

Méthodes Numériques et Calcul Scientifique

Contrôle Continu du 25 avril 2024

Tous les documents de l'UE sont autorisés

Les questions indépendantes sont signalées par le symbole ★

Durée : 3 h

Préciser en commentaire au début des fichiers source vos nom et prénom (sans accent).

- Créer, sous votre répertoire personnel, un répertoire de travail appelé `mncs-cc3` pour l'examen.
- Les scalaires réels seront représentés par des `float`. Les vecteurs (respectivement les matrices) seront déclarés comme des objets `VectorXf` (resp. `MatrixXf`) de la librairie `Eigen`.
- Les prototypes des fonctions sont imposés dans le texte. Ces prototypes seront écrits dans un fichier d'extension `.h` associé au fichier contenant les définitions des fonctions.
- Plusieurs fonctions utilisées dans le code sont fournies sur Moodle. L'énoncé précise quand utiliser ces fonctions.
- La fourniture d'un fichier `makefile` permettant de compiler le code sera valorisée.
- Ne pas oublier d'activer l'environnement python de l'UE : `conda activate MU4PY209`
- Une version couleur de l'énoncé est également disponible sur Moodle.

Plusieurs questions demandent des réponses à rédiger sur votre copie d'examen.

Étude d'un condensateur plan

I Introduction et notations

Le but du problème est d'étudier le potentiel électrique créé par un condensateur plan dans le vide.

Le condensateur plan est constitué de deux armatures, supposées infiniment étendues dans la direction z , et portées aux potentiels électriques $+V_0$ et $-V_0$ (cf. Figure 1). La simulation numérique est effectuée dans le plan (Oxy) et s'appuie sur la symétrie du problème par rapport au plan médian des deux armatures. On ne simulera donc que l'espace grisé sur la figure 1, de dimension $L_x \times L_y$.

Par symétrie, le potentiel est nul dans le plan médian des armatures. On impose de plus un potentiel nul sur les trois autres bords du domaine simulé, ce qui revient à faire l'hypothèse que ces bords sont situés infiniment loin du condensateur.

Dans l'ensemble de l'espace simulé, le potentiel électrique $V(x, y)$ vérifie l'équation de Poisson :

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

Afin de résoudre numériquement cette équation, l'intérieur de l'espace simulé est discrétisé en nl points d'abscisses $(x_i)_{0 \leq i \leq nl-1}$ selon l'axe x orienté du haut vers le bas, et nc points d'ordonnées $(y_j)_{0 \leq j \leq nc-1}$ selon l'axe y orienté de la gauche vers la droite (cf Figure 2). Le pas du maillage est noté p_{in} :

$$p_{in} = \frac{L_x}{nl + 1} = \frac{L_y}{nc + 1} \quad (2)$$

Les $nl \times nc$ potentiels en chacun des points du maillage $(V(x_i, y_j))$ sont les inconnues du problème à déterminer numériquement. Ces inconnues sont rangés dans un vecteur \mathbf{V} d'indice k ($0 \leq k \leq nl \times nc - 1$) représenté en rouge sur la Figure 2 et vérifiant :

$$k = i \times nc + j \quad (3)$$

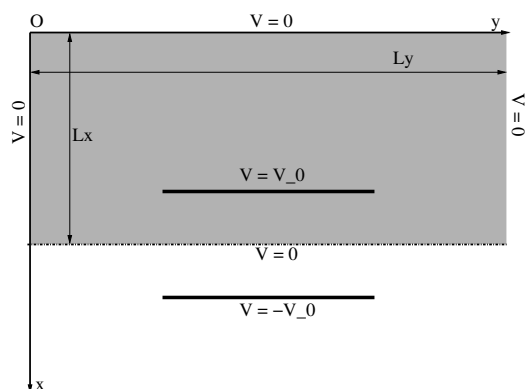


FIGURE 1 – Schéma du condensateur plan : axes utilisés, armatures (gras), et axe de symétrie (tiretés). L'axe z est perpendiculaire au plan de la figure. La simulation numérique n'est effectuée que dans la partie grisée du plan ($0xy$).

