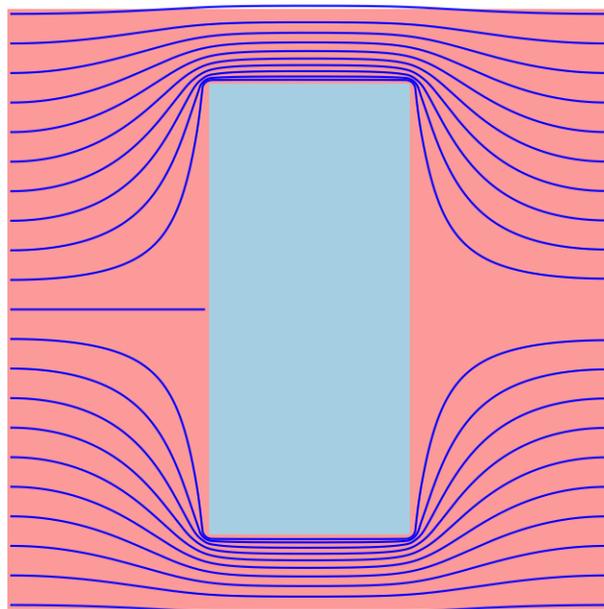


Projet numérique

Écoulements potentiels



Référent du projet : Vincent Ballenegger
Institut UTINAM, Université de Franche-Comté

1 Introduction et description du projet

De nombreux écoulements de fluide d'intérêt pratique sont **potentiels**, c'est-à-dire à la fois incompressible ($\operatorname{div} \vec{v} = 0$) et irrotationnel ($\operatorname{rot} \vec{v} = 0$). Tout écoulement d'un fluide parfait incompressible arrivant avec une vitesse uniforme sur un obstacle est par exemple potentiel. La théorie des écoulements potentiels joue notamment un rôle essentiel en aérodynamique pour calculer l'écoulement de l'air autour d'une aile.¹

Le but de ce projet est d'écrire un programme calculant plusieurs propriétés physiques d'écoulements potentiels dans diverses géométries. On souhaite déterminer en particulier le champ des vitesses, le champ de pression et les lignes de courant. Par souci de simplicité, on ne considère que des écoulements potentiels **plans** et on admet les hypothèses suivantes :

- le fluide est parfait (i.e. non-visqueux) ;
- le fluide est incompressible ;
- l'écoulement est irrotationnel \Rightarrow les particules-fluides ne tournent pas sur elles-mêmes ;
- l'écoulement est bi-dimensionnel dans le plan Oxy \Rightarrow invariance par translation dans la direction z.

Le champ des vitesses d'un écoulement potentiel dérive d'un potentiel des vitesses $\phi(x, y)$ via

$$\vec{v} = -\vec{\nabla} \phi(x, y). \quad (1)$$

On peut déduire de cette relation que la vitesse du fluide est perpendiculaire aux courbes de niveau de la fonction ϕ (la vitesse a en effet une composante nulle dans la direction où ϕ ne varie pas). Comme démontré en cours de Mécanique des fluides en licence, le potentiel des vitesses obéit à l'équation de Laplace

$$\Delta \phi(x, y) = 0 \quad \forall (x, y) \in D \quad (2)$$

en tout point (x, y) d'un domaine $D \subset \mathbb{R}^2$. Il s'agit donc de résoudre l'équation de Laplace dans un domaine spécifié, avec des conditions sur la solution ϕ adéquates sur les bords du domaine D . La condition aux limites en un point sur le bord peut être

- vitesse fixée : \vec{v} connu (débit entrant si $\vec{v} \cdot \vec{n} < 0$, sortant si $\vec{v} \cdot \vec{n} > 0$) ;
- paroi infranchissable : $\vec{v} \cdot \vec{n} = 0$;
- paroi franchissable avec vitesse non spécifiée.

Ici \vec{n} est le vecteur unité normal à la surface, orienté vers l'extérieur du domaine.

On envisagera différentes géométries pour l'écoulement (forme du domaine D), en particulier :

1. un canal rectiligne simple ;
2. un canal avec un élargissement ;
3. un canal avec un coude ;
4. un canal rectiligne contenant un obstacle (la forme de cet obstacle est laissée à votre choix).

