FILIÈRE MP

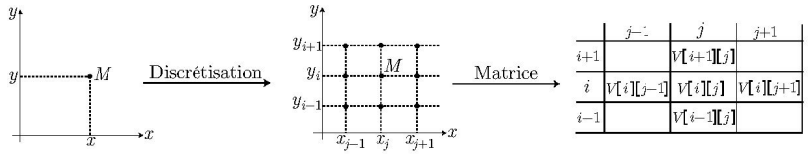
PHYSIQUE

Équation de Laplace

On cherche ici à développer un script qui permettrait d'évaluer numériquement le potentiel $V(x,y)$, solution de l'équation de Laplace :

$$\Delta V = 0 \Rightarrow \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0 \qquad (1)$$

Pour y parvenir, nous adoptons le modèle suivant : la fonction $V(x,y)$ sera discrétisée, c'est-à-dire que les variables x et y seront supposées varier par incréments $\mathrm{d}x = 1$ et $\mathrm{d}y = 1$ (et non de manière continue), ce qui permettra de remplacer la fonction $V(x,y)$ par une matrice $V[i][j]$, dans laquelle se trouvent consignées les valeurs de V dans la $i^{\text{ème}}$ ligne et la $j^{\text{ème}}$ colonne :



1. Considérons la fonction $V(x,y)$ au voisinage du point $M(x_0,y_0)$. Écrire les développements limités d'ordre 2 de $V(x_0 \pm \mathrm{d}x, y_0)$ et de $V(x, y_0 \pm \mathrm{d}y)$.

   a- En déduire qu'en posant $\mathrm{d}x = \mathrm{d}y = 1$, le Laplacien de V vérifie :

   $$\Delta V \simeq V(x_0+\mathrm{d}x, y_0) + V(x_0-\mathrm{d}x, y_0) + V(x_0, y_0+\mathrm{d}y) + V(x_0, y_0-\mathrm{d}y) - 4\,V(x_0,y_0)$$

   b- Désormais, la fonction $V(x,y)$ sera remplacée par une matrice (encore appelée ici *tableau*). Montrer que l'élément $V[i][j]$ de ce tableau (à la ligne i et à la colonne j), vérifiant l'équation de Laplace (1), est donné par la relation :

   $$V[i][j] = \frac{V[i-1][j] + V[i+1][j] + V[i][j-1] + V[i][j+1]}{4} \qquad (2)$$

   Ainsi, pour calculer l'élément $V[i][j]$, il suffira de connaître les quatre autres éléments qui l'encadrent dans le tableau.
   L'objectif du script ci-dessous est la mise en œuvre de cette méthode.

2. Dans ce script :
   - on définit au préalable deux tableaux de dimension $N \times N$ (par exemple 60×60) ; chaque élément du tableau `tab[i][j]` contient initialement une valeur de potentiel (arbitrairement 10 partout et 0 aux bords) et une valeur `B[i][j]` indique le statut de `tab[i][j]` : `B[i][j]=1` lorsque `tab[i][j]` est une condition limite (donc immuable) et `B[i][j]=0` si `tab[i][j]` peut être modifié par la relation (2) :

```
08| def inittab():# Donne la valeur de V[i][j]=tab[i][j] et son statut
09|     B=np.zeros((N,N),bool)# B[i][j]=0 partout
10|     tab=np.zeros((N,N))+10# tab[i][j]=10 partout
11|     B[:,0]=1# B=1 au bord de gauche
12|     B[:,N-1]=1# B=1 au bord de droite
13|     B[0,:]=1# B=1 au bord supérieur
14|     B[N-1,:]=1# B=1 au bord inférieur
15|
16|     tab[:,0]=V0# Potentiel au bord de gauche
17|     tab[:,N-1]=V0# Potentiel au bord de droite
18|     tab[0,:]=V0# Potentiel au bord supérieur
19|     tab[N-1,:]=V0# Potentiel au bord inférieur
20|     return (tab,B)
```

   - la formule (2) sera appliquée à chaque élément `tab[i][j]` du tableau `tab`, sauf aux éléments immuables repérés par `B[i][j]=1`. À chaque itération, on calcule l'écart maximum $e_{\max}$ entre le nouveau tableau (appelé `temp`) et le tableau `tab` avant l'opération (2). La fonction `itera(tab)` retourne ainsi le nouveau tableau de potentiel et son écart par rapport au précédent [1]

1. L'instruction `temp=tab.copy()` permet de garder les matrices `temp` et `tab` indépendantes. En utilisant l'instruction `temp=tab`, toute modification d'une matrice se reporterait sur l'autre matrice et l'écart `temp-tab` serait toujours nul !


```
22| def itera(tab):#
23|     ecart=0
24|     temp=tab.copy()# tab ne sera pas affecté par temp
25|     for i in range(N):
26|         for j in range(N):
27|             if B[i,j]==0:
28|                 temp[i][j]=(tab[i-1][j]+tab[i+1][j]+tab[i][j-1]+tab[i]
[j+1])/4
29|                 ecart=max(ecart,abs(temp[i][j]-tab[i][j]))
30|     return (temp,ecart)
```