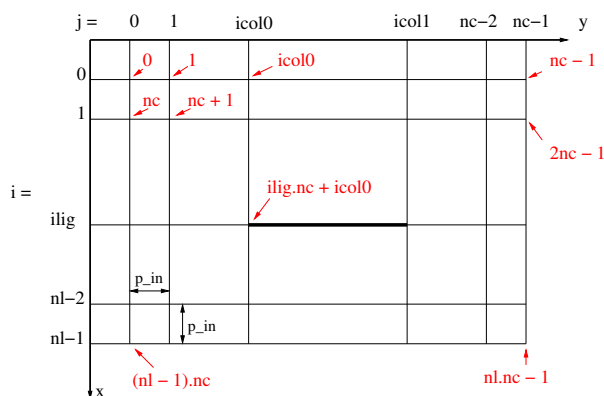


FIGURE 2 – Représentation du domaine simulé. Les indices de ligne (i) et de colonnes (j) des points du maillage sont mentionnés le long de l'axe x et y respectivement. Les potentiels en chaque point du maillage sont rangés dans le vecteur V ordonné comme indiqué par les indices en rouge sur la figure.

Dans ce maillage, l'armature de potentiel $+V_0$ est situé à la ligne d'indice i_{lig} , entre les colonnes d'indices i_{col0} et i_{col1} .

La résolution numérique de l'équation (1) revient ainsi à trouver le vecteur V , solution du système d'équations linéaires :

$$L_{2D} V = B \quad (4)$$

où L_{2D} est la matrice du Laplacien en deux dimensions et B , le vecteur second membre associé aux potentiels imposés (sur les bords du domaine ainsi que sur l'armature du condensateur).

On notera que le potentiel des points du maillage correspondant à l'armature du condensateur est déjà fixé. Ces points sont toutefois laissés dans le vecteur V afin de conserver la structure par bloc de la matrice L_{2D} .

II Mise en place de la simulation

Dans toute la suite du problème, on travaillera dans un domaine carré : $L_x = L_y = L$ et par (2) $nl = nc$. On conservera toutefois dans le programme les deux variables nl et nc pour le rendre facilement apte à représenter un domaine rectangulaire.

★ **Q 1 :** Écrire le début du programme principal `main.cpp`, afin de :

- déclarer les variables nl , nc , i_{lig} , i_{col0} , i_{col1} , L , p_{in} et V_0 ,
- demander à l'utilisateur la valeur de nl , imposer $nc = nl$ ainsi que $L = 6$,
- calculer p_{in} en utilisant l'une des deux égalités de l'équation (2),
- déclarer la matrice L_{2D} et les vecteurs V et B (sans préciser leur taille à ce stade),
- imposer $i_{\text{lig}} = 2(nl + 1)/3 - 1$, $i_{\text{col0}} = (nc + 1)/3 - 1$, et $i_{\text{col1}} = 2(nc + 1)/3 - 1$,
- imposer enfin $V_0 = 5$.

★ **Q 2 :** On cherche ici à construire la matrice du Laplacien à deux dimensions, sans prendre en compte dans cette question la présence de l'armature du condensateur. Écrire dans le fichier

`matrice.cpp` la fonction `matl2d` de prototype :

```
Eigen::MatrixXf matl2d(const int nl, const int nc);
```

Cette fonction renvoie la matrice L_{2D} et peut faire appel à d'autres fonctions de votre choix. Pour rappel :

$$L_{2D} = \left(\begin{array}{cccc|cccc|cccc} -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 \\ \hline \leftarrow nc \rightarrow & & \leftarrow nc \rightarrow & & \leftarrow nc \rightarrow & & \leftarrow nc \rightarrow & & \leftarrow nc \rightarrow & & \leftarrow nc \rightarrow & \end{array} \right) \begin{array}{l} \uparrow \\ nl \text{ blocs} \\ \text{de taille} \\ nc \times nc \\ \downarrow \end{array} \quad (5)$$

lorsque $nl = 3$ et $nc = 4$.

