

Tracé de lignes de champ et d'équipotentiels

L'objectif est de réaliser des tracés de lignes de champ électrique \vec{E} et du potentiel associé V de manière numérique grâce à un programme *Python*.

1 Conditions de l'étude et calcul numérique

1.1 Présentation générale

On travaille dans le vide. On se restreindra à un carré dans un plan où l'on placera des charges ponctuelles. L'étude pour être quelque peu réaliste s'effectuera avec des charges q qui seront de l'ordre de grandeur - pour leur valeur absolue - de la charge élémentaire $e = 1,6 \times 10^{-19} \text{ C}$ et pour un carré de côté $L = 1 \text{ nm} = 10^{-9} \text{ m}$ qui est l'ordre de grandeur des longueurs pertinentes au niveau microscopique. On placera des charges ponctuelles dans le plan qui se présente comme sur la figure 1.

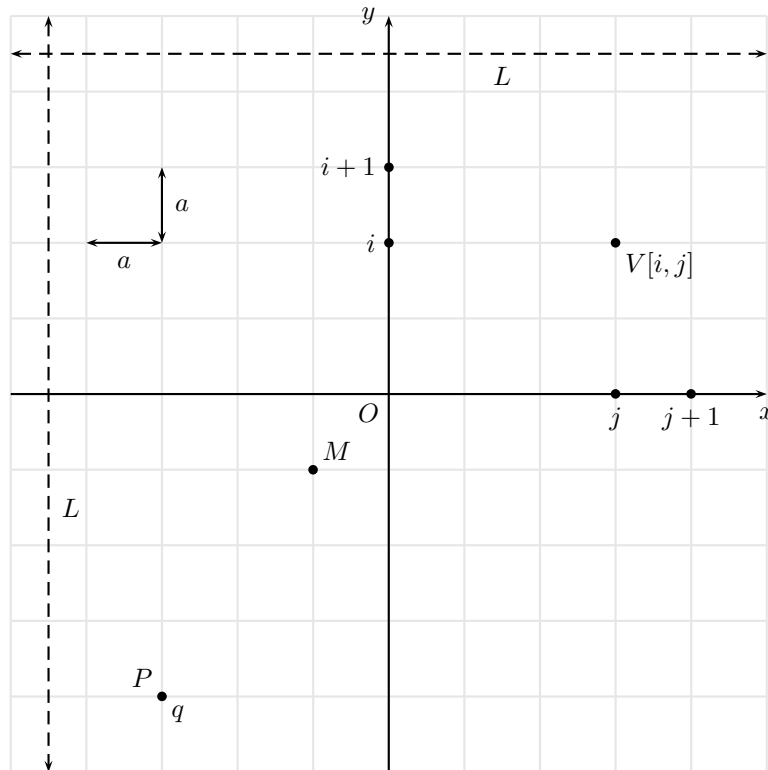


FIGURE 1 – Le carré délimitant l'étude des lignes de champ et le maillage associé

Le maillage dans le carré correspond à une matrice dans le programme *Python*. Il est réalisé par la fonction *meshgrid* de *numpy*. Attention, un point de coordonnées $[i, j]$ correspond à la ligne i et à la colonne j puisque le premier attribut correspond au numéro de la ligne et le second au numéro de la colonne. Ainsi, si l'on veut se déplacer d'une case selon l'axe Ox depuis $M[i, j]$ d'un pas, il faut considérer le point $[i, j + 1]$. De la même façon, pour se déplacer d'une case dans le sens de l'axe Oy , il faut considérer le point $[i + 1, j]$.

