

# TP2 Résolution des équations d'Euler isothermes compressibles en 2D avec la méthode des volumes finis

Sara Essakhi et Dorian Hugonnet

December 2025

## Q1- Equations d'Euler isothermes

### Hypothèses

- Dans le contexte aérodynamique, les forces de gravité sont négligeables par rapport aux forces de pression et d'inertie. Donc  $\rho g = 0_{\mathbb{R}^2}$
- Cas isotherme : L'équation de l'énergie est donc inutile pour résoudre la dynamique du fluide et la pression s'écrit  $p = a^2 \rho$
- Ecoulement 2D : Donc la vitesse  $\vec{u}$  a pour composantes  $(u, v)$

Sous ces hypothèses, le système d'Euler s'écrit de la manière suivante :

$$\left\{ \begin{array}{l} \partial_t \rho + \nabla \cdot (\rho \vec{u}) = 0 \\ \partial_t (\rho \vec{u}) + \nabla \cdot (\rho \vec{u} \otimes \vec{u} + a^2 \rho I_2) = 0 \\ \\ \text{Conservation de Masse} \\ \partial_t \rho + \partial_x (\rho u_x) + \partial_y (\rho u_y) = 0 \\ \\ \text{Conservation de Quantité de Mouvement en x} \\ \partial_t (\rho u_x) + \partial_x (\rho u_x^2 + a^2 \rho) + \partial_y (\rho u_x u_y) = 0 \\ \\ \text{Conservation de Quantité de Mouvement en y} \\ \partial_t (\rho u_y) + \partial_x (\rho u_y u_x) + \partial_y (\rho u_y^2 + a^2 \rho) = 0 \end{array} \right. \quad (1)$$

## Q2- Passage à la forme vectorielle conservative

L'objectif est d'écrire le système sous la forme d'une loi de conservation générique :

$$\frac{\partial \vec{U}}{\partial t} + \nabla \cdot \mathbf{F}(\vec{U}) = \vec{S}(\vec{U}) \quad (2)$$

Où  $\vec{U}$  est le vecteur des variables conservatives :

$$\vec{U} = \begin{pmatrix} \rho \\ \rho u_x \\ \rho u_y \end{pmatrix}$$

et  $\vec{S}(\vec{U}) = 0_{\mathbb{R}^3}$  est le terme source.

En 2D, l'opérateur divergence appliqué au tenseur des flux  $\mathbf{F} = (\vec{F}_x, \vec{F}_y)$  se développe comme :

$$\nabla \cdot \mathbf{F}(\vec{U}) = \frac{\partial \vec{F}_x(\vec{U})}{\partial x} + \frac{\partial \vec{F}_y(\vec{U})}{\partial y} \quad (3)$$

En identifiant terme à terme avec le système (1), on définit les vecteurs de flux physique suivants, qui seront utilisés par le solveur de Riemann :

$$\vec{F}_x(\vec{U}) = \begin{pmatrix} \rho u_x \\ \rho u_x^2 + a^2 \rho \\ \rho u_x u_y \end{pmatrix}, \quad \vec{F}_y(\vec{U}) = \begin{pmatrix} \rho u_y \\ \rho u_y u_x \\ \rho u_y^2 + a^2 \rho \end{pmatrix}$$

## Résolution numérique par la méthode des Volumes Finis

Dans cette étape, nous voulons résoudre le système d'équations d'Euler sous forme conservative. À chaque instant  $t^n$  et pour chaque cellule  $K$ , on approxime la solution  $\vec{U}(x, y, t^n)$  par une valeur moyenne constante  $\vec{U}_K^n$  :

$$\vec{U}_K^n = \frac{1}{|K|} \int_K \vec{U}(x, y, t^n) dK \quad (4)$$

représentant l'état moyen dans la cellule  $K$  au temps  $t^n$ , où  $|K|$  est la surface de la cellule. Le but est alors de calculer l'évolution de cette valeur moyenne au pas de temps suivant  $t^{n+1}$ , et donc d'obtenir  $\vec{U}_K^{n+1}$ .

Notons le vecteur d'état conservatif :

$$\vec{U}_K^{n+1} = \begin{pmatrix} \rho \\ \rho u_x \\ \rho u_y \end{pmatrix}_K^{n+1} \quad (5)$$

Nous intégrons l'équation de conservation sur une cellule de contrôle  $K$ . En l'absence de termes sources (gravité négligée), cela nous donne :

$$\int_K \frac{\partial \vec{U}}{\partial t} dK + \int_K \nabla \cdot \mathbf{F}(\vec{U}) dK = 0 \quad (6)$$

En appliquant le théorème de la divergence (Green-Ostrogradsky), nous transformons l'intégrale de volume du terme de divergence en une intégrale de surface sur le bord  $\partial K$  de la cellule :

$$\int_K \frac{\partial \vec{U}}{\partial t} dK + \oint_{\partial K} \mathbf{F}(\vec{U}) \cdot \vec{n} d\Gamma = 0 \quad (7)$$

où  $\vec{n}$  est le vecteur normal unitaire sortant de la frontière.

Nous effectuons maintenant une discréttisation en temps et en espace.