Q 3 : Poursuivre le programme principal pour appeler `matl2d` et vérifier (pour nl petit) que la matrice L_{2D} est bien construite.

★ **Q 4 :** La fonction fournie `insert_armature.cpp` reçoit comme premier argument la matrice L_{2D} et la modifie pour prendre en compte la présence de l'armature du condensateur. Expliquer précisément le but des modifications effectuées.

★ **Q 5 :** Inclure le code de la fonction `insert_armature` dans votre fichier `matrice.cpp` et poursuivre le programme principal pour appeler `insert_armature`. On pourra afficher la matrice L_{2D} modifiée pour $nl = 5$

★ **Q 6 :** Écrire dans le fichier `condlim.cpp` la fonction `sm_condensateur` de prototype :

```
Eigen::VectorXf sm_condensateur(const int nl, const int nc,
    const int ilig, const int icol0, const int icol1, const float V_0);
```

Cette fonction renvoie le vecteur B de l'équation (4). On rappelle que $V = 0$ sur les bords du domaine, et que l'armature du condensateur est au potentiel V_0 .

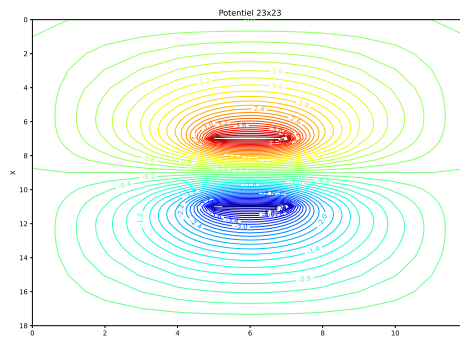
Q 7 : Poursuivre le programme principal pour appeler `sm_condensateur`. On s'assurera, que dans le cas $nl = 5$:

$$B = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \mid 0 \ 0 \ 0 \ 0 \ 0 \ 0 \mid 0 \ 0 \ 0 \ 0 \ 0 \ 0 \mid 0 \ 5 \ 5 \ 5 \ 0 \mid 0 \ 0 \ 0 \ 0 \ 0 \ 0)$$

Q 8 : Poursuivre le programme principal pour résoudre le système d'équations (4) à l'aide de la décomposition QR de la matrice L_{2D} (et sans en exploiter le caractère creux).

Q 9 : Finaliser le programme principal pour appeler la fonction `ecrit_pot1` fournie dans le fichier `io.cpp` (on choisira d'appeler le fichier de sortie `laplace.dat`)

Exécuter le code en choisissant $nl = 23$. Visualiser le potentiel simulé avec le script python `plot-laplace-sym.py` également fourni. Ce script représente le potentiel dans l'espace simulé, ainsi que dans son symétrique par rapport au plan médian des armatures. S'assurer que le résultat obtenu est similaire à celui de la figure ci-contre.



★ **Q 10 :** Mettre en place dans le code le chronométrage de la résolution du système d'équations.

Q 11 : Exécuter le code en choisissant successivement $nl = 23$, $nl = 29$ et $nl = 35$, et noter les durées de résolution. Commenter les durées mesurées. À quelle durée pourrait-on s'attendre si l'on choisissait $nl = 59$?

III Simulation avec un maillage de pas non-constant

Éloigner significativement les bords du domaine avec un maillage de pas constant nécessiterait un nombre trop important de points. On décide donc d'étudier dans cette partie un maillage de pas non-constant, tel que celui représenté sur la Figure 3. Le domaine de simulation est donc désormais divisé en deux parties :

- un domaine intérieur avec un maillage similaire à celui de la partie II : p_{in} dans les deux directions de l'espace,
- un domaine extérieur où le pas du maillage est soit p_{in} soit p_{out} , et n'est pas forcément le même dans les deux directions.