1.2 Calcul du potentiel et tracé des équipotentiels

On utilise la formule du potentiel consécutive à la loi de COULOMB donnant le champ créé par une charge ponctuelle à savoir $V(M) = \frac{q}{4\pi\epsilon_0} \frac{1}{PM}$ ce qui donne en utilisant les coordonnées des deux points :

$$V(M) = \frac{q}{4\pi\epsilon_0} \frac{1}{\sqrt{(x_M - x_P)^2 + (y_M - y_P)^2}}$$

On parcourt l'ensemble de la grille par des boucles *for* sur i et j pour calculer à chaque fois l'expression de $V[i, j]$. Ceci est réalisé pour chaque charge de la distribution que l'on a créée et on applique le théorème de

superposition en sommant au point $[i, j]$ chacun des potentiels créés par chaque charge en utilisant encore une boucle *for* sur les charges de la distribution. C'est exactement ce que l'on a fait lors du calcul analytique du potentiel créé à grande distance par un dipôle rigide. Ensuite, on obtient une équipotentielle en écrivant que le potentiel $V[i, j]$ est fixé à une valeur donnée :

$$V[i, j] = V_0$$

Le tracé des équipotentielles s'effectue avec la fonction *contour* de *matplotlib.pyplot*.

1.3 Calcul du champ électrique et tracé des lignes de champ

Pour le champ électrique, on se place en statique et on peut écrire que $\vec{E} = -\overrightarrow{\text{grad}} V$. Dans le contexte du plan *Oxy* étudié, on aura donc :

$$\vec{E} = E_x \vec{e}_x + E_y \vec{e}_y = -\frac{\partial V}{\partial x} \vec{e}_x - \frac{\partial V}{\partial y} \vec{e}_y$$

La numérisation utilise la méthode d'EULER pour le calcul de la dérivée en n'oubliant pas que dériver partiellement par rapport à x revient à se déplacer de j à $j + 1$ à i fixé. Si a est le pas de grille, alors on aura :

$$E_x[i, j] = -\frac{V[i, j + 1] - V[i, j]}{a} \quad \text{et} \quad E_y[i, j] = -\frac{V[i + 1, j] - V[i, j]}{a}$$

Pour tracer une ligne de champ, il faut commencer par rappeler sa définition : c'est une ligne qui en chacun de ses points est tangente au champ électrique. Si on note $d\vec{OM} = dx\vec{e}_x + dy\vec{e}_y$ le déplacement élémentaire dans le cas de notre étude bidimensionnelle, le caractère tangent est exprimé par le produit vectoriel du déplacement élémentaire et du champ électrique qui doit être nul. On obtient donc la relation :

$$(E_x \vec{e}_x + E_y \vec{e}_y) \wedge (dx \vec{e}_x + dy \vec{e}_y) = \vec{0} \quad \text{d'où} \quad E_x dy - E_y dx = 0$$

C'est donc grâce à la relation $\frac{dy}{dx} = \frac{E_y}{E_x} = f(x, y)$ que l'on peut envisager le tracé des lignes de champ. Pour l'aspect numérique, il n'y a pas de souci à se faire parce qu'il existe un outil dédié à ce type de situations. La fonction *streamplot* de *matplotlib.pyplot* fait tout cela pour nous une fois connues les valeurs numériques de E_x et E_y en chacun des points de la grille. Attention, il faut quand même se rappeler que par la nature du calcul par EULER des composantes du champ, la grille finale comportera une incrémentation de moins en abscisse et en ordonnée. Il faut dimensionner la matrice pour le champ à la bonne taille.

Par la suite, on commencera par vérifier que le programme donne des résultats satisfaisants pour une charge ponctuelle placée à l'origine du repère *Oxy*. Pour ajouter des charges ponctuelles à la distribution, on construit une fonction qui ajoute les charges par la commande *ajouter_charge*, on crée une liste des charges avec pour chacune sa position dans le plan et la valeur de sa charge grâce à la commande *charges.append*.

2 Une charge ponctuelle

On considère un proton placé seul à l'origine du repère, puis un électron. Le programme fonctionne de façon satisfaisante comme les graphiques de la figure 2.