La première géométrie, le canal rectiligne, ne présente pas d'intérêt physique notable, mais permet de tester le bon fonctionnement du programme dans un cas très simple.

La résolution numérique de l'équation de Laplace sera réalisée en utilisant la **méthode des différences finies**. Dans cette méthode, on introduit une grille régulière, composée de $N = N_x \times N_y$ points, couvrant tout le domaine D , puis on discrétise l'équation continue (2) sur cette grille. La somme de dérivées secondes $\Delta \phi$ est alors remplacée par des différences de ϕ entre des points voisins de la grille. La discrétisation de l'équation (2) conduit à un système de N équations linéaires couplées pour les valeurs de ϕ en chaque point de la grille, système que l'on peut écrire sous la forme $A\vec{x} = \vec{b}$, où A est une matrice de taille $N \times N$. La solution de ce système pourra être obtenue en utilisant un algorithme standard de résolution de systèmes linéaires.

1. Les effets visqueux sont confinés à une petite couche vers les surfaces et la théorie des écoulements potentiels peut être corrigée pour tenir compte de ces effets. La vitesse de l'air reste de plus typiquement petite par rapport à la vitesse du son (écoulement incompressible).

2 Cahier des charges

Il s'agit de réaliser un programme présentant les fonctionnalités suivantes :

Version alpha Le programme permet de calculer l'écoulement d'un fluide dans le cas de la géométrie 2 (canal avec élargissement) et affiche les résultats suivants :

1. une représentation de la géométrie du domaine D ;
2. le champ des potentiels avec tracé des lignes isopotentielles (courbes de niveaux) ;
3. la représentation graphique des vitesses dans tout le domaine, et les valeurs numériques des vitesses d'entrée et de sortie du fluide ;
4. le tracé des lignes de courant (via la commande `streamplot()` de matplotlib) ;
5. le champ de pression avec tracé des lignes isobares.

La spécification du débit entrant, du nombre de points de la grille dans les 2 directions, et du pas h (en m) de la grille (ou l'extension L_x et L_y du domaine physique étudié) se fait via des variables faciles à identifier dans le programme.

Comme le système est invariant par translation dans la direction z , définir le débit comme un débit par unité de hauteur (en m^2/s). Un même graphique peut regrouper plusieurs résultats mentionnés ci-dessus.

Version beta

Le programme

1. permet d'étudier les écoulements dans les 4 géométries listées en page 2. Les divers paramètres de ces géométries (largeur et longueur des diverses sections du canal, dimension et position de l'obstacle) doivent être faciles à ajuster en modifiant des variables du programme ;
Pour optimiser la lisibilité du code, définir des fonctions pour définir les différentes géométries.
2. calcule les lignes de courant en effectuant une intégration explicite des trajectoires de quelques particules-fluides ;
3. fonctionne correctement dans le cas d'un domaine physique non-carré ($L_x \neq L_y$).

Pour chaque géométrie, le programme doit afficher, en plus des graphiques de la version alpha,

1. le graphique du profil des vitesses dans une ou des section(s) jugées intéressante(s) ;
2. un graphique des lignes de courant obtenues par intégration de quelques trajectoires (ne plus utiliser `streamplot()`, mais tracer les trajectoires calculées) ;
3. un tracé des pressions s'exerçant sur l'obstacle et sur le déviateur du coude, complété avec le calcul de la force résultante s'exerçant sur le déviateur du coude et sur l'obstacle.

Version gold

Une telle version peut se distinguer notamment par

1. Entrée des divers paramètres (géométrie, débit entrant, taille de grille, ...) en arguments du programme ;
Exemple : appel du programme via `./prog_ecoul geometrie=1 Q=200 Nx=60 Ny=60`
2. Graphiques particulièrement soignés au niveau de l'esthétique et de la lisibilité ;
3. Affichage des éventuels points de vitesse nulle (point d'arrêt, point de stagnation) sur le graphique des lignes de courant ;
4. Intégration des lignes de courant par une méthode d'intégration plus précise que la méthode d'Euler ;
5. Graphiques montrant l'influence de certains paramètres sur certains résultats, par exemple l'influence de la finesse de la grille de discrétisation, ou la qualité de l'intégrateur temporel pour les lignes de courant.

