Concours Commun INP (CCINP), France

Session 2017, filière PC

Modélisation de systèmes physiques ou chimiques

Durée : 4 heures

Ce que traite le sujet

- Partie I, sous-parties 1 à 3 : équation de Poisson et discrétisation
- Partie I, sous-parties 4 à 6 : méthodes itératives et champ électrique
- Partie II.1 : fil cylindrique chargé uniformément
- Partie II.2 : électron dans un tube d'oscilloscope

Sujet reproduit sans retouche à partir de la page suivante.

Résumé des parties, chapitres de cours liés et autres années :
prepahelp.com/annales/ccinp/pc/2017/modelisation

Prepa Help ne revendique aucun droit sur le sujet, propriété de l'organisateur du concours.

CONCOURS COMMUNS POLYTECHNIQUES

EPREUVE SPECIFIQUE - FILIERE PC

MODELISATION DE SYSTEMES PHYSIQUES OU CHIMIQUES

Jeudi 4 mai : 8 h - 12 h

N.B. : le candidat attachera la plus grande importance à la clarté, à la précision et à la concision de la rédaction. Si un candidat est amené à repérer ce qui peut lui sembler être une erreur d'énoncé, il le signalera sur sa copie et devra poursuivre sa composition en expliquant les raisons des initiatives qu'il a été amené à prendre.

Les calculatrices sont autorisées

Le sujet est composé de deux parties, largement indépendantes.

Autour de l'équation de Poisson

Ce problème s'intéresse à la résolution numérique de quelques problèmes d'électrostatique. Il se compose de deux parties.

I. *Étude de l'équation de Poisson et de différentes méthodes de résolution numérique.*

II. *Deux études de cas : fil infini chargé et mouvement d'un électron entre les plaques d'un condensateur.*

Les différentes parties sont largement indépendantes.

Un aide-mémoire `numpy/matplotlib/pyplot` *est présent à la fin du sujet.*

Partie I - Équation de Poisson

I.1 - Établissement de l'équation

Q1. Rappeler l'équation de Maxwell-Gauss ainsi que la relation entre le champ $\vec{E}$ et le potentiel électrostatique V. En déduire l'équation de Poisson :

$$\Delta V + \frac{\rho}{\varepsilon_0} = 0 \, .$$

Préciser les noms et les unités usuelles de ρ et ε_0.

Q2. Citer plusieurs situations physiques en dehors de l'électrostatique pour lesquelles il existe une équation analogue.

I.2 - Équation adimensionnée pour un problème plan

On veut résoudre l'équation de Poisson dans une portion de plan $\mathcal{P}$ carrée de côté L. On pose :

$$X = x/L, Y = y/L \, .$$

Q3. Montrer qu'on peut écrire l'équation sous la forme suivante :

$$\frac{\partial^2 V(X,Y)}{\partial X^2} + \frac{\partial^2 V(X,Y)}{\partial Y^2} + \rho'(X,Y) = 0$$

où $\rho'(X,Y)$ sera exprimé en fonction de ρ, L et ε_0.

I.3 - Discrétisation

Afin de résoudre numériquement l'équation de Poisson, on va utiliser un maillage de $\mathcal{P}$, de pas $h = 1/N$, et on va transformer les dérivées partielles par des différences entre les valeurs de V aux différents points du maillage (on parle aussi des *nœuds* du maillage). La **figure 1** (page suivante) représente le maillage de $\mathcal{P}$ pour $N = 5$.

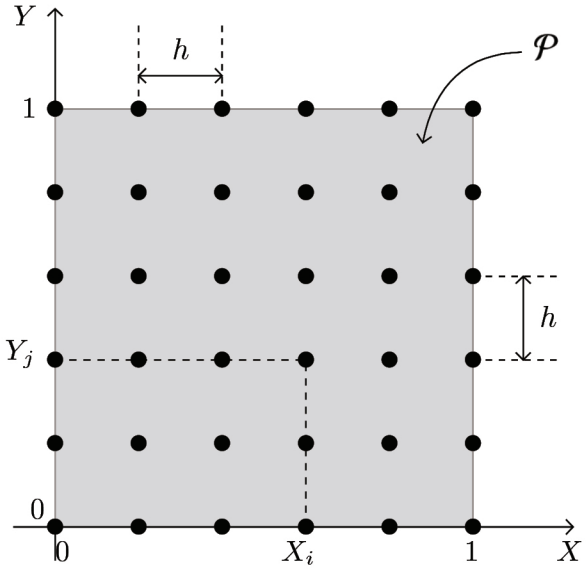


Figure 1 – Maillage de $\mathcal{P}$ pour $N = 5$

Q4. En faisant un développement limité à l'ordre 2 autour du point de coordonnées (X_i, Y_j), montrer qu'on peut exprimer la valeur de $\dfrac{\partial^2 V}{\partial X^2} + \dfrac{\partial^2 V}{\partial Y^2}$ en ce point sous la forme suivante :

$$\frac{\partial^2 V}{\partial X^2} + \frac{\partial^2 V}{\partial Y^2} = \frac{V(X_i + h, Y_j) + V(X_i - h, Y_j) + V(X_i, Y_j + h) + V(X_i, Y_j - h) - 4V(X_i, Y_j)}{h^2} + O(h)\,.$$

Q5. Comme $X_i = ih$ et $Y_j = jh$, on note désormais $V(i, j)$ le potentiel $V(X_i, Y_j)$ en un point (X_i, Y_j) du maillage. Montrer alors qu'on peut écrire l'équation de Poisson sous la forme suivante :

$$V(i + 1, j) + V(i - 1, j) + V(i, j + 1) + V(i, j - 1) - 4V(i, j) + \rho''(i, j) = 0 \qquad (1)$$

$\rho''(i, j)$ étant une fonction à définir en fonction de ρ, L, ε_0 et h.