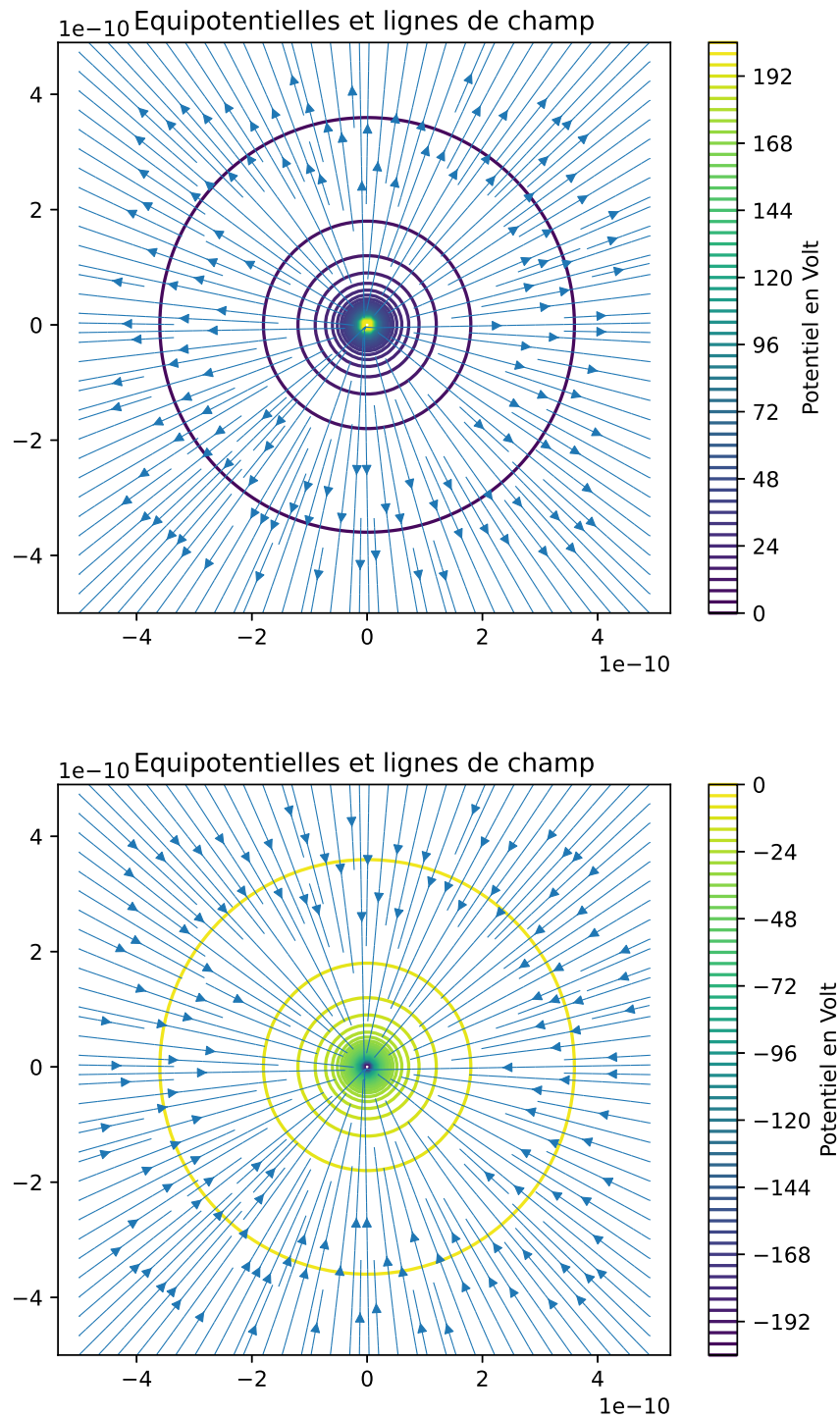


FIGURE 2 – Proton ou électron placé à l'origine du repère

3 Un ensemble de deux charges ponctuelles

Analyser les graphiques de la figure 3.