→ Tous les graphiques générés par le programme doivent être enregistrés au format PDF avec un nom de fichier descriptif.

→ Le programme doit fonctionner correctement même dans le cas où la grille utilisée pour la discrétisation contient très peu de points.

→ Soigner la présentation et la lisibilité du code.

Environnement de développement

Le programme doit être écrit dans le langage **Python**. Il fera usage des bibliothèques numériques **numpy** (tableaux multidimensionnels) et **matplotlib** (graphiques). La bibliothèque de méthodes numériques **scipy** pourra se révéler utile, bien que non indispensable à la réalisation de ce projet.

3 Tâches à réaliser

1. Comprendre la structure du système d'équations résultant de la discrétisation de l'équation de Laplace ainsi que les conditions sur $\phi(x, y)$ à appliquer aux limites du domaine : voir sections 4 et 5 ;
2. Apprendre à utiliser les commandes `pyplot.imshow()` et `numpy.meshgrid()` : voir l'annexe et la documentation de la bibliothèque graphique *matplotlib* ;
3. Ecrire le programme Python demandé dans le cahier des charges ;
4. Dans le rapport technique :
 - Lister les limitations du programme : conditions à respecter lors de la définition de la géométrie du domaine de l'écoulement, phénomènes physiques non pris en compte, etc.
 - Inclure une analyse critique des résultats (phénomènes observés, interprétation).

4 Eléments théoriques nécessaires au projet

4.1 Discrétisation par différences finies

On expose les principes de base de la méthode des différences finies permettant de discrétiser l'équation (2) sur une grille pour obtenir un système d'équations linéaires. L'idée est de remplacer dans l'équation, en chaque point P de la grille, les dérivées dans $\Delta\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2}$ par des différences du potentiel ϕ entre des points de la grille proches de P.

Tout d'abord, il s'agit de discrétiser le domaine d'écoulement en introduisant une grille régulière couvrant tout le domaine. Dans le cadre de ce projet, on considère une grille de maille carrée de dimension h , alignée avec un repère Cartésien Oxy. Le point de coordonnées (i, j) sur la grille a pour coordonnées cartésiennes $(x, y) = (ih, jh)$. En un point (x, y) , la dérivée première dans la direction x peut être estimée par l'une des formules de différences finies suivantes

$$v_x(i, j) = -\frac{\partial\phi}{\partial x}\Big|_{i,j} \simeq - \begin{cases} \frac{\phi_{i+1,j} - \phi_{i,j}}{h} & \text{(différence progressive)} \\ \frac{\phi_{i,j} - \phi_{i-1,j}}{h} & \text{(différence rétrograde)} \\ \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h} & \text{(différence centrée de pas } 2h) \\ \frac{\phi_{i+\frac{1}{2},j} - \phi_{i-\frac{1}{2},j}}{h} & \text{(différence centrée de pas } h : \text{ points hors-grille!)} \end{cases} \quad (3)$$

La dérivée seconde peut être approximée en appliquant 2 fois la formule d'une différence centrée de pas h

$$\frac{\partial^2\phi}{\partial x^2}\Big|_{i,j} \simeq \frac{\frac{\partial\phi}{\partial x}\Big|_{i+1/2,j} - \frac{\partial\phi}{\partial x}\Big|_{i-1/2,j}}{h} \simeq \frac{\phi_{i-1,j} - 2\phi_{i,j} + \phi_{i+1,j}}{h^2}. \quad (4)$$

Des formules similaires permettent d'obtenir les dérivées dans la direction y .