I.4 - Résolution

La fonction $\rho''(i, j)$ étant connue, on montre en mathématiques que la solution de l'équation de Poisson est unique si on fixe les conditions aux limites sur la frontière $\mathcal{F}$ du domaine $\mathcal{P}$. Ces conditions sont essentiellement de deux types :

— on impose le potentiel en tout point de $\mathcal{F}$ (conditions de Dirichlet),
— on impose une condition sur les dérivées partielles de V en tout point de $\mathcal{F}$ (conditions de Neumann).

Dans ce problème, on ne va considérer que des conditions de Dirichlet.

La frontière $\mathcal{F}$ contient naturellement les points du bord de $\mathcal{P}$ (donc appartenant aux quatre côtés du carré), mais elle peut aussi contenir certains points à l'intérieur de $\mathcal{P}$ où le potentiel est fixé en raison de la présence d'électrodes.

L'ensemble des points de coordonnées (i, j) est donc composé de deux sous-ensembles :

— ceux dont le potentiel est connu, appartenant à la frontière $\mathcal{F}$,
— ceux dont le potentiel est inconnu, appartenant à $\mathcal{P}$ mais pas à $\mathcal{F}$ (donc dans $\mathcal{P}\backslash\mathcal{F}$).

Méthode de Jacobi

À partir de l'équation (1), on peut exprimer :

$$V(i,j) = \frac{1}{4}(V(i+1,j) + V(i-1,j) + V(i,j+1) + V(i,j-1) + \rho''(i,j)) \,. \tag{2}$$

La résolution s'effectue alors en deux étapes.

— Initialisation

a) On fixe le potentiel des points de $\mathcal{F}$ à la valeur imposée physiquement (bords et électrodes).

b) On donne aux points de potentiel inconnu, donc appartenant à $\mathcal{P}\backslash\mathcal{F}$, une valeur arbitraire $V_0(i,j)$, en général nulle.

— Itérations

On calcule une nouvelle valeur $V_1(i,j)$ des potentiels en appliquant l'équation (2) pour tous les points de $\mathcal{P}\backslash\mathcal{F}$, tandis que $V_1(i,j) = V_0(i,j)$ pour les points de $\mathcal{F}$.

Le processus est répété jusqu'à obtenir des valeurs du potentiel quasiment stables. En notant k le nombre d'itérations, on a donc pour le point de coordonnées (i,j) n'appartenant pas à la frontière :

$$V_{k+1}(i,j) = \frac{1}{4}(V_k(i+1,j) + V_k(i-1,j) + V_k(i,j+1) + V_k(i,j-1) + \rho''(i,j)) \,. \tag{3}$$

La convergence de la méthode est vérifiée à l'aide du critère de convergence e_k, défini par :

$$e_k = \sqrt{\frac{1}{N^2}\sum_{i,j}(V_{k+1}(i,j) - V_k(i,j))^2} \quad . \tag{4}$$

Le calcul sera stoppé au bout de k itérations, quand e_k deviendra inférieur à un seuil de convergence ε fixé arbitrairement.

Implémentation informatique

On va utiliser la bibliothèque `numpy` permettant une utilisation simple des tableaux de flottants à deux dimensions ; un aide-mémoire est disponible en fin de sujet.

Le chargement des bibliothèques classiques est assuré par les lignes suivantes :

```
# importation des bibliothèques
import numpy as np
import matplotlib.pyplot as plt
import math
```

On supposera que les tableaux `numpy` suivants, utilisés comme arguments dans les fonctions à définir dans les questions qui suivent, ont pour signification :

— `V[i,j]`, $(i,j) \in [\![0 \dots N]\!]^2$: tableau courant du potentiel en un point de $\mathcal{P}$,

— `rhos[i,j]`, $(i,j) \in [\![0 \dots N]\!]^2$: tableau contenant la densité de charge ρ'' en un point de $\mathcal{P}$,

— `frontiere[i,j]`, $(i,j) \in [\![0 \dots N]\!]^2$: tableau de booléens indiquant si le point de coordonnées (i,j) appartient ou non à $\mathcal{F}$. En particulier, tous les points du bord du domaine seront tels que `frontiere[i,j]==True`.

Q6. Écrire la fonction `nouveau_potentiel(V, rhos, frontiere, i, j)` retournant la nouvelle valeur du potentiel au point $(i, j) \in [\![0 \dots N]\!]^2$ selon l'équation (3).

Q7. Montrer que pour modifier toutes les valeurs contenues dans `V[i,j]` pendant une itération, il est nécessaire de disposer d'une copie de ce tableau.

On rappelle que l'attribut `shape` *permet de récupérer les dimensions d'un tableau* `numpy`.

Q8. Écrire la fonction `itere_J(V, rhos, frontiere)` modifiant la totalité du tableau `V[i,j]` lors d'une seule itération et retournant l'erreur calculée conformément à l'équation (4).

Q9. Écrire la fonction `poisson(f_iter, V, rhos, frontiere, eps)` ayant pour premier argument une fonction du même type que celle définie à la question précédente, pour dernier argument `eps` le seuil arbitraire de convergence ε et dont le rôle est de modifier le tableau des potentiels `V[i,j]` jusqu'à convergence.

