Partie V : Électromagnétisme
TP

TP 17 – Résolution numérique de l'équation de Laplace

Objectif : écrire puis utiliser un algorithme numérique de résolution de l'équation de Laplace.

Données : vous disposez d'un fichier script dans lequel sont déjà présentes quelques fonctions Python. Ces fonctions serviront à l'affichage des graphiques et à l'initialisation du domaine. Ouvrir ce fichier, et en copier-coller le contenu au début de votre propre fichier Python.

I Introduction

Une fonction f suit l'équation de Laplace si son Laplacien scalaire est nul : $\Delta f = 0$.

Il s'agit d'une équation que l'on rencontre dans de nombreux domaines de la physique. Nous l'avons vue en thermodynamique, puisque l'équation de diffusion en régime stationnaire s'écrit $\Delta T = 0$, en électrostatique puisqu'en l'absence de charges on a $\Delta V = 0$, et on la rencontre également en gravitation, pour l'étude des vibrations d'une membrane, en hydrodynamique, etc.

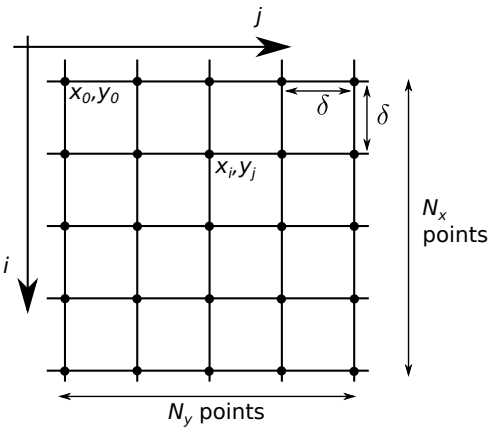
On peut trouver des solutions analytiques à cette équation dans des cas simples. Cependant, de façon plus générale, il est nécessaire d'utiliser un algorithme de résolution numérique. C'est ce que l'on propose ici dans la partie II. Ensuite, dans la partie III, nous utiliserons cet algorithme pour obtenir le potentiel dans le cas de l'effet de pointe (ou effet paratonnerre).

II Écriture de l'algorithme de résolution

II.1 Discrétisation de l'équation

On se restreint au cas d'un espace à deux dimensions. L'espace est ainsi représenté par une grille de taille `Nx`×`Ny`. La distance entre deux points de la grille est le pas (spatial) δ. On note (x_i, y_i) les coordonnées du point (i, j) de la grille.

Le potentiel V est représenté par un tableau `V`, avec `V[i][j]` donnant la valeur du potentiel $V(x_i, y_i)$, ce qui correspond au point `i,j` de la grille.



Il est nécessaire de discrétiser l'opérateur laplacien. Comme en coordonnées cartésiennes il s'écrit :

$$\Delta V = \left(\frac{\partial^2 V}{\partial x^2}\right)_y + \left(\frac{\partial^2 V}{\partial y^2}\right)_x,$$

il faut choisir une écriture discrète de la dérivée seconde.

On effectue pour cela un développement limité à l'ordre 2 de V autour du point `i,j` :

$$V(x_i + \delta, y_j) = V(x_i, y_j) + \delta \left(\frac{\partial V}{\partial x}\right)_y (x_i, y_j) + \frac{\delta^2}{2}\left(\frac{\partial^2 V}{\partial x^2}\right)_y (x_i, y_j) + O(\delta^2).$$

et de même en changeant δ en $-\delta$:

$$V(x_i - \delta, y_j) = V(x_i, y_j) - \delta \left(\frac{\partial V}{\partial x}\right)_y (x_i, y_j) + \frac{\delta^2}{2}\left(\frac{\partial^2 V}{\partial x^2}\right)_y (x_i, y_j) + O(\delta^2).$$

1 - En déduire que

$$\left(\frac{\partial^2 V}{\partial x^2}\right)_y [i, j] \simeq \frac{V[i+1, j] + V[i-1, j] - 2V[i, j]}{\delta^2},$$

puis que

$$\boxed{(\Delta V)[i, j] \simeq \frac{V[i+1, j] + V[i-1, j] + V[i, j+1] + V[i, j-1] - 4V[i, j]}{\delta^2}.} \quad (1)$$