Plus précisément, comme indiqué sur la Figure 4, dans le domaine extérieur :

- le pas du maillage selon x est p_{out} si $i < i_{ligext}$, et p_{in} sinon,
- le pas du maillage selon y est p_{out} si $j < i_{col0ext}$ ou $j > i_{col1ext}$, et p_{in} sinon.

Le but de cette section est d'introduire ce nouveau maillage dans la simulation. Les premières questions portent sur l'expression du laplacien dans un domaine de pas variable.

★ **Q 12 :** On choisit $p_{out} = \alpha p_{in}$, où α est un réel ≥ 1 . On se place de plus dans la partie du domaine extérieur telle que $i < i_{ligext}$ et $i_{col0ext} < j < i_{col1ext}$ (« centre-haut »). Le pas du maillage est donc αp_{in} selon x , et p_{in} selon y . En utilisant un développement de Taylor de la fonction potentiel $V(x, y)$, exprimer les dérivées secondes $\partial^2 V / \partial x^2$ et $\partial^2 V / \partial y^2$ en fonction de $V(x, y)$, $V(x \pm \alpha p_{in}, y)$, $V(x, y \pm p_{in})$, p_{in} et de α .

Q 13 : Écrire $\Delta V = 0$ dans cette partie du domaine extérieur en fonction des mêmes variables, et en déduire que les composantes du vecteur V vérifient l'équation linéaire :

$$V(k + nc) + V(k - nc) + \alpha^2 V(k + 1) + \alpha^2 V(k - 1) - 2(\alpha^2 + 1)V(k) = 0 \quad (6)$$

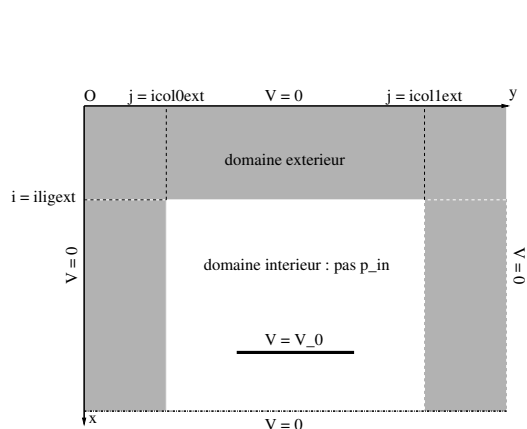


FIGURE 3 – Domaines de la simulation à pas non-constant. Le domaine intérieur où le pas du maillage est p_{in} dans les deux directions est représenté en blanc. Le domaine extérieur est représenté en grisé.

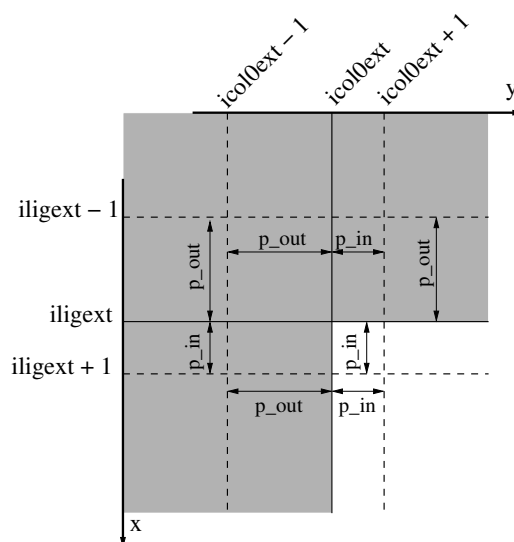


FIGURE 4 – Zoom de la grille autour de la ligne $i = i_{\text{ligext}}$ et de la colonne $j = j_{\text{col0ext}}$. Noter que le pas du maillage varie dans le domaine extérieur : p_{in} ou p_{out} .