I.5 - Améliorations

Méthode de Gauss-Seidel

C'est une modification de la méthode de Jacobi, pour laquelle on montre que la convergence est légèrement plus rapide. Supposons que l'on balaye le tableau des potentiels selon les indices i et j croissants : dans ces conditions, les points situés à gauche et en dessous du point courant ont déjà été calculés. On va utiliser ces nouvelles valeurs, probablement plus proches de la solution, dans la formule permettant le calcul de $V_{k+1}(i, j)$. Ceci donne l'algorithme de Gauss-Seidel :

$$V_{k+1}(i, j) = \frac{1}{4}(V_k(i+1, j) + V_{k+1}(i-1, j) + V_k(i, j+1) + V_{k+1}(i, j-1) + \rho''(i, j)) \,. \tag{5}$$

Q10. Montrer qu'il n'est plus nécessaire de copier le tableau `V[i,j]` pour la mise à jour lors d'une itération en utilisant l'équation (5). Faut-il modifier la fonction `nouveau_potentiel` pour passer de la méthode de Jacobi à celle de Gauss-Seidel ?

Q11. Écrire la fonction `itere_GS(V, rhos, frontiere)` modifiant la totalité du tableau `V[i,j]` lors d'une seule itération et retournant l'erreur calculée conformément à l'équation (4).

Méthode de Gauss-Seidel adaptative

Les méthodes de Jacobi et de Gauss-Seidel n'utilisent pas la valeur de $V_k(i, j)$ pour calculer $V_{k+1}(i, j)$. La méthode de sur-relaxation (***S**uccessive **O**ver **R**elaxation method*) consiste à calculer la nouvelle valeur d'un nœud comme une combinaison linéaire de la valeur courante et de celle donnée par le schéma de Gauss-Seidel. En introduisant le paramètre de relaxation ω, on a alors :

$$V_{k+1}(i, j) = (1-\omega)V_k(i, j) + \frac{\omega}{4}(V_k(i+1, j) + V_{k+1}(i-1, j) + V_k(i, j+1) + V_{k+1}(i, j-1) + \rho''(i, j)) \,. \tag{6}$$

L'étude mathématique de cette relation permet de montrer les résultats suivants :

- la méthode converge uniquement si $0 < \omega < 2$ et elle converge plus rapidement que la méthode de Gauss-Seidel si $1 < \omega < 2$,
- il existe une valeur optimale de ω qui permet la convergence avec un nombre d'itérations en $O(N)$ pour une valeur de ε fixée.

Pour la résolution de l'équation de Poisson envisagée dans ce problème (conditions de Dirichlet sur un maillage carré), on montre que la valeur optimale ω_{opt} est :

$$\omega_{\text{opt}} = \frac{2}{1 + \pi/N} \,. \tag{7}$$

Q12. Écrire la fonction `nouveau_potentiel_SOR(V, rhos, frontiere, i, j, omega)` retournant la nouvelle valeur du potentiel au point (i, j) selon l'équation (6).

Q13. Écrire la fonction `itere_SOR(V, rhos, frontiere)` optimale modifiant la totalité du tableau `V[i,j]` lors d'une seule itération et retournant l'erreur calculée conformément à l'équation (4).

La résolution du problème peut alors se faire par un appel de la forme

`poisson(itere_SOR, V, rhos, frontiere, eps)`,

les tableaux carrés `V, rhos, frontiere` étant de dimensions convenables pour représenter un maillage comportant $(N + 1)^2$ nœuds.

Q14. Quelle est la complexité temporelle de l'appel précédent quand $\omega = \omega_{\text{opt}}$?
La **figure 2** représente, pour $\varepsilon = 10^{-4}$, la durée d'exécution T (en secondes) en fonction de N.
Cette courbe est-elle en accord avec la complexité temporelle attendue ?
Quelle serait la durée d'exécution pour $N = 1\,000$? Commenter.

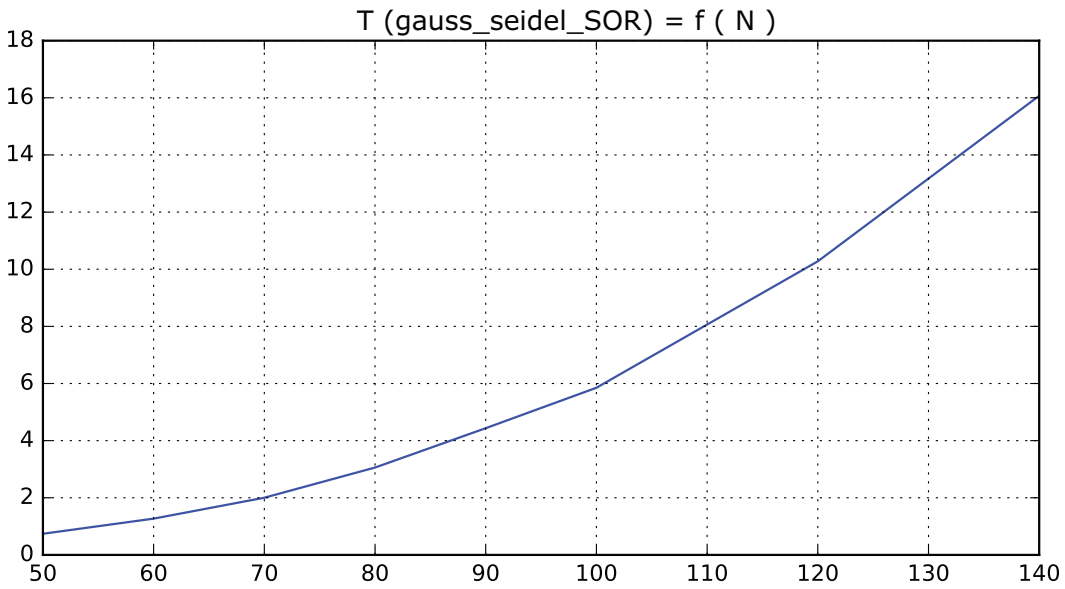


Figure 2 – Durée d'exécution (en secondes) de `poisson(itere_SOR, V, rhos, frontiere, eps)` en fonction de N

I.6 - Détermination du champ électrique

Connaissant le potentiel `V[i,j]`, il est souvent nécessaire de calculer numériquement les composantes E_x et E_y du champ électrique au niveau des nœuds du maillage, qui sont alors conservées dans deux tableaux `Ex` et `Ey` dimensionnés correctement (ce sont donc des tableaux carrés de $(N+1)^2$ éléments).