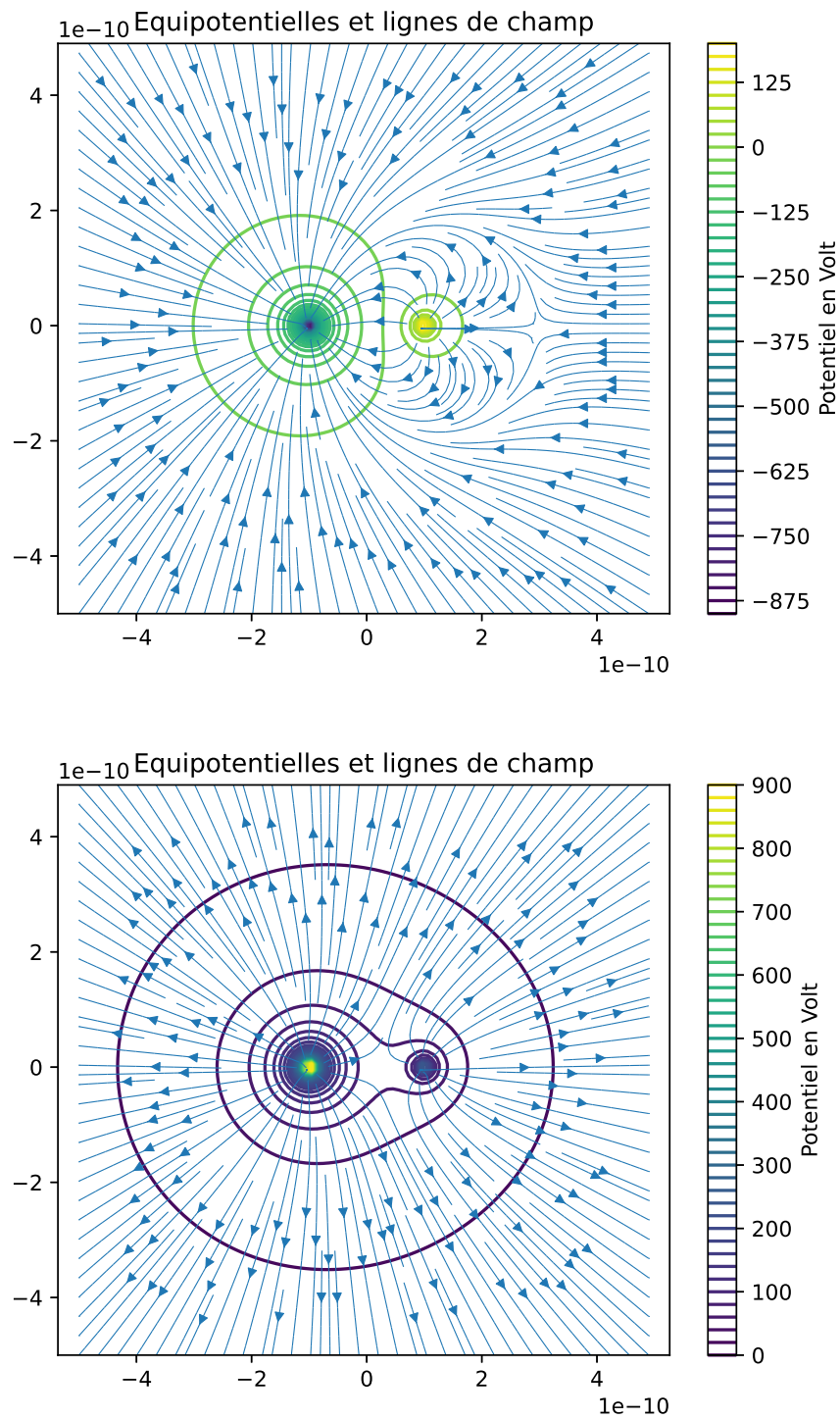


FIGURE 3 – Ensemble de deux charges

4 Un ensemble de trois charges ponctuelles alignées

Analyser les graphiques de la figure 4.

