]$ à l'aide de la fonction `np.gradient` (attention à l'ordre des variables) :

```
(Ey,Ex) = np.gradient(-V)
```

- ouvrir une figure ;
- tracer des surfaces équipotentielles à l'aide de la fonction `plt.contour` (une centaine par exemple) et des lignes de champ à l'aide de la fonction `plt.streamplot` :

```
plt.contour(X,Y,V,100)
plt.streamplot(X,Y,Ex,Ey)
```

- afficher la figure.

III.2. Molécule de chlorure d'hydrogène

On considère une molécule de chlorure d'hydrogène HCl. La différence d'électronégativité entre les deux atomes séparés d'une distance $a = 127\,\text{pm}$ entraine l'apparition d'une charge partielle $+\delta q = 2{,}9.10^{-20}\,\text{C}$ portée par l'atome d'hydrogène, et d'une charge partielle $-\delta q$ portée par l'atome de chlore. On modélise alors le système par :

- une charge ponctuelle $+\delta q$ située en $\left(0, \frac{a}{2}\right)$;
- une charge ponctuelle $-\delta q$ située en $\left(0, -\frac{a}{2}\right)$.

On souhaite observer les surfaces équipotentielles et les lignes de champ autour de cette molécule dans un espace de coté $L = 1\,\text{nm}$ constitué de $N = 100\,\text{points}$.

Application :

> En utilisant les bibliothèques `numpy as np` et `matplotlib.pyplot as plt`, écrivons un programme Python permettant de tracer les surfaces équipotentielles et les lignes de champ.

III.3. Condensateur plan

On considère un condensateur plan constitué de deux armatures de largeur $\ell = 7\,\text{mm}$ séparées d'une distance $d = 3\,\text{mm}$, et qui portent une charge $\pm Q = \pm 1.10^{-13}\,\text{C}$. On modélise alors le système par :

- un ensemble de 20 charges ponctuelles $+\frac{Q}{20}$ régulièrement espacées entre $\left(-\frac{\ell}{2}, \frac{d}{2}\right)$ et $\left(\frac{\ell}{2}, \frac{d}{2}\right)$;
- un ensemble de 20 charges ponctuelles $-\frac{Q}{20}$ régulièrement espacées entre $\left(-\frac{\ell}{2}, -\frac{d}{2}\right)$ et $\left(\frac{\ell}{2}, -\frac{d}{2}\right)$.

On souhaite observer les surfaces équipotentielles et les lignes de champ autour de ce condensateur dans un espace de coté $L = 1\,\text{cm}$ constitué de $N = 100\,\text{points}$.

Application :

> En utilisant les bibliothèques `numpy as np` et `matplotlib.pyplot as plt`, écrivons un programme Python permettant de tracer les surfaces équipotentielles et les lignes de champ.