Q15. Expliquer rapidement comment il serait possible de définir la fonction `calc_ExEy(Ex, Ey, V, h)`, permettant, à partir du tableau `V` et du pas du maillage `h`, le remplissage des deux tableaux `Ex` et `Ey` passés en arguments.
Remarque : on ne demande pas d'écrire le code de la fonction, juste de décrire précisément les étapes de calcul, ainsi que les différents cas à considérer.

Partie II - Deux études de cas

II.1 - Fil cylindrique chargé uniformément

Étude théorique

On considère dans le vide un fil cylindrique infini d'axe z et de rayon R, portant une charge volumique constante ρ.

Q16. En se plaçant en coordonnées cylindriques d'axe z, montrer par des considérations de symétrie et d'invariance que le champ $\vec{E} = E(r)\,\vec{u}_r$. En déduire la forme des surfaces équipotentielles.

Q17. En appliquant le théorème de Gauss, calculer le champ $\vec{E}$ dans tout l'espace. Tracer rapidement l'allure de $E(r)$ en fonction de r.

Q18. On donne : $\varepsilon_0 = 8{,}85 \times 10^{-12}$ F.m^{-1}, $\rho = 1{,}00 \times 10^{-5}$ C.m^{-3}, $R = 5{,}00$ cm. Calculer la valeur maximale de la norme du champ électrique, ainsi que la valeur pour $r = 2R$.

Étude numérique

Pour pouvoir utiliser la méthode de Gauss-Seidel adaptative, on place le fil infini au centre d'une enceinte de longueur infinie et de section carrée ($L \times L$), portée au potentiel nul (**figure 3**).

Dans la suite, on prendra $L = 4R = 20{,}0$ cm.

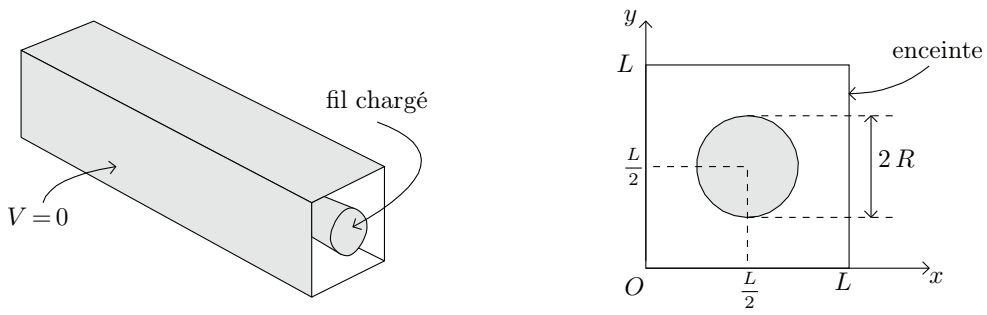


Figure 3 – Fil infini dans une enceinte de section carrée, portée au potentiel nul

Le programme permettant la résolution de ce problème commence ainsi :

```
# initialisations
eps0 = 8.85e-12                 # epsilon_0
L = 20.0e-2                     # 20 cm
N = 100 ; h = L/N               # définition du maillage
rho = 1.00e-5                   # densité vol. de charge rho

# les tableaux globaux numpy pour le cylindre chargé
rhos_cyl = np.zeros((N+1,N+1))  # tableau des valeurs de rho''
V_cyl = np.zeros((N+1,N+1))     # le potentiel
Ex_cyl = np.zeros((N+1,N+1))    # la composante Ex
Ey_cyl = np.zeros((N+1,N+1))    # la composante Ey

# le tableau définissant la frontière est initialement
# rempli entièrement par la valeur False
frontiere_cyl = np.zeros((N+1,N+1), bool)
```

Q19. Écrire la fonction `dans_cylindre(x,y,xc,yc,R)` retournant un résultat booléen indiquant si le point de coordonnées (x, y) est à l'intérieur ou sur le bord du cercle de centre (x_c, y_c) et de rayon R.

Q20. Écrire la fonction `initialise_rhos_cylindre(tab_rhos)`, initialisant le tableau `rhos_cyl` contenant les valeurs $\rho''(i, j)$ pour les nœuds du maillage.

Q21. Écrire la fonction `initialise_frontiere_cylindre(tab_f)`, mettant à `True` les points appartenant à la frontière, donc de potentiel fixé.

La résolution numérique avec la méthode de Gauss-Seidel adaptative, utilisant les valeurs numériques précédentes et un seuil de convergence $\varepsilon = 10^{-5}$, mène à la **figure 4**, où on a tracé un réseau de courbes équipotentielles, le potentiel V et la composante E_x du champ le long de l'axe de symétrie défini par $y = L/2$. En outre, la valeur calculée de `Ex_cyl[50, 50]` est égale à `-0.0023321214257521206`.

Q22. Commenter le plus complètement possible ces résultats ; on veillera, en particulier, à les comparer au modèle théorique (allure des courbes, valeurs numériques ...).