Les formules de différences finies centrées ci-dessus permettent d'approximer l'équation de Laplace $\Delta\phi = 0$ en un point (i, j) de la grille par

$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = 0 \quad \Longrightarrow \quad \frac{\phi_{i-1,j} - 2\phi_{i,j} + \phi_{i+1,j}}{h^2} + \frac{\phi_{i,j-1} - 2\phi_{i,j} + \phi_{i,j+1}}{h^2} = 0 \quad (5)$$

$$\Longrightarrow \quad \phi_{i,j} = \frac{\phi_{i-1,j} + \phi_{i+1,j} + \phi_{i,j-1} + \phi_{i,j+1}}{4}. \quad (6)$$

Le potentiel en un point intérieur (i, j) de la grille correspond ainsi à la moyenne des potentiels des 4 points adjacents. La formule (6) ne s'appliquant qu'aux points intérieurs, comment calculer le potentiel sur les bords de la grille? La réponse est dans l'imposition de conditions aux limites.

4.2 Conditions aux limites

Aux limites du domaine D , des conditions doivent être imposées sur la fonction ϕ en se référant à la physique du problème. Il faut en effet imposer comment se comportent les particules fluides aux bords du domaine : paroi infranchissable ou non, débit imposé ou non.

Les conditions aux limites peuvent être de deux types : on peut imposer la valeur de la fonction (condition de Dirichlet) ou sa dérivée (condition de Neumann).

Condition aux limites de Neumann Une condition de type Neumann impose la valeur du gradient de la fonction. Comme le gradient de ϕ donne la vitesse du fluide (au signe près), une telle condition revient à imposer des composantes de la vitesse.

Supposons que l'on désire imposer une vitesse d'entrée (ou de sortie) v_x selon l'axe x sur le bord *gauche* du domaine. On introduit alors une condition de Neumann

$$-\frac{\partial\phi}{\partial x}\Big|_{x=0} = v_x \quad \Longrightarrow \quad -\frac{\phi_{i+1,j} - \phi_{i,j}}{h} = v_x \quad \Longrightarrow \quad \phi_{i,j} = \phi_{i+1,j} + v_x h. \quad (7)$$

Pour imposer une vitesse sur le bord *droit*, on procéderait de même, mais en utilisant la formule de différence finie rétrograde.

Une condition de paroi imperméable ($\vec{v} \cdot \vec{n} = 0$) se traduit par l'imposition d'un gradient nul pour ϕ dans la direction orthogonale à la paroi. Par exemple, si le bord inférieur de la grille est en contact avec une paroi imperméable, on impose la condition $\frac{\partial\phi}{\partial y}\Big|_{y=0} = 0$, qui peut être discrétisée en

$$\frac{\partial\phi}{\partial y}\Big|_{y=0} = 0 \quad \Longrightarrow \quad \frac{\phi_{i,j+1} - \phi_{i,j}}{h} = 0 \quad \Longrightarrow \quad \phi_{i,j+1} = \phi_{i,j}. \quad (8)$$

D'après l'éq. (8), les deux points de la grille juste au-dessus de la paroi sont au même potentiel. En pratique, on n'implémentera en fait pas l'éq. (8) exactement de cette manière, pour la raison donnée à la section 5.1, mais comme une moyenne similaire à l'éq. (6) [voir éq. (11)]. Les formules (7) et (8) [ou plutôt (11)] montrent comment traiter les points de la grille sur les bords du domaine dans le cas d'une paroi imperméable ou d'une entrée/sortie avec vitesse fixée.

Condition aux limites de Dirichlet Une condition de type Dirichlet impose la valeur de la fonction à la frontière. Cela revient à imposer une valeur de référence ϕ_{ref} au potentiel. Comme la vitesse du fluide est perpendiculaire aux lignes équipotentielles $\phi = \text{cste}$, les lignes de courant traversent la ligne de potentiel ϕ_{ref} perpendiculairement et ces lignes de courant sont réparties uniformément. Cela revient dès lors à imposer un passage uniforme du fluide, mais sans fixer la vitesse du fluide.

Unicité de la solution Il est nécessaire d'imposer une condition aux limites de type Dirichlet en au moins une frontière du domaine. En effet, l'équation de Laplace étant une équation présentant uniquement des dérivées secondes, la solution ne serait déterminée qu'à une constante près si l'on

n'imposait que des conditions de type Neumann. Imposer une condition de Dirichlet revient à imposer un niveau de référence pour la fonction ϕ .

