

**Résolution numérique de l'équation de Laplace**

# 1 Propriété du potentiel électrique dans le vide

## 1.1 Position du problème

Soit  $V(x, y, z)$  un potentiel électrique créé par une distribution de charges électriques, En dehors de la distribution (dans le vide de charges),  $V$  obéit à l'équation de Laplace :

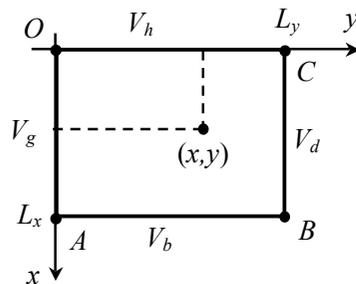
$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = 0$$

Dans la suite, nous allons considérer pour simplifier un problème à 2 dimensions, où le potentiel ne dépend que de  $x$  et  $y$  et satisfait à des conditions aux limites sur les bords d'un rectangle.

### Problème à résoudre :

Dans le plan  $(Oxy)$  un rectangle d'arêtes  $L_x$  et  $L_y$  a pour sommets les 4 points :  $O(0,0)$ ,  $A(L_x,0)$ ,  $B(L_x,L_y)$  et  $C(0,L_y)$ .

Une distribution appropriée de charges électriques fait en sorte que chacune des arêtes est portée à un potentiel électrique donné et constant :  $V_b$  correspond au bord horizontal  $x = L_x$ ,  $V_g$  au bord vertical  $y = 0$ ,  $V_h$  à  $x = 0$  et  $V_d$  à  $y = L_y$ .



L'intérieur du rectangle est supposé vide de charges ce qui implique qu'en tout point  $M(x, y)$  intérieur, le potentiel  $V(x, y)$  satisfait à l'équation de Laplace en dimension 2 :

$$\Delta V = \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0$$

## 1.2 Propriété fondamentale du potentiel

Si  $h$  est un réel positif, un développement limité du potentiel selon chaque variables  $x$  ou  $y$  donne, à l'ordre 3 inclus :

$$V(x \pm h, y) = V(x, y) \pm h \frac{\partial V}{\partial x}(x, y) + \frac{h^2}{2} \frac{\partial^2 V}{\partial x^2}(x, y) \pm \frac{h^3}{3!} \frac{\partial^3 V}{\partial x^3}(x, y) + o(h^3)$$

et

$$V(x, y \pm h) = V(x, y) \pm h \frac{\partial V}{\partial y}(x, y) + \frac{h^2}{2} \frac{\partial^2 V}{\partial y^2}(x, y) \pm \frac{h^3}{3!} \frac{\partial^3 V}{\partial y^3}(x, y) + o(h^3)$$

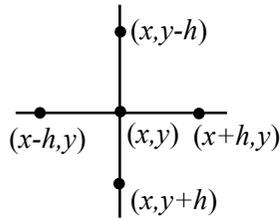
En sommant les quatre équations obtenus, nous obtenons :

$$V(x+h, y) + V(x-h, y) + V(x, y+h) + V(x, y-h) = 4V(x, y) + h^2 \left( \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} \right) + o(h^3)$$

Compte tenu de l'équation de Laplace, cette équation peut se mettre sous la forme :

$$V(x, y) = \frac{V(x+h, y) + V(x-h, y) + V(x, y+h) + V(x, y-h)}{4} + o(h^3)$$

Ainsi, dans la limite où  $h$  est suffisamment petit, le potentiel  $V(x, y)$  en  $(x, y)$  est la **moyenne** des potentiels en  $(x \pm h, y)$  et  $(x, y \pm h)$ . Cette propriété sera désignée par la suite sous le nom de **propriété fondamentale** du potentiel.



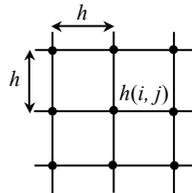
### 1.3 Méthode des éléments finis

Le principe de la résolution numérique de l'équation de Laplace est fondé sur la propriété précédente et sur la *méthode des éléments finis* :

- L'espace continu  $(Oxy)$  est remplacé par un maillage discret. Dans ce but, on choisit un **pas**  $h$  de sorte à définir deux entiers  $N_x$  et  $N_y \in \mathbb{N}$ , tels que :

$$L_x = h N_x \quad \text{et} \quad L_y = h N_y$$

- On ne s'intéresse qu'aux points dont les coordonnées vérifient  $(x, y) = (h i, h j) = h (i, j)$ , avec  $(i, j) \in \llbracket 0, N_x \rrbracket \times \llbracket 0, N_y \rrbracket$ . Le potentiel n'est calculé qu'en ces points et on pose :  $m_{ij} = V(h i, h j)$ . Cela revient donc à **échantillonner** le potentiel, non pas dans le temps comme avec un signal  $s(t)$ , mais dans l'espace<sup>1</sup> en ne prenant ses valeurs qu'en des points régulièrement répartis sur le plan  $(Oxy)$ .