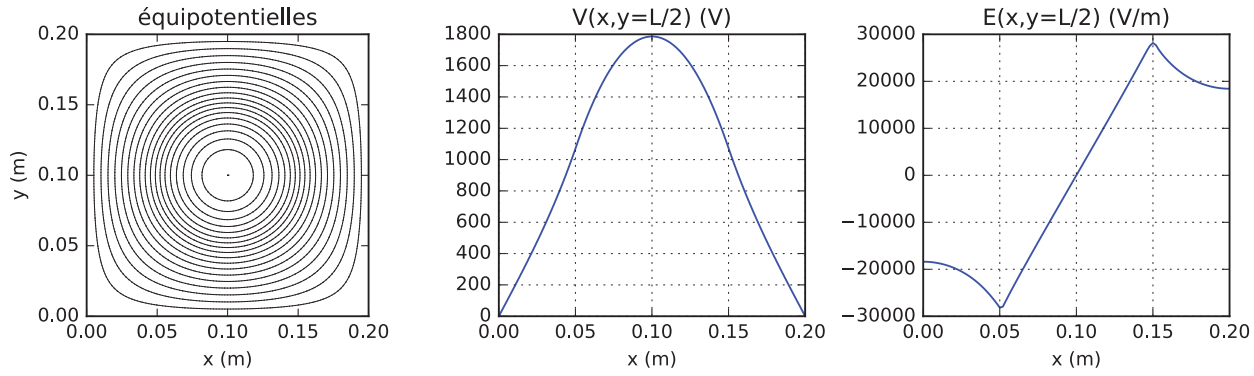


Figure 4 – Équipotentielles, $V(x, y)$ et $\vec{E}(x, y) \cdot \vec{u}_x$ en fonction de x pour $y = L/2$

Une autre résolution est effectuée, avec une répartition de charges dans le cylindre différente de la précédente, utilisant la même valeur de la densité volumique $\rho = 10^{-5}$ C.m^{-3}. Elle mène aux courbes de la **figure 5** (page suivante).

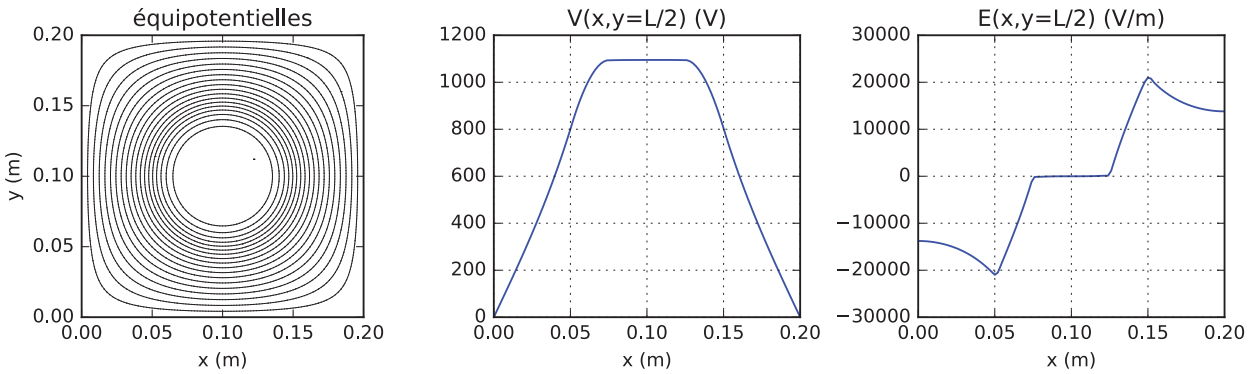


Figure 5 – Équipotentielles, $V(x,y)$ et $\vec{E}(x,y)\cdot\vec{u}_x$ en fonction de x pour la nouvelle répartition de charge

Q23. Déduire de ces courbes la répartition de charges dans le cylindre dans cette deuxième situation. Calculer le champ électrique en tout point pour cette répartition de charges dans le cas d'un cylindre infini seul dans l'espace.
On vérifiera que la valeur maximale du champ électrique calculée à l'aide de cette modélisation est compatible avec celle déduite de la **figure 5**.

II.2 - Mouvement d'un électron dans un tube d'oscilloscope

La **figure 6** montre un tube d'oscilloscope de petite dimension, dans lequel des électrons émis par la cathode sont accélérés et déviés vers un écran luminescent. La déviation est assurée par le passage des électrons entre les plaques de deux condensateurs plans : un pour la déviation horizontale, l'autre pour la déviation verticale. L'étude qui suit ne concernera que le condensateur responsable de la déviation verticale.

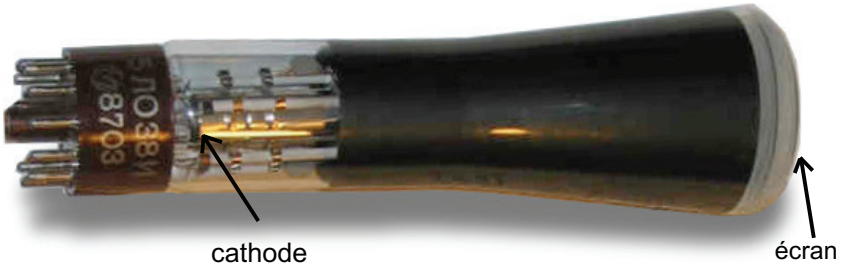


Figure 6 – Petit tube d'oscilloscope, de longueur d'environ 20 cm

On modélise la trajectoire d'un électron de la façon suivante (**figure 7** page suivante, où la zone de déviation est grisée) :

— on négligera l'effet de la pesanteur,
— émis à vitesse nulle par effet thermo-électronique au niveau de la cathode portée au potentiel nul, l'électron est accéléré à l'aide d'une tension $V_0 > 0$ afin d'acquérir à l'entrée de la zone de déviation une vitesse $\vec{v}_0$,
— pendant son trajet dans la zone de déviation, il est soumis à un champ électrique $\vec{E}$ lié aux potentiels $\pm V_p$ des plaques du condensateur, de longueur ℓ et séparées par une distance d,
— poursuivant son mouvement, il arrive sur la surface de l'écran à une distance y_s de l'axe x, l'écran étant situé à la distance D du centre du condensateur.