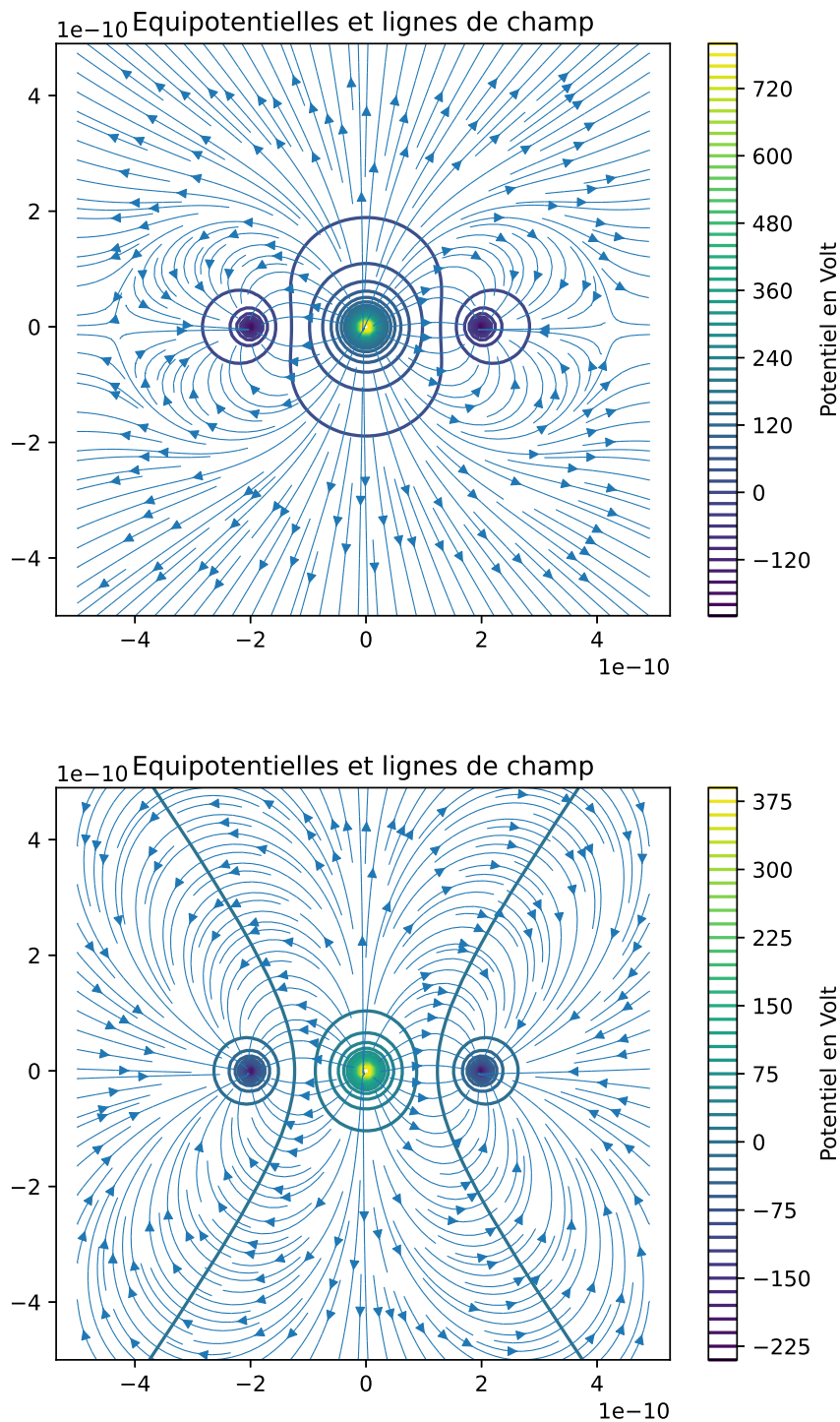


FIGURE 4 – Ensemble de trois charges alignées

5 Un ensemble de quatre charges ponctuelles

Analyser les graphiques de la figure 5 sachant que les charges ont pour valeur $+e$ ou $-e$.