Pour illustrer l'importance de fixer un niveau de référence, considérons le cas d'un domaine discrétisé en 3 mailles et sur lequel on résout l'équation de Laplace en 1 dimension avec uniquement des conditions de Neumann imposant des dérivées nulles aux deux extrémités.



On a alors les relations suivantes

- éq. de Laplace pour le point intérieur : $\phi_2 = \frac{\phi_1 + \phi_3}{2}$ (moyenne des 2 cases adjacentes)
- condition de Neumann pour la cellule 1 : $\phi_1 = \phi_2$
- condition de Neumann pour la cellule 3 : $\phi_2 = \phi_3$

Les équations de ce système sont linéairement dépendantes, et il ne peut donc pas être résolu. Pour éviter ce problème, il faut imposer une condition de Dirichlet à au moins l'une des deux extrémités.

4.3 Principes de résolution

Calcul du potentiel des vitesses

La première étape de la résolution consiste à discrétiser le domaine sur une grille régulière ($N_x \times N_y$ mailles carrées). Certaines mailles contiennent du fluide, tandis que d'autres sont inaccessibles à celui-ci (mur, paroi).

La deuxième étape consiste à attribuer une inconnue ϕ_m à chaque case contenant du fluide ($m = 1, \dots, N_{\text{fluide}}$, où N_{fluide} est le nombre de cases occupées par le fluide).

Troisièmement, on construit le système d'équations $A\vec{x} = \vec{b}$ en appliquant l'équation (6) à tous les points intérieurs, et les équations des conditions aux limites (Dirichlet ou Neumann) aux points aux bords du domaine. Chaque ligne de la matrice A contient au maximum 5 termes non nuls. Les composantes du vecteur \vec{x} sont les inconnues ϕ_m et les composantes du vecteur \vec{b} sont toutes nulles, sauf lorsque l'indice m correspond à une cellule obéissant à une condition de Dirichlet ($\phi_m = \phi_{\text{ref}}$).

La dernière étape consiste à résoudre le système linéaire pour obtenir la solution pour le potentiel des vitesses en chaque point de la grille occupé par du fluide. Cette étape ne pose pas de difficulté particulière car on peut utiliser une méthode de résolution standard de systèmes linéaires, par exemple l'algorithme proposé par la commande `solve()` du module `numpy`. Les points délicats résident plus dans les 3 premières étapes que dans la résolution à proprement parler.

Calcul du champ des vitesses

Lorsque le potentiel $\phi(x, y)$ des vitesses est connu en chaque point de la grille, les composantes (v_x, v_y) de la vitesse peuvent être obtenues, en chaque point, via la formule $\vec{v} = -\vec{\nabla}\phi$. Il convient naturellement d'approximer le gradient par une formule de différence finie, voir eq. (3).

Remarque : La fonction `numpy.gradient()` peut rendre service.

Calcul du champ de pression

Dans un écoulement potentiel plan, la densité volumique d'énergie $P + \frac{1}{2}\rho v^2$ (où $P = P(x, y)$ est la pression du fluide) reste constante en tout point de l'écoulement, et non seulement le long d'une ligne de courant comme dans le théorème de Bernoulli (cf. cours de licence de Mécanique des fluides). Si la vitesse et la pression du fluide sont connues à l'entrée, et que le champ des vitesses a été déterminé, la formule $P + \frac{1}{2}\rho v^2 = cste$ permet de calculer la pression en tout point de l'écoulement.

Calcul d'une ligne de courant

Dans un écoulement *stationnaire*, c'est-à-dire indépendant du temps, les lignes de courant se confondent avec les trajectoires des particules de fluide. La trajectoire $\vec{r}(t) = (x(t), y(t))$ obéit à

l'équation

$$\frac{d}{dt}\vec{r}(t) = \vec{v}(x(t), y(t)), \quad (9)$$

par définition du champ des vitesses. Une ligne de courant s'obtient en intégrant numériquement cette équation différentielle du premier ordre, en tenant compte de la condition initiale (à savoir que la particule est située en un point (x_0, y_0) au temps $t = 0$). Dans le cadre de ce projet, on pourra effectuer cette intégration en utilisant la méthode d'Euler.