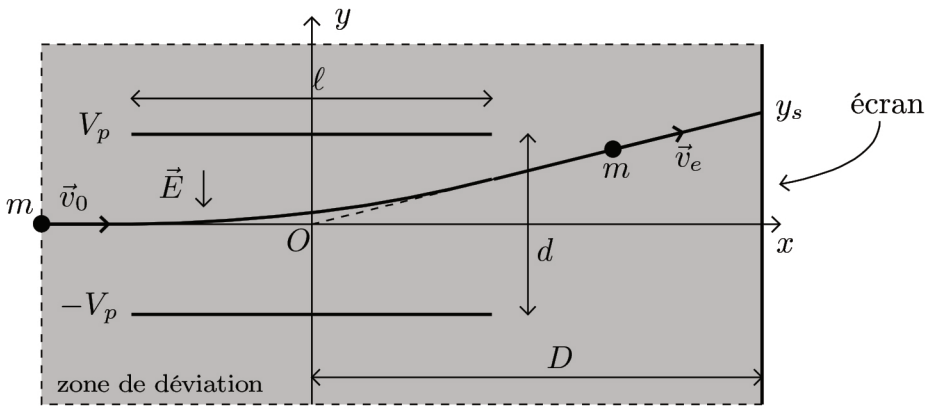


Figure 7 – Schéma de la zone de déviation

Valeurs numériques :

masse d'un électron $m = 9{,}11 \times 10^{-31}$ kg ; charge élémentaire $e = 1{,}60 \times 10^{-19}$ C

$V_0 = 950$ V ; $V_p = 180$ V ; $D = 7{,}00$ cm ; $d = 2{,}00$ cm ; $\ell = 4{,}00$ cm .

Étude physique

Q24. En appliquant la conservation de l'énergie, calculer la vitesse v_0 de l'électron à l'entrée de la zone de déviation. Faire l'application numérique. Commenter.

Q25. On modélise les plaques de déviation comme un condensateur sans effets de bord : le champ électrique est donc considéré comme nul si $|x| > \ell/2$ et uniforme si $|x| \leqslant \ell/2$, ses lignes de champ étant parallèles à l'axe Oy. Exprimer le champ $\vec{E}$ entre les plaques en fonction de V_p et d.

Q26. On suppose que la vitesse d'entrée de l'électron dans la zone de déviation est $\vec{v}_0 = v_0\,\vec{u}_x$. En appliquant les lois de la mécanique, établir l'équation de la trajectoire de l'électron entre les plaques (pour $|x| < \ell/2$).

Q27. Montrer que l'équation de la trajectoire pour $x > \ell/2$ est donnée par : $y = \dfrac{2eV_p\ell}{mdv_0^2}x$.

En déduire que l'ordonnée du spot sur l'écran est : $y_s = \dfrac{V_p}{V_0} \times \dfrac{\ell\, D}{d}$.

Faire l'application numérique pour y_s.

Étude numérique

Pour savoir si la modélisation précédente est pertinente, on va envisager une détermination numérique de la trajectoire de l'électron. Pour cela, on place le condensateur de déviation dans une enceinte carrée au potentiel nul de côté $L = 10{,}0$ cm, le centre du condensateur étant à 3,0 cm du bord gauche de l'enceinte (**figure 8** page suivante). Les autres caractéristiques électriques et géométriques sont les mêmes que précédemment.

La résolution numérique se déroule alors en deux étapes :

— calcul du potentiel et du champ électrique par la méthode de Gauss-Seidel adaptative dans l'enceinte,

— calcul de la trajectoire de l'électron à l'aide de la méthode d'Euler.

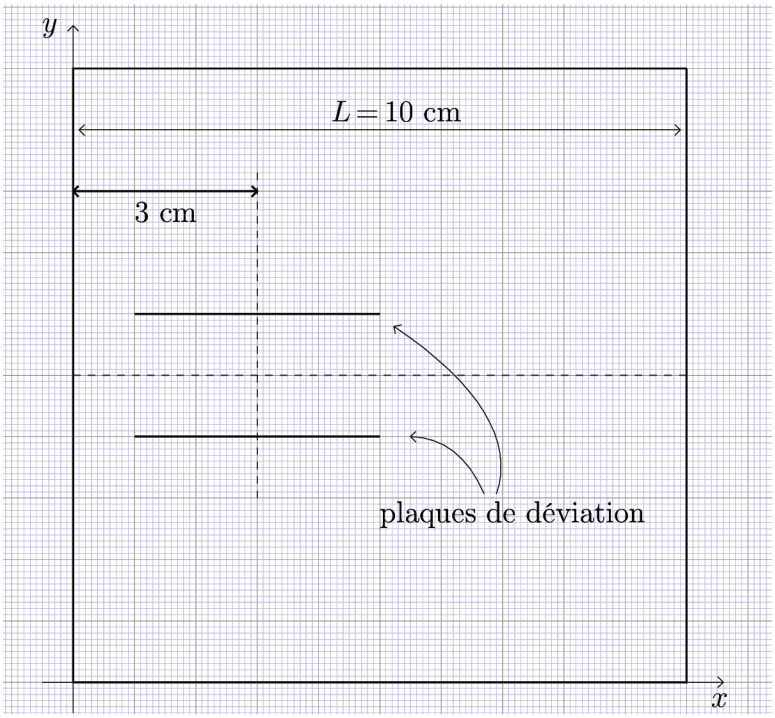


Figure 8 – Modélisation des plaques de déviation dans l'enceinte

Le programme permettant cette résolution numérique commence ainsi :

```
L = 0.100 ; N = 100 ; h = L/N
V0 = 950 ; Vp = 180
m = 9.11e-31 ; e = 1.60e-19
rhos_osc = np.zeros((N+1,N+1))
V_osc = np.zeros((N+1,N+1))
Ex_osc = np.zeros((N+1,N+1))
Ey_osc = np.zeros((N+1,N+1))
frontiere_osc = np.zeros((N+1,N+1), dtype=bool)
```

Calcul du potentiel et du champ électrique dans l'enceinte

Q28. Quelles sont les valeurs qui doivent être contenues dans le tableau `rhos` pour le problème considéré ?