II.2 Résolution avec la méthode de Jacobi

L'équation de Laplace s'écrit donc, pour tout i, j dans le domaine :

$$(\Delta V)[i,j] = \frac{1}{\delta^2}\left(V[i+1,j] + V[i-1,j] + V[i,j+1] + V[i,j-1] - 4V[i,j]\right) = 0,$$

soit encore

$$V[i,j] = \frac{V[i+1,j] + V[i-1,j] + V[i,j+1] + V[i,j-1]}{4}. \tag{2}$$

Nous allons donc utiliser cette dernière équation pour construire par itération les valeurs de V sur la grille. Avant cela, il faut parler des bords du domaine.

Nous avons dit en cours que l'équation de Laplace admet une unique solution dans un domaine $\mathcal{D}$ de l'espace si la valeur du potentiel V est fixée sur les bords du domaine. Ici, les bords du domaine peuvent simplement être le cadre de la grille, sur lequel on impose un potentiel V_0 choisi. Mais les bords peuvent aussi inclure d'autres points de la grille, qui peuvent par exemple représenter les armatures d'un condensateur, ou un conducteur métallique qui est à un potentiel V_0 fixe.

Nous introduisons donc le tableau `B`, de taille `Nx`×`Ny`, qui est tel que :

- Si `B[i,j] = 1`, alors c'est que la case `i,j` de la grille appartient au bord du domaine, et que son potentiel est imposé. Il ne faudra pas changer la valeur de `V` en ce point.
- Si `B[i,j] = 0`, alors c'est que la case `i,j` n'appartient pas au bord du domaine. Il faudra alors utiliser l'équation 2.

2 - Écrire une fonction `initialisation_cadre(B,V,V0)` qui prend en argument une matrice `B` et qui initialise les points de son cadre à la valeur 1, et une matrice `V` et qui initialise les points de son cadre à la valeur `V0`. Cette fonction ne retourne rien, elle modifie simplement les valeurs des matrices `B` et `V`.

Pour tester on peut créer les matrices `B` et `V` à l'aide de :

```
B = np.zeros((Nx,Ny),bool)  # type bool car ne contient que des 0 ou des 1
V = np.zeros((Nx,Ny))
```

puis appeler `initialiser_cadre`, et afficher la grille obtenue à l'aide de la fonction `graphe_bords(B)` fournie dans le script.

L'algorithme est ensuite le suivant :

- Initialisation :
  - On crée la matrice `B` et on l'initialise avec des 0 ou des 1 pour décrire les bords du domaine.
  - On crée la matrice `V` et on l'initialise avec les valeurs voulues sur les bords du domaine, et avec des 0 ailleurs.

  Cette étape dépend donc du problème que l'on veut traiter.
- Itérations. Une itération consiste à effectuer :

  Pour tout point `i,j` qui n'appartient pas à un bord, on calcule une nouvelle valeur `V1` de `V` selon l'équation 2. Ainsi V à l'itération `k` est obtenu en fonction de V à l'itération `k-1` via :

$$V_k[i,j] = \frac{V_{k-1}[i+1,j] + V_{k-1}[i-1,j] + V_{k-1}[i,j+1] + V_{k-1}[i,j-1]}{4}. \tag{3}$$

- Critère de terminaison :

  On stoppe la simulation lorsque les itérations ne font quasiment plus évoluer le potentiel.

On définit pour cela

$$e = \sqrt{\frac{1}{N_x N_y} \sum_{i,j} [V_k(x_i, y_j) - V_{k-1}(x_i, y_j)]^2}, \tag{4}$$

qui représente l'écart entre la nouvelle valeur de V et l'ancienne en moyenne sur toute la grille.

On stoppe le programme lorsque e devient inférieur à une valeur ϵ (que l'on choisira dans la suite).

3 - Écrire une fonction `une_iteration(B,V)` qui prend en argument une matrice de bords `B` et la matrice des potentiels `V`, et qui effectue une itération de l'algorithme décrit plus haut.