1. Le pas  $h$  est l'équivalent spatial de la fréquence d'échantillonnage  $T_e$  pour les signaux temporels.

- Les coefficients  $m_{ij}$  peuvent alors être considérés comme les éléments d'une matrice  $M (N_x + 1, N_y + 1)$  de  $N_x + 1$  lignes et  $N_y + 1$  colonnes. Dans la suite, cette matrice sera appelée **matrice potentiel**. La **propriété fondamentale** du potentiel impose alors la relation :

$$m_{ij} = \frac{1}{4} [m_{i+1,j} + m_{i-1,j} + m_{i,j+1} + m_{i,j-1}] \quad (1)$$

pour tout  $i \in \llbracket 1, N_x - 1 \rrbracket$  et tout  $j \in \llbracket 1, N_y - 1 \rrbracket$ .

- Les conditions aux limites sur les arêtes du rectangle imposent de plus les valeurs de  $M$  sur sa première et dernière ligne et sur sa première et dernière colonne :

$$M = \begin{pmatrix} V_h & V_h & \dots & V_h & V_h \\ V_g & \dots & \dots & \dots & V_d \\ V_g & \dots & \dots & \dots & V_d \\ \dots & \dots & \dots & \dots & \dots \\ V_b & V_b & \dots & V_b & V_b \end{pmatrix}$$

Le problème consiste donc à trouver une matrice  $M$  ayant la forme ci-dessus et dont les coefficients  $m_{ij}$  satisfont à la relation (1).

## 2 Algorithme de résolution

### 2.1 Principe

L'algorithme procède par itération :

- **Initialisation** : on part d'une première matrice potentiel creuse  $M^{(0)}$  ayant les bonnes valeurs  $V_b, V_h, V_g$  et  $V_d$  sur ses bords et nulle à l'intérieur :  $m_{ij}^{(0)} = 0$  si  $1 \leq i \leq N_x - 1$  et  $1 \leq j \leq N_y - 1$ .

- **Itération** : on calcule une nouvelle matrice  $M^{(1)}$  en utilisant l'équation (1) mais en n'utilisant dans le membre de droite de l'équation que les coefficients  $m_{ij}^{(0)}$  de  $M^{(0)}$  :

$$m_{ij}^{(1)} = \frac{1}{4} \left[ m_{i+1,j}^{(0)} + m_{i-1,j}^{(0)} + m_{i,j+1}^{(0)} + m_{i,j-1}^{(0)} \right]$$

- On recommence en calculant les matrices  $M^{(2)}$ ,  $M^{(3)}$ , etc... Le processus finit par converger et tendre vers une matrice limite  $M$  qui donne le potentiel recherché sur les points du quadrillage.

## 2.2 Algorithme en pseudo-code

Voici l'algorithme dans du pseudo-code qui s'inspire du langage Python. On décide que le calcul est terminé lorsqu'à partir d'un certain rang  $k$ , les coefficients de la **matrice erreur** :  $E^{(k)} = M^{(k+1)} - M^{(k)}$  n'excèdent pas, en valeur absolue, une **erreur maximale**  $e_{max}$  qu'on se donne à l'avance :

$$\forall (i, j) \in \llbracket 0, N_x \rrbracket \times \llbracket 0, N_y \rrbracket, \quad |E_{ij}^{(k)}| \leq e_{max}$$

Cette condition est équivalente à dire que la norme infinie  $\|E^{(k)}\|_\infty$  de la matrice erreur doit être inférieure à  $e_{max}$  :

$$\|E^{(k)}\|_\infty = \max_{(i,j)} |E_{ij}^{(k)}| \leq e_{max}$$

### Algorithme calcul de M :

```

1  définir  $N_x, N_y$  : entiers
2  définir M0, M1, E : matrices ( $N_x, N_y$ )
3  # ----- Initialisations -----
4  M0 = matrice creuse avec conditions aux limites
5  erreur = 1
6  emax = 1e-3 # erreur maximale pour arrêt de l'itération
7  # ----- Début -----
8  tant que erreur > emax faire :
9      pour i allant de 1 à  $N_x - 1$  :
10         pour j allant de 1 à  $N_y - 1$  :
11             M1ij = 0.25 * (M0i+1,j + M0i-1,j + M0i,j+1 + M0i,j-1)
12         E = M1 - M0
13         erreur = || E ||∞
14         M0 = M1

```

## 2.3 Programme en Python

Les matrices peuvent être facilement gérées grâce à **numpy** :

```
import numpy as np
```

Quelques fonctions utiles :

- Si  $M$  est une matrice alors **abs**( $M$ ) est la matrice dont tous les coefficients sont les *valeurs absolues* des coefficients de  $M$ .
- On peut extraire l'élément maximal de toute matrice  $M$  en utilisant la fonction **np.amax**( $M$ ) qui retourne cet élément maximal. Vous disposez aussi de **np.amin**( $M$ ) qui retourne l'élément minimal de  $M$ .

Ceux qui sont intéressés par le programme Laplace.py le trouveront sur le site de la MP1.