Q29. Écrire la fonction `initialise_frontiere_condensateur(tab_V, tab_f)`, permettant l'initialisation des tableaux `V_osc` et `frontiere_osc` à l'aide de la ligne de code suivante :

```
initialise_frontiere_condensateur(V_osc, frontiere_osc)
```

Pour pouvoir utiliser la méthode d'Euler, il est nécessaire de pouvoir calculer les composantes $E_x(x,y)$ et $E_y(x,y)$ du champ $\vec{E}$ pour $x \in [0, L[$ et $y \in [0, L[$. Cependant, la méthode de résolution (associée à la fonction `calc_ExEy` définie dans la question **15**) ne permet de calculer les composantes `Ex_osc[i,j]` et `Ey_osc[i,j]` qu'aux nœuds du maillage.

Soit un point P de coordonnées $(x, y) \in [0, L[\times[0, L[$. Ce point est dans la cellule (i, j), où $i = \lfloor x/h \rfloor$ et $j = \lfloor y/h \rfloor$. Posons $r_x = x - ih$ et $r_y = y - jh$.

Q30. Montrer alors que :

$$E_x(x, y) \approx \texttt{Ex[i,j] + ((Ex[i+1,j] - Ex[i,j]) * rx + (Ex[i,j+1] - Ex[i,j]) * ry) / h}\,.$$

Écrire de même la formule permettant de calculer $E_y(x, y)$. En déduire qu'il est possible de calculer les composantes du champ en tout point de $\mathcal{P}$.

On supposera dans la suite que les fonctions `val_Ex(Ex,Ey,x,y,h)` et `val_Ey(Ex,Ey,x,y,h)` sont définies et retournent les valeurs des composantes du champ électrique pour le point M de coordonnées (x, y) calculées à l'aide des formules précédentes.

Calcul de la trajectoire par la méthode d'Euler

Q31. Montrer que les équations permettant de décrire le mouvement de l'électron par la méthode d'Euler sont les suivantes, avec δt comme petit incrément temporel et $\delta x, \delta v_x, \delta y, \delta v_y$ les variations pendant δt des grandeurs x, v_x, y, v_y :

$$\begin{cases} \delta x = v_x \delta t & \delta y = v_y \delta t \\ \delta v_x = -\dfrac{e}{m} E_x(x, y) \delta t & \delta v_y = -\dfrac{e}{m} E_y(x, y) \delta t \end{cases}.$$

Q32. Compte tenu du changement de la position d'origine du repère (imposée par la résolution numérique de l'équation de Poisson, **figure 8**), quelles sont les conditions initiales du mouvement de l'électron ?
Déterminer δt pour calculer environ 200 points successifs le long de la trajectoire. Faire l'application numérique.

Q33. En tenant compte des réponses aux questions précédentes, compléter le code d'initialisation des variables de la simulation (`******` dans le code suivant) :

```
Npts = 200 # nombre de points pour le tracé de la trajectoire
v0 = ******  # vitesse initiale de l'électron
dt = ******  # incrément temporel

# tableaux des coordonnées x et y de l'électron
lx = np.zeros(Npts) ; ly = np.zeros(Npts)
# tableaux des vitesses en x et en y
lvx = np.zeros(Npts) ; lvy = np.zeros(Npts)
#conditions initiales
lx[0] = ****** ; ly[0] = ******
lvx[0] = ******* ; lvy[0] = *******
```

Q34. Écrire les lignes de code implémentant la boucle de remplissage des tableaux `lx, ly, lvx, lvy` selon la méthode d'Euler.

Comparaison théorie/simulation

La **figure 9** montre le résultat de la simulation précédente. On y voit le réseau de courbes équipotentielles, ainsi que deux trajectoires 1 et 2, l'une étant associée au calcul théorique, l'autre à la simulation numérique. Chaque trajectoire est constituée de 200 points de calcul séparés d'une durée δt.

Q35. Reproduire sommairement sur la copie la **figure 9**, y ajouter le tracé de quelques lignes de champ orientées dans les différentes parties de la zone de déviation.
Identifier, en le justifiant, chaque trajectoire. Expliquer pourquoi la trajectoire 1 est plus courte que la trajectoire 2.
À votre avis, peut-on se contenter de l'étude théorique pour prévoir le point d'impact de l'électron sur l'écran ? (On attend une réponse chiffrée.)