On rappelle que pour effectuer une copie d'un tableau numpy on peut utiliser `V_copie = V.copy()`

Cette fonction devra retourner l'écart e défini par l'équation 4.

4 - Écrire une fonction `iterations(B,V,eps)` qui prend en argument une matrice de bords `B`, la matrice des potentiels `V`, un réel `eps`, et qui itère l'algorithme tant que l'écart défini équation 4 est supérieur à `eps`.

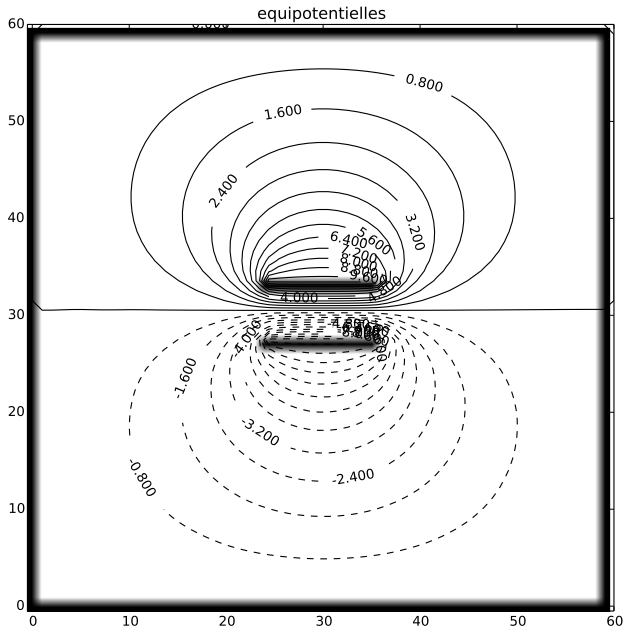
La fonction retournera le nombre d'itérations effectuées.

5 - On va tester notre algorithme. Dans votre script, il faut pour cela réaliser les opérations suivantes :

- Initialiser des tableaux `B` et `V` de taille `Nx=60` par `Ny=60`.
- Initialiser le cadre du domaine au potentiel nul. Et initialiser un condensateur plan dont l'armature du bas est au potentiel $a = -10\,\mathrm{V}$ et l'armature du haut au potentiel $b = +10\,\mathrm{V}$ à l'aide de la fonction `initialisation_condensateur(B,V,a,b)`.
- Lancer les itérations. On prendra `eps=1.e-3`.
- Afficher les surfaces équipotentielles en exécutant la fonction `graphe_equipot(B,V)`.

Commenter par rapport à ce qui a été vu en cours sur le condensateur plan.

On voit donc qu'un tel algorithme permet d'obtenir de façon précise les surfaces équipotentielles pour un condensateur réel (non infini !).



II.3 Résolution avec la méthode optimisée de Gauss-Seidel

Si on note $g(N)$ le nombre d'itérations nécessaire à la convergence, on peut constater que $g(N)$ augmente plus que linéairement avec N. Ceci n'est pas très bon pour les temps de calcul, car comme à

chaque itération on effectue N^2 opérations pour parcourir toute la grille, cela implique que la complexité de l'algorithme est pire que $O(N^3)$.

Une amélioration possible est d'utiliser la méthode de Gauss-Seidel adaptative, qui remplace l'équation 2 par :

$$V_k[i,j] = (1-\omega)V_{k-1}[i,j] + \omega \times \frac{V_{k-1}[i+1,j] + V_k[i-1,j] + V_{k-1}[i,j+1] + V_k[i,j-1]}{4}. \quad (5)$$

avec ω un paramètre. Il y a donc deux modifications par rapport à l'algorithme précédent :

- L'introduction de la valeur $V_{k-1}[i,j]$ de V au point i,j de l'itération précédente. Il apparaît avec un facteur ω.
- Le calcul de $V_k[i,j]$ à l'itération k fait intervenir des valeurs à l'itération précédente $k-1$ comme c'était déjà le cas, mais également certaines valeur à l'itération k elle même : $V_k[i-1,j]$ et $V_k[i,j-1]$. Ceci est motivé par le fait que, comme on parcourt la grille à i et j croissants, ces valeurs ont en fait déjà été actualisées lors de cette itération. Cela permet donc d'améliorer la vitesse de convergence.