★ **Q 14 :** Que retrouve-t-on dans l'équation (6) lorsque $\alpha = 1$?

★ **Q 15 :** Inclure la fonction `insert_domaine_ext` dans votre fichier `matrice.cpp`. Cette fonction modifie la matrice L_{2D} pour introduire le domaine extérieur. Compléter le code de cette fonction pour modifier la matrice du laplacien dans la même partie « centre-haut » du domaine extérieur.

* **Q 16 :** Expliquer les modifications de la matrice du laplacien effectuée par la fonction `insert_domain_ext` dans le coin ($i = i_{ligext}, j = i_{col0ext}$).

★ **Q 17 :** Expliquer enfin pourquoi le code de la fonction `insert_domaine_ext` ne modifie pas la matrice L_{2D} pour les points du domaine extérieur tels que $i < i_{\text{ligext}}$ et $j < i_{\text{col0ext}}$?

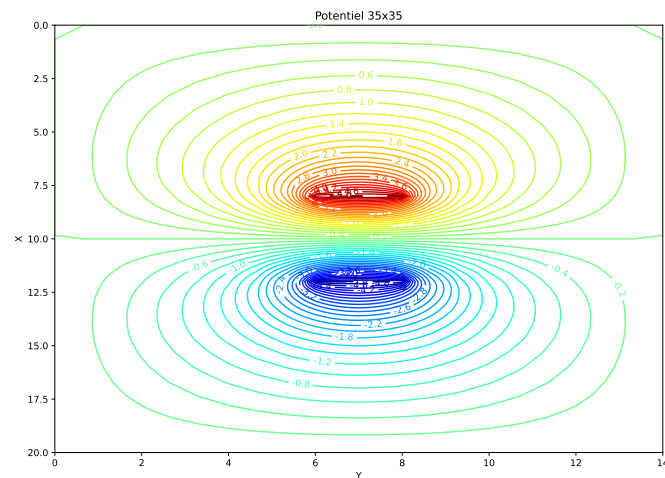
★ Q 18 : Compléter le programme principal pour :

- demander à l'utilisateur la valeur des variables : i_{ligext} , i_{col0ext} , i_{col1ext} et α ,
- calculer p_{out} ,
- appeler `insert_domaine_ext`,
- et enregistrer les résultats de la simulation en appelant la fonction `ecrit_pot2` fournie dans `io.cpp` (on choisira de nouveau d'appeler le fichier de sortie `laplace.dat`)

NB : Ne pas supprimer les éventuelles parties du code correspondant à la partie II qui ne sont plus utiles : les passer en commentaire.

Q 19 : Exécuter le code en choisissant $nl = 35$, $i_{\text{ligext}} = i_{\text{col0ext}} = 7$, $i_{\text{col1ext}} = 27$ et $\alpha = 4$. Visualiser le champ calculé avec le script python et s'assurer que le résultat est similaire à celui de la figure ci-dessous.

Q 20 : Comparer la durée d'exécution avec celle de la simulation à pas constant et commenter.



IV Champ électrique

Le champ électrique \vec{E} est solution de :

$$\vec{E} = -\overrightarrow{\text{grad}} V \quad (7)$$

★ **Q 21** : Sans chercher à implémenter le calcul, détailler comment il serait possible d'estimer numériquement \vec{E} ? Préciser le calcul sur les points du domaine en contact avec les bords.

En fin d'épreuve,

- Ouvrir un terminal dans le répertoire d'examen `mnscs-cc3` (utiliser éventuellement la commande `cd` pour vous déplacer dans l'arborescence de fichiers)
 - Lancer la création d'une archive de ce répertoire `tar -cvzf cc3_[NOM].tgz ./`
 - Déposer le fichier `cc3_[NOM].tgz` dans la boîte de dépôt sur Moodle
 - Rendre votre copie d'examen
-