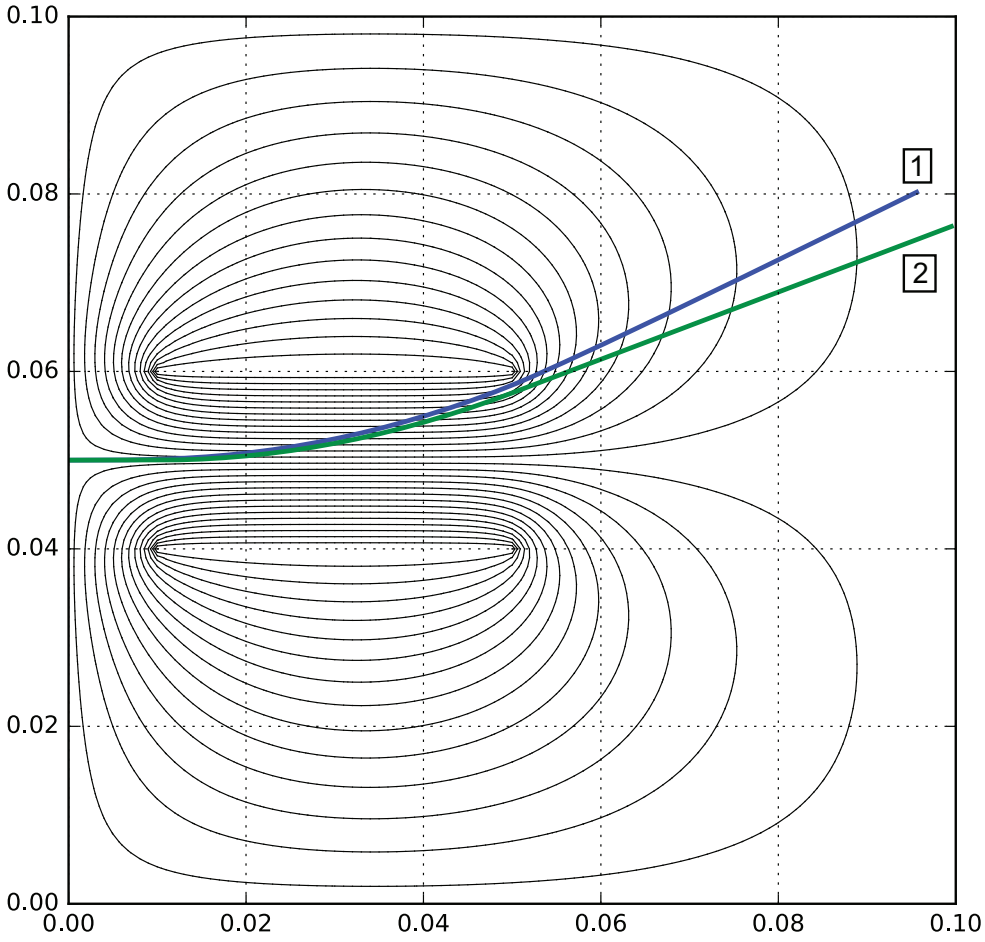


Figure 9 – Tracé dans la zone de déviation de quelques courbes équipotentielles, des trajectoires de l'électron (théorique et simulée numériquement)

Aide-mémoire numpy/matplotlib/pyplot

Importation des bibliothèques

Les bibliothèques sont importées de la façon suivante :

```
import math
import numpy as np
import matplotlib.pyplot as plt
```

Manipulation des tableaux numpy

La création d'un tableau `numpy` à deux dimensions dont toutes les valeurs sont initialisées à 0 est faite par l'instruction `np.zeros(format)`, `format` étant un doublet de la forme `(n_lignes , n_colonnes)` :

```
>> t0=np.zeros((2,3)); print(t0)
[[ 0.  0.  0.]
 [ 0.  0.  0.]]
```

Pour avoir un tableau rempli de 1, on utilise `np.ones(format)` :

```
>> t1=np.ones((2,2)); print(t1)
[[ 1.  1.]
 [ 1.  1.]]
```

On peut récupérer le format d'un tableau en demandant son attribut `shape`, ce qui retourne un doublet :

```
>> print(t0.shape); print(t1.shape)
(2, 3)
(2, 2)
```

Dans le cas d'un tableau carré, on peut donc récupérer le nombre de lignes, égal au nombre de colonnes, en accédant au premier élément du doublet :

```
>> t1.shape[0]
 2
```

On peut créer un tableau `numpy` de booléens en ajoutant le type `bool`. La valeur 0 est associée à `False`, la valeur 1 à `True` :

```
>> np.zeros((2,3), bool)
[[False False False]
 [False False False]]}
>> np.ones((2,3),bool)
[[ True True True]
 [ True True True]]
```

L'accès à un élément du tableau `a` (en lecture ou en modification) se fait par `a[i,j]`, les lignes et les colonnes étant numérotées à partir de 0 :

```
>> a=np.zeros((2,3)) ; a[0,0]=1 ; a[1,2]=2
>> a
[[ 1. 0. 0.]
 [ 0. 0. 2.]]
```

Une copie indépendante d'un tableau `a` se fait à l'aide de `np.copy(a)`

```
>> b = np.copy(a) ; b[0,0]=3 ; b[1,1]=5
>> a
[[ 1. 0. 0.]
 [ 0. 0. 2.]]
>> b
[[ 3. 0. 0.]
 [ 0. 5. 2.]]
```

FIN

Travailler les chapitres de ce sujet

Prepa Help suit le programme de PC chapitre par chapitre : cours détaillé, méthodes, pièges, exercices et problèmes corrigés pas à pas, tirés des concours. L'essai gratuit donne accès aux chapitres offerts de ta filière, sans carte bancaire.

Chapitres liés à ce sujet

- Électrostatique : formulation locale et condensateur
- Calcul infinitésimal du physicien : dérivées et différentielles
- Bases de Python : expressions, types, instructions, fonctions
- Description et paramétrage du mouvement d'un point

Ouvrir l'essai gratuit

Scanne le code ou clique sur le bouton.

prepahelp.com/#essai

Une question sur le concours ou sur ta filière : **écrire sur WhatsApp**

Les autres années et épreuves : prepahelp.com/annales/ccinp/pc