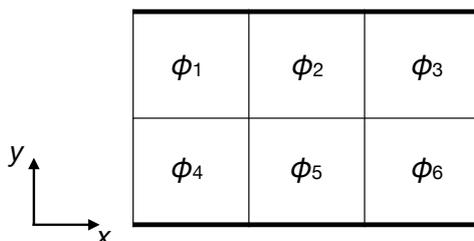
Un bref exposé de la méthode d'Euler et de l'algorithme correspondant peut être trouvé par exemple au §4.2.1 du document <https://femto-physique.fr/omp/pdf/euler.pdf>.

Le calcul d'une ligne de courant nécessite de connaître la vitesse du fluide en un point quelconque du domaine. Cette vitesse peut être obtenue par interpolation² des vitesses connues aux noeuds de la grille. La commande `scipy.interpolate.interp2d()` est utile pour effectuer une telle interpolation.

5 Considérations pratiques

5.1 Discrétisation du domaine et condition de paroi imperméable

Prenons comme exemple la discrétisation d'un canal de section constante, de longueur 3 m et de largeur 2 m, et choisissons de discrétiser le domaine occupé par le fluide sur une grille très grossière composée de 3×2 cellules.



Il est important de réaliser qu'il convient de **discrétiser l'équation de Laplace aux centres des différentes cellules de la grille, et non aux points où les lignes de la grille se croisent** (voir dessin ci-dessus). Ici, toutes les 6 cellules sont aux bords du domaine de l'écoulement. À l'entrée de la conduite (cases ϕ_1 et ϕ_4), on peut appliquer la condition aux limites de Neumann (7) pour imposer la vitesse du fluide. À la sortie (cases ϕ_3 et ϕ_6), il convient d'imposer une condition aux limites de Dirichlet : $\phi_3 = \phi_6 = \phi_{\text{ref}}$, où ϕ_{ref} est une valeur arbitraire. En appliquant la condition de paroi imperméable [eq. (8)] aux 6 cases en contact avec une paroi, on obtient les égalités : $\phi_1 = \phi_4$, $\phi_2 = \phi_5$ et $\phi_3 = \phi_6$. Le système d'équations obtenu ainsi ne peut pas être résolu, car rien ne détermine la valeur commune de $\phi_2 = \phi_5$.³

Pour résoudre ce problème de manière générale, il convient de discrétiser la condition de Neumann $\partial\phi/\partial y|_{y=0} = 0$ un peu différemment, en utilisant une différence rétrograde plutôt que la différence progressive dans l'éq. (8). Pour la case 5, on obtient alors que ϕ_5 est égal à $\tilde{\phi}_5$ où $\tilde{\phi}_5$ est le potentiel dans une case fictive située à l'extérieur du domaine, en-dessous de la case 5. L'égalité $\phi_5 = \tilde{\phi}_5$ s'interprète comme l'imposition d'un gradient nul entre ces deux cases, et donc à l'emplacement réel de la paroi. Avec l'ajout de la case fictive $\tilde{\phi}_5$, la case ϕ_5 peut être traitée comme une case intérieure. Selon l'éq. (6), on a alors

$$\phi_5 = \frac{\phi_4 + \phi_2 + \phi_6 + \tilde{\phi}_5}{4} \quad \text{où} \quad \tilde{\phi}_5 = \phi_5, \quad (10)$$

d'où

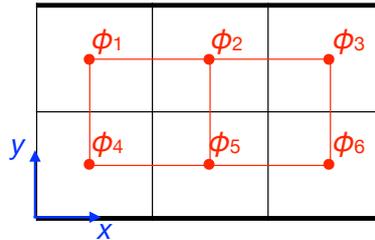
$$\phi_5 = \frac{\phi_4 + \phi_2 + \phi_6}{3}. \quad (11)$$

2. par exemple interpolation bi-linéaire, voir https://fr.wikipedia.org/wiki/Interpolation_bilin%C3%A9aire

3. Noter que ce problème ne se poserait pas si la grille contenait au moins 3 cellules dans la direction y .

Le potentiel ϕ_5 est donc donné par la moyenne des potentiels des 3 cases adjacentes de fluide. Pour une case de coin qui serait en contact avec 2 parois imperméables, on trouverait similairement que le potentiel est donné par la moyenne des 2 cases adjacentes. On en conclut que les cases en contact avec une ou plusieurs paroi(s) peuvent être traitées comme les cases intérieures, c'est-à-dire par une moyenne de la forme (6), mais où la moyenne n'inclut que les cases (réelles) adjacentes contenant du fluide.