  En pratique dans l'algorithme, il ne faudra plus utiliser une matrice copie de celle de V, mais directement itérer avec

```
V[i,j] = (1.-w)*V[i,j] + w*(V[i+1,j]+V[i,j+1]+V[i-1,j]+V[i,j-1])/4.
```

On peut montrer qu'il existe une valeur optimale de ω qui permet d'avoir un nombre d'itération en $O(N)$ pour une grille de taille $N_x = N_y = N$:

$$\omega_{\text{opt}} = \frac{2}{1+\frac{\pi}{N}}.$$

6 - Écrire les fonctions `une_iteration_GS(B,V)` et `iterations_GS(B,V,eps)` qui réalisent la même chose que `une_iteration(B,V)` et `iterations(B,V)`, mais en utilisant l'algorithme de Gauss-Seidel.

Tester ceci sur le même cas que précédemment avec le condensateur. Le nombre d'itérations est-il moins important que précédemment ?

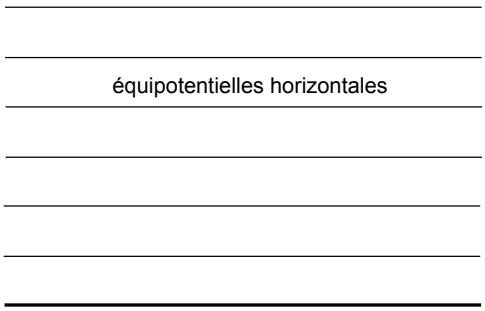
III Application à l'étude de l'effet de pointe

On s'intéresse maintenant à une autre application.

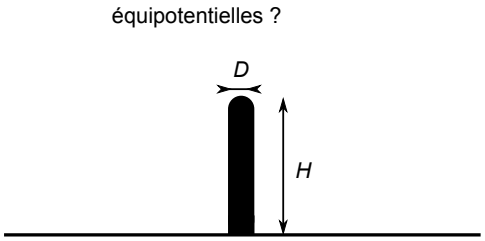
En l'absence de perturbations, il y a dans l'atmosphère terrestre un gradient de potentiel entre le sol (qui est à un potentiel V_{sol} que l'on prendra égal à 0 V) et l'ionosphère terrestre (qui est à un autre potentiel). Ainsi, entre le sol et une hauteur L donnée, on passe quasi linéairement du potentiel V_{sol} à un potentiel $V(L)$ plus élevé. Dans un atmosphère non perturbée, par temps calme, le gradient de potentiel est de 100 V/m environ.

On perturbe maintenant cette situation en plaçant une tige verticale de hauteur H et de largeur D, dont la pointe est une demi-sphère de diamètre D, faite dans un matériau conducteur et qui est donc au même potentiel en tout point. Cette tige est reliée au sol, son potentiel est donc aussi celui du sol. Ceci va déformer les surfaces équipotentielles, et donc en particulier modifier la valeur du champ électrique. On s'attend à ce que ce dernier augmente vers la pointe : c'est ce que l'on nomme l'effet de pointe, et qui explique que la foudre tombe préférentiellement sur les objets pointus.

Notre programme va permettre de simuler cette situation, et d'obtenir de façon précise les surfaces équipotentielles et surtout la valeur maximale du champ électrique.



situation de repos : sol à 0V et gradient de 100V/m à mesure que l'on monte



situation perturbée : ajout d'une tige métallique qui est au potentiel 0V.

L'objectif est le suivant : obtenir une relation entre la valeur $E_{\max}$ du champ électrique et le diamètre D de la pointe de la tige.

Remarque : Notre programme étant à 2D, nous simulons en fait un plan de hauteur H, épaisseur D, de longueur très grande dans la direction perpendiculaire au plan de la simulation. Les temps de calculs seraient trop grands à 3D sur les ordinateurs du lycée.