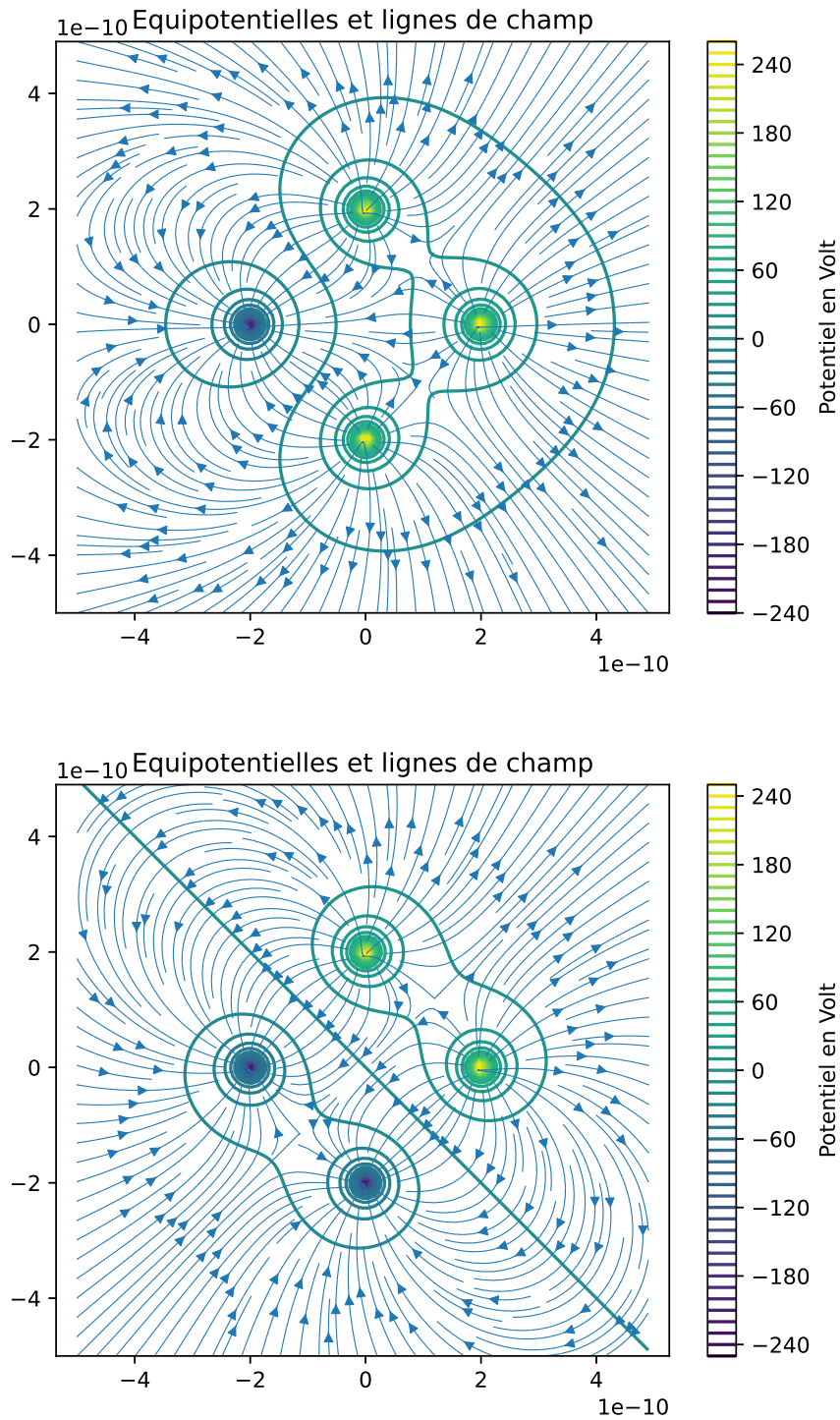


FIGURE 5 – Ensemble de quatre charges

6 Vers le plan chargé

Analyser les graphiques de la figure 6.