   - l'opération précédente est recommencée jusqu'à ce que $e_{\max}$ soit inférieur à une valeur `eps` (choisie ici arbitrairement égale à 0,3 : compromis entre précision et temps de calcul) :

```
37| while (itera(pot)[1])>eps:# La récurrence est appliquée jusqu'à ce que
e<eps
38|     pot=itera(pot)[0]
```

Dans sa configuration de base, le script s'écrit ainsi :

```
1  from math import*
2  import matplotlib.pyplot as
3  plt import numpy as np
4
5  V0=0
6  V1=200
7
8  def inittab():# Donne la valeur de V[i][j]=tab[i][j] et son statut
9      B=np.zeros((N,N),bool)# B[i][j]=0 partout
10     tab=np.zeros((N,N))+10# tab[i][j]=10 partout
11     B[:,0]=1# B=1 au bord de gauche de la matrice
12     B[:,N-1]=1# B=1 au bord de droite
13     B[0,:]=1# B=1 au bord supérieur
14     B[N-1,:]=1# B=1 au bord inférieur
15     tab[:,0]=V0# Potentiel au bord de gauche
16     tab[:,N-1]=V0# Potentiel au bord de droite
17     tab[0,:]=V0# Potentiel au bord supérieur
18     tab[N-1,:]=V0# Potentiel au bord inférieur
19     return (tab,B)
20
21 def itera(tab):
22     ecart=0
23     temp=tab.copy()# tab ne sera pas affecté par temp
24     for i in range(N):
25         for j in range(N):
26             if B[i,j]==0:
27                 temp[i][j]=(tab[i-1][j]+tab[i+1][j]+tab[i][j-1]+tab[i][j+1])/4
28                 ecart=max(ecart,abs(temp[i][j]-tab[i][j]))
29     return (temp,ecart)
30
31 N=60# Dimensions du tableau
32 eps=0.3# Ecart maximum entre deux itérations
33 pot=inittab()[0]# Le potentiel prend sa valeur initiale
34 B=inittab()[1]# Les bords sont définis dans la matrice B
35
36 while (itera(pot)[1])>eps:# La récurrence est appliquée jusqu'à ce que e<eps
37     pot=itera(pot)[0]
38
39 def graphe(B,f,a=0,nb_equipot=25):#Représentation de 25 équipotentielles
40     N,N=B.shape# Les bords du domaine apparaissent comme un cadre noir
41     plt.figure()
42     plt.imshow(B,origin='lower',cmap='binary',interpolation='nearest')
43     if (a==1):# Si a=1 les niveaux de potentiel sont colorés
44         plt.imshow(f,origin='lower')
45         plt.colorbar()
46     x=np.linspace(0,N-1,N)# L'axe x contient N valeurs (cases)
47     y=np.linspace(0,N-1,N)# L'axe y contient N valeurs
48     cont=plt.contour(y,x,f,nb_equipot,colors='k')# Trace les équipotentielles
49     plt.clabel(cont,fmt='%1.1f')# fmt contrôle l'affichage sur les courbes
50 plt.show()
51 graphe(B,pot)
```

a- Rajouter au script précédent les instructions permettant de représenter les équipotentielles générées par une charge ponctuelle de potentiel $V_1 = 200$ V, située au centre de la figure (*cf.* figure 1).

b- Modifier le script de manière à représenter les équipotentielles générées par un dipôle électrostatique (*cf.* figure 2).

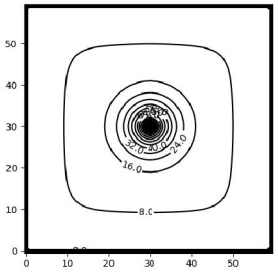


Figure 1

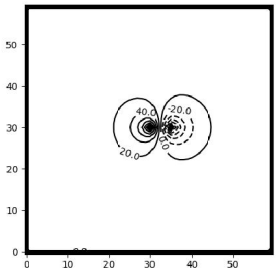


Figure 2

c- Modifier à nouveau le script afin de représenter les équipotentielles générées par un condensateur (*cf.* figure 3).

d- Enfin, ajouter les modifications nécessaires à la représentation de l'effet de pointe (*cf.* figure 4).

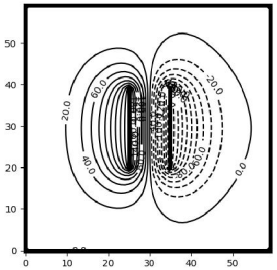


Figure 3

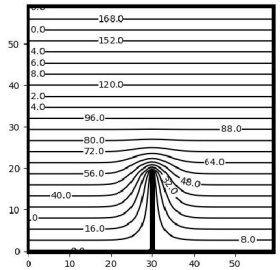


Figure 4