Comme on cherche à déterminer le potentiel des vitesses aux centres des cases, il est commode de travailler non pas avec la grille représentée en noir, mais avec la grille représentée en rouge dans la figure ci-dessous :



Les 6 potentiels à déterminer correspondent alors aux noeuds de cette grille, dont l'origine est située en $(x, y) = (h/2, h/2)$ où h est le pas de la grille. Dans le programme Python, le champ des potentiels peut être représenté par un tableau numpy à 2 dimensions, équivalent à une matrice.

Remarque : Le module numpy adopte la même convention pour les indices des composantes $M_{i,j}$ d'une matrice M qu'en algèbre linéaire : le premier indice d'un élément $M[i, j]$ d'un tableau numpy M correspond au numéro de ligne, et le second indice au numéro de colonne. Avec le repère Cartésien introduit dans la figure ci-dessus, l'axe i du tableau correspond donc à la direction $-y$, et l'axe des j à la direction x . Il faut ainsi tenir compte d'une éventuelle différence d'orientation du repère "matriciel" (i, j) par rapport au repère Cartésien Oxy lors de l'écriture du programme. Noter qu'il est possible d'inverser l'orientation de l'axe matriciel i lors de la représentation du contenu de la matrice sous forme d'une image (voir annexe). Alternativement, on peut choisir d'orienter différemment le repère Oxy (indiquer dans le rapport technique l'orientation choisie du repère).

5.2 Matrice décrivant la géométrie de l'écoulement

Il est utile d'introduire un tableau à deux dimensions décrivant la géométrie du domaine étudié et les conditions aux limites à appliquer. La dimension de ce tableau (nommé par exemple \mathbf{G}) est identique à celle de la matrice contenant les valeurs $\phi(i, j)$ du potentiel. Un exemple de contenu d'un tableau \mathbf{G} décrivant une conduite présentant un coude à angle droit est représenté ci-dessous.

| | | | |
|---|---|---|---|
| 0 | 3 | 3 | 3 |
| 0 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 |

0 = mur
 1 = case de fluide
 2 = case de fluide avec débit entrant fixé
 3 = case de fluide avec débit sortant

L'élément $\mathbf{G}[i, j]$ contient ainsi un nombre entre 0 et 3 en fonction du contenu de la case et du type de condition aux limites à appliquer en cet endroit. Les emplacements des parois (représentées ci-dessus par des traits épais) peuvent se déduire de la matrice \mathbf{G} : tous les bords des cases aux limites du domaine sont supposées être des parois imperméables, à moins qu'il ne s'agisse d'une case avec un débit de fluide entrant ou sortant.

Il n'est pas obligatoire de suivre à la lettre le système de codage proposé dans cet exemple : vous pouvez introduire votre propre manière de définir la géométrie du problème.

5.3 Construction de la matrice A et des vecteurs \vec{x} et \vec{b}

La dimension de la matrice A est $N_{\text{fluide}} \times N_{\text{fluide}}$, où N_{fluide} est le nombre de cases de la grille contenant du fluide.

Pour construire la matrice A , il faut d'abord parcourir toutes les cases pour compter le nombre de cases de fluide et pour attribuer un numéro d'inconnue (m) à chaque case (l'inconnue est le potentiel de la case, noté ϕ_m , $m = 1, \dots, N_{\text{fluide}}$). On doit être capable de faire facilement le lien entre un numéro d'inconnue m et les coordonnées (i, j) de la case correspondante.