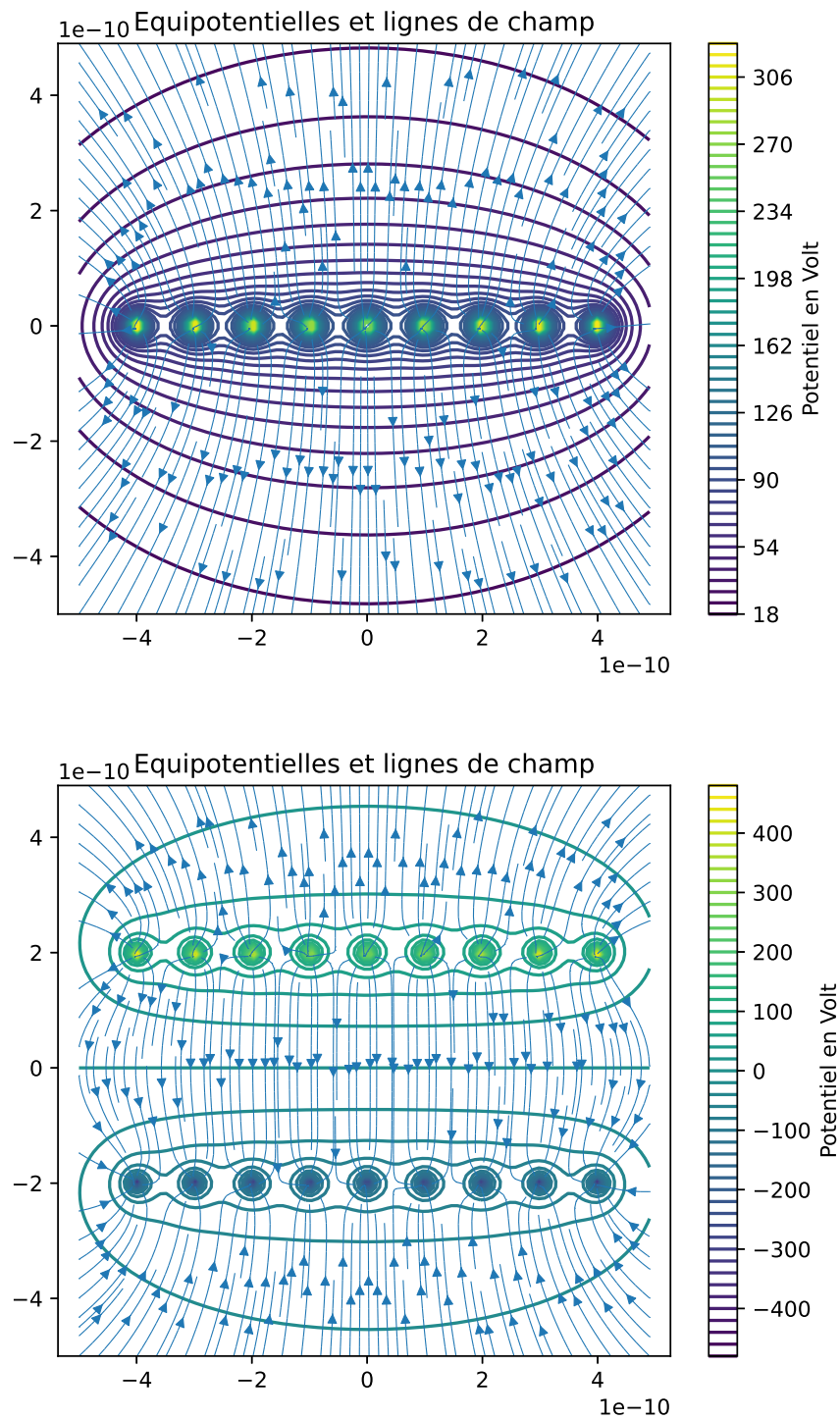


FIGURE 6 – Structure ligne pour donner l'idée d'un plan

7 Distributions plus exotiques

Analyser les graphiques de la figure 7.