7 - Quelle est la valeur de $\|\vec{E}\|$ et la direction de $\vec{E}$ dans la situation non perturbée ?

8 - La fonction `initialisation_effet_pointe(B,V,a,b,h,d)` fournie dans le script permet d'initialiser un domaine avec :

- Un cadre dont le bord bas est au potentiel a et le bord haut au potentiel b. Sur les bords droit et gauche du cadre, le potentiel est aussi fixé et augmente linéairement de a à b.
- Une tige de hauteur h et épaisseur d (en nombre de points), au potentiel fixé a. Le bout de la tige est arrondi. Attention, d doit être *impair* pour que cela fonctionne bien.

Initialiser le problème à l'aide de cette fonction. On prendra un pas de grille $\delta = 3 \times 10^{-2}\,\text{m}$, et :

- $h = 50$ et $d = 13$, ce qui correspond à une tige de hauteur $H = h \times \delta = 1.5\,\text{m}$ et de largeur $D = d \times \delta = 39\,\text{cm}$,
- un domaine de taille $N_x \times N_y = 120 \times 120$, ce qui correspond à $N_x\delta \times N_y\delta = 3.6\,\text{m} \times 3.6\,\text{m}$,
- un potentiel $a = 0\,\text{V}$ au sol et $b = N_x \times \delta \times 100\,\text{V}$ en haut du cadre, ce qui donne bien un gradient au repos de $100\,\text{V/m}$.

Visualiser si tout est correct à l'aide de `graphe_bords(B)`.

Utiliser ensuite l'algorithme de Gauss-Seidel pour obtenir le potentiel. On prendra encore $\epsilon = 10^{-3}$. Visualiser la solution à l'aide de `graphe_equipot(B,V)`.

9 - Sur la solution obtenue précédemment, quelles sont les zones où le champ électrique sera le plus fort ? Quelle propriété vue en cours utilisez-vous pour affirmer ceci ?

On souhaite ensuite obtenir le champ électrique en tout point du domaine.

10 - Rappeler la relation locale entre le champ électrique et le potentiel, qui est valable ici car la situation est stationnaire.

Écrire cette relation dans le cas d'une dépendance en x et y seulement.

11 - On choisit ici d'évaluer la dérivée partielle première à l'ordre 1 avec le schéma suivant :

$$\left(\frac{\partial V}{\partial x}\right)_y (x_i, y_j) = \frac{V(x_i + \delta, y_j) - V(x_i - \delta, y_j)}{2\delta},$$

et de même dans la direction y pour la dérivée par rapport à y (à x fixé). (On peut montrer avec un DL que ceci est précis en $O(\delta^2)$.)

Écrire une fonction `calcul_E(B,V,delta)` qui prend en argument la matrice `B` des bords du domaine et la matrice `V` des potentiels, toutes deux de taille `Nx`×`Ny`, ainsi que le pas de la

grille `delta`, et qui retourne trois matrices `Ex`, `Ey`, `norme_E` de tailles `Nx`×`Ny` qui contiennent respectivement les composantes x et y du champ électrique et sa norme.

On ne traitera pas les valeurs sur le bord du domaine.

Tester cette fonction sur le cas précédent et noter la valeur maximale de $\|\vec{E}\|$. Par combien le champ de la situation non perturbée a-t-il été multiplié par la présence de la pointe ?

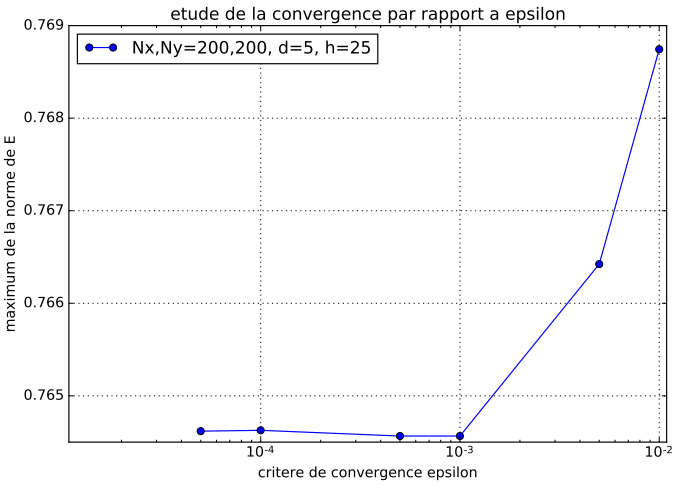
12 - Répéter ceci pour quelques valeurs de d (par exemple 7, 13 déjà fait, 17, 23). Tracer ensuite le maximum de la norme du champ électrique en fonction de d. Ceci va-t-il dans le sens de l'effet de pointe mentionné plus tôt ?