Remarque : une manière d'établir ce lien consiste à définir un tableau dans lequel la $m^{\text{ème}}$ ligne contient les coordonnées i et j de la case correspondante.

Inversément, il faut aussi pouvoir faire facilement le lien entre une case de coordonnées (i, j) données et l'index m d'inconnue associé à cette case. Il est commode d'introduire une matrice pour stocker en mémoire cette association $(i, j) \rightarrow m$.

Exemple de matrice de numération des cases de fluide :

| | | | |
|---|---|---|----|
| | 8 | 9 | 10 |
| | 5 | 6 | 7 |
| 1 | 2 | 3 | 4 |

La structure générale (et donc la construction) de la matrice A se comprend ensuite bien à partir d'un exemple. Considérons le cas d'un canal coudé à angle droit avec les cases de fluide numérotées comme ci-dessus. L'application des formules introduites aux sections 4.1, 4.2 et 5.1 conduit au système linéaire suivant d'équations :

$$\begin{bmatrix}
 -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & -3 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & -3 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & -2 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & -3 & 1 & 0 & 1 & 0 & 0 \\
 0 & 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\
 0 & 0 & 0 & 1 & 0 & 1 & -3 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{bmatrix}
 \begin{bmatrix}
 \phi_1 \\
 \phi_2 \\
 \phi_3 \\
 \phi_4 \\
 \phi_5 \\
 \phi_6 \\
 \phi_7 \\
 \phi_8 \\
 \phi_9 \\
 \phi_{10}
 \end{bmatrix}
 =
 \begin{bmatrix}
 -hv_x \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 \phi_{\text{ref}} \\
 \phi_{\text{ref}} \\
 \phi_{\text{ref}}
 \end{bmatrix}
 \quad (12)$$

La $i^{\text{ème}}$ ligne de ce système correspond à une équation pour la case numéro i . On reconnaît sur les 3 dernières lignes des conditions aux limites de Dirichlet. La 1re ligne correspond à une condition aux limites de Neumann fixant la vitesse dans la case 1 à v_x [éq. (7)]. Les autres lignes correspondent à des moyennes sur les cases fluides adjacentes [éq. (6) et (11)]. Pour les cases de fluide autres que celles à l'entrée ou à la sortie, le nombre sur la diagonale de la matrice A est égal à l'opposé du nombre de cases de fluide adjacentes.

6 Structure du programme à réaliser

La structure du programme suivra les différentes étapes de résolution expliquées dans la section 4.3. Les différents objets à introduire sont indiqués la section 2 (cahier des charges) et dans la section 5 (considérations pratiques).

Dans le but d'augmenter la lisibilité du code, il est recommandé d'introduire des fonctions (quand cela est utile) et d'utiliser la notation "slice" pour accéder/modifier des portions de tableaux.

7 Algorithmes à utiliser

- Méthode de résolution d'un système linéaire : `numpy.solve()`
- Calcul du gradient d'un champ scalaire : `numpy.gradient()` ou propre code utilisant des différences finies
- Interpolation d'une grandeur à partir de ses valeurs connues sur une grille : `scipy.interpolate.interp2d()` (ou propre code). La commande `interp2d()` ne fonctionne bien que si le champ ne contient aucune valeur `nan` (**n**ot **a** **n**umber).
- Intégration d'une équation différentielle d'ordre 1 par la méthode d'Euler : propre code
Voir §4.2.1 du document <https://femto-physique.fr/omp/pdf/euler.pdf>.
- Intégration d'une fonction sur un intervalle : propre code utilisant la méthode de votre choix (méthode des rectangles, de Simpson, ...) ou usage de la bibliothèque `scipy.integrate`.
Voir <https://docs.scipy.org/doc/scipy/reference/integrate.html>.

8 Annexe

Le document `matrice_image2D.ipynb` (un notebook jupyter) montre

- comment une matrice peut être représentée graphiquement dans un plan sous forme d'une carte de couleurs à l'aide de la bibliothèque **matplotlib** ;
- comment l'orientation de l'axe matriciel vertical `i` peut être inversée, si souhaité, dans la représentation graphique ;
- la création d'une grille de points via la commande `numpy.meshgrid()`.