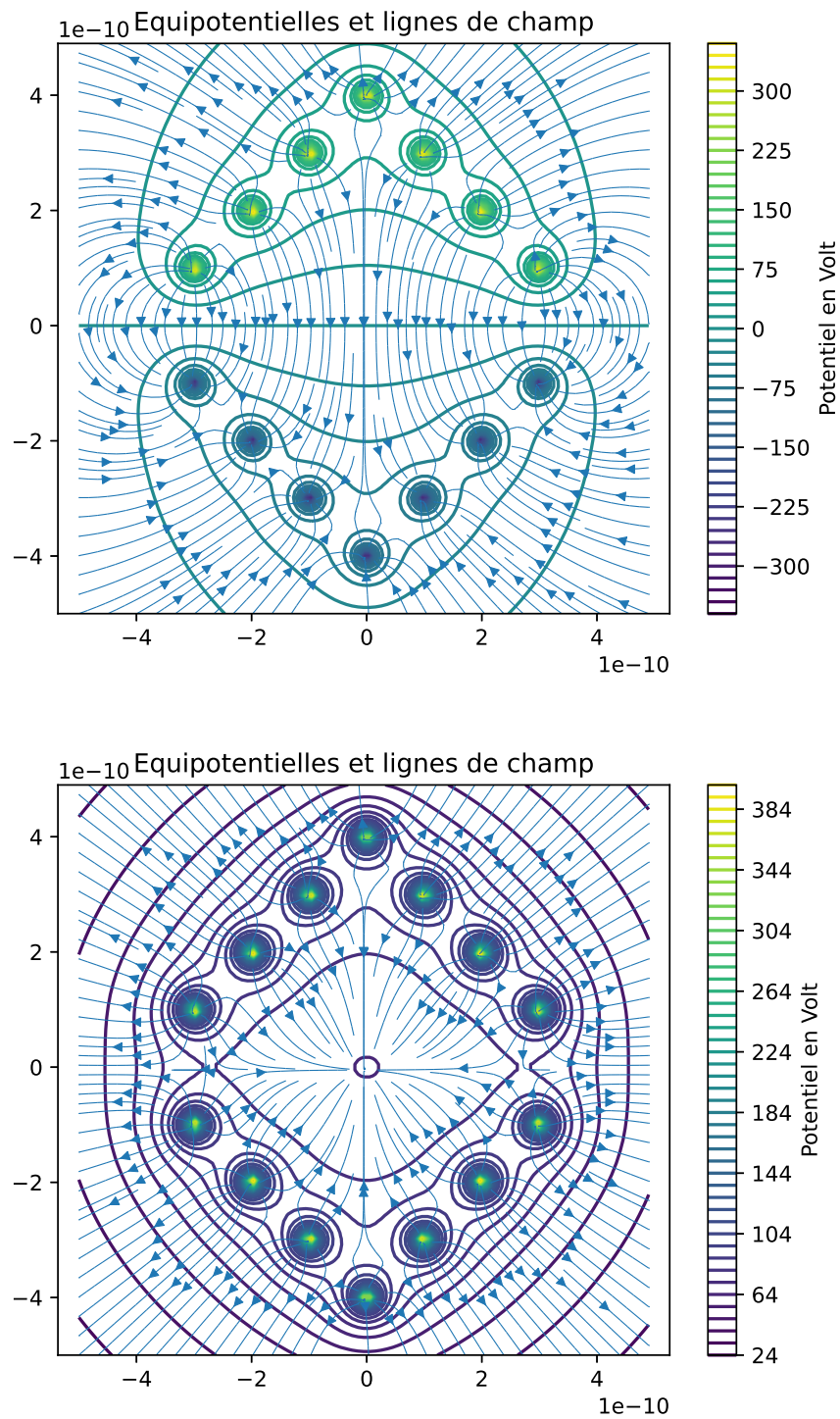


FIGURE 7 – Distributions plus exotiques