Compléments : étude de la convergence des résultats

Un point majeur en simulations numériques est l'influence du critère de convergence et de la taille du domaine.

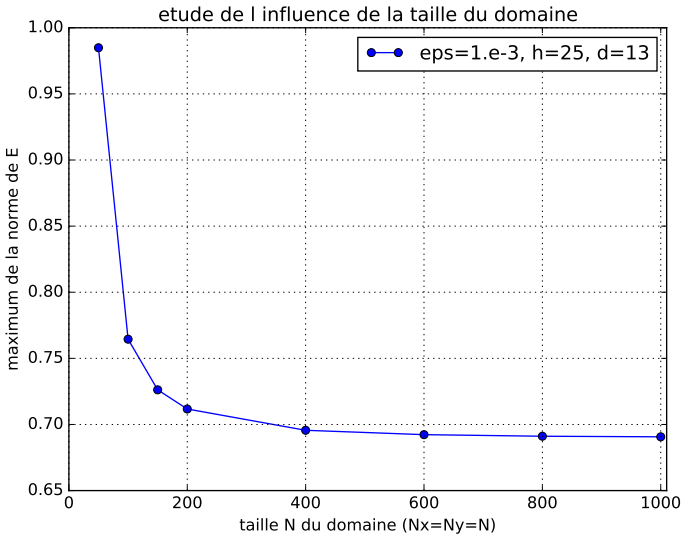
13 - Influence du critère de convergence ϵ :

Dans l'idéal, on souhaiterait prendre un critère ϵ le plus petit possible, mais cela impliquerait des temps de calculs immenses. On doit donc trouver un compromis entre rapidité et précision. Le critère est que les résultats de la simulation ne doivent *pas* dépendre de ϵ. S'ils en dépendent, c'est que ϵ est encore trop grand.
Le graphique ci-contre donne le résultat de plusieurs simulations, pour $d = 5$, $h = 25$ et $N_x = N_y = 200$ et différentes valeurs de ϵ.
Justifier alors pourquoi l'énoncé suggère de prendre $\epsilon = 10^{-3}$.



14 - Influence de la taille du domaine :

Ici aussi, on prendrait dans l'idéal un domaine de taille immense, mais cela implique une place en mémoire et un temps de calcul beaucoup trop grand. Il faut donc vérifier si les résultats de la simulation dépendent ou non de la taille du domaine. Ils ne doivent pas en dépendre.
Le graphique ci-contre donne le résultat de plusieurs simulations, pour $d = 13$, $h = 25$ et $\epsilon = 10^{-3}$, et différentes valeurs de N (le domaine est de taille $N_x = N$ par $N_y = N$).
Pourquoi peut-on dire que les résultats obtenus précédemment dans le TP ne sont en fait pas valides ? Quelle taille de domaine faudrait-il prendre au minimum ? Estimer alors le temps de calcul nécessaire (en vous référant à la complexité de l'algorithme). Pourquoi est-ce impossible pour un TP de 2h ?



Autre exemple : équation de la chaleur en régime stationnaire

La méthode de résolution vue dans ce TP peut également s'appliquer dans d'autres contextes où l'équation est la même. Ainsi, l'équation de la chaleur vue en thermodynamique, $\dfrac{\partial T}{\partial t} = \kappa \Delta T$, devient en régime stationnaire

$$\Delta T = 0,$$

que l'on résout de la même façon.
L'exemple ci-contre montre le champ de température dans un solide maintenu à 20°C sur la face du haut et à 60°C sur les trois autres faces.