- **Pour le premier terme** (dérivée temporelle), on suppose  $\vec{U}$  constant dans la maille et on approche la dérivée par une différence finie en temps (schéma d'Euler explicite d'ordre 1) :

$$\int_K \frac{\partial \vec{U}}{\partial t} dK \approx |K| \frac{\vec{U}_K^{n+1} - \vec{U}_K^n}{\Delta t} \quad (8)$$

- **Pour le second terme** (flux), on approxime le flux sortant par les bords à l'aide d'une somme sur les arêtes  $e \in \partial K$  (les 3 faces du triangle) :

$$\oint_{\partial K} \mathbf{F}(\vec{U}) \cdot \vec{n} d\Gamma \approx \sum_{e \in \partial K} \vec{\Phi}(\vec{U}_K^n, \vec{U}_{K_e}^n, \vec{n}_e) \cdot L_e \quad (9)$$

où :

- $L_e$  est la longueur de l'arête  $e$ .
- $\vec{U}_{K_e}^n$  est l'état dans la maille voisine partageant l'arête  $e$ .
- $\vec{\Phi}$  est le **flux numérique** à travers l'interface, solution d'un problème de Riemann (calculé ici par le solveur de Roe).

En rassemblant les termes, on obtient la formule de mise à jour explicite de la solution :

$$\vec{U}_K^{n+1} = \vec{U}_K^n - \frac{\Delta t}{|K|} \sum_{e \in \partial K} \vec{\Phi}(\vec{U}_K^n, \vec{U}_{K_e}^n, \vec{n}_e) \cdot L_e \quad (10)$$

### Q5- Conditions aux limites

D'après la fonction `boundary condition` de `solver.py`, la condition à appliquer en présence d'une paroi est la wall condition. Cette condition se traduit par le fait que la composante de la vitesse normale à la paroi doit être nulle sur la face frontière  $\vec{u} \cdot \vec{n} = 0$

où  $\vec{n}$  est le vecteur normal unitaire à la face de la paroi.

Pour imposer la condition numérique d'imperméabilité ( $\vec{u} \cdot \vec{n} = 0$  sur la face) en utilisant une cellule fictive (*ghost cell*), on applique une symétrie "miroir" par rapport au plan tangent de la paroi.

Le code de la fonction `wall_condition` explicite précisément cet état via la logique suivante :

```

qg[0] = qi[0]
qg[1:] = qi[1:] - 2*np.dot(qi[1:], n)*n

```

Voici le détail de l'état  $\mathbf{q}_g$  à appliquer dans la cellule fictive :

— **A. Pour la densité ( $\rho$ ) : Condition de Neumann homogène**

La densité dans la cellule fictive est égale à la densité dans la cellule interne adjacente :

$$\rho_{\text{fictive}} = \rho_{\text{interne}} \quad (11)$$

Cela assure qu'aucun gradient de densité normal à la paroi n'est artificiellement créé par la condition limite.

— **B. Pour la quantité de mouvement ( $\rho\vec{u}$ ) : Réflexion du vecteur**

La vitesse (et donc la quantité de mouvement) dans la cellule fictive doit être le reflet de la vitesse interne par rapport à la paroi :

- La composante **tangentielle** est conservée (condition de glissement).

- La composante **normale** est inversée (pour annuler la vitesse sur la face interface).

Si l'on note  $\vec{V} = \rho\vec{u}$  le vecteur quantité de mouvement, la formule vectorielle implémentée correspond à :

$$\vec{V}_{\text{ghost cell}} = \vec{V}_{\text{interne}} - 2(\vec{V}_{\text{interne}} \cdot \vec{n})\vec{n} \quad (12)$$

En résumé, le vecteur d'état complet de la cellule fictive s'écrit :

$$\mathbf{q}_{\text{ghost}} = \begin{pmatrix} \rho_{\text{interne}} \\ (\rho u_x)_{\text{interne}} - 2(\vec{V}_{\text{in}} \cdot \vec{n})n_x \\ (\rho u_y)_{\text{interne}} - 2(\vec{V}_{\text{in}} \cdot \vec{n})n_y \end{pmatrix} \quad (13)$$

## Q9- Résultats avec le maillage mesh/naca0012.msh

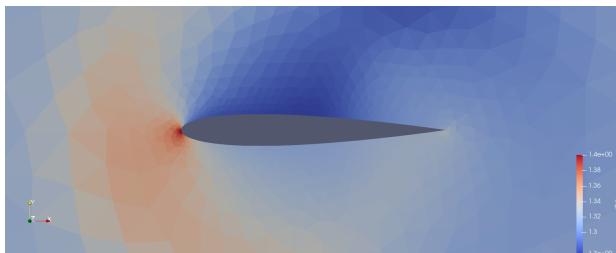


FIGURE 1 – Densité ( $\rho$ )

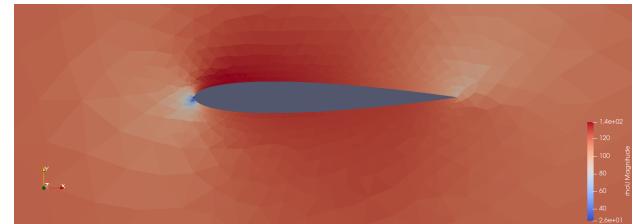


FIGURE 2 – Quantité de mouvement ( $\rho U$ )

Pour un nombre de Mach = 0.273, nous remaquons que :

— **Densité :**

- On observe une petite zone rouge intense **au bord d'attaque**. C'est le point d'arrêt où la vitesse du fluide s'annule quasi-instantanément. L'énergie cinétique est transformée en énergie de pression. C'est, donc, ici que la densité est maximale.
- **Au niveau de l'extrados**, la densité est minimale. C'est parce que le fluide contourne la courbure de l'aile et accélère fortement. Selon le principe de Bernoulli, une augmentation de vitesse s'accompagne d'une chute de pression et de densité, ce qui est vérifié ici.
- **Au niveau de l'intrados**, la densité est plus élevée que sur l'extrados. C'est cette différence de densité et donc de pression qui va induire la portance.

— **Quantité de mouvement :**

- Au niveau du bord d'attaque la vitesse devient nulle.
- L'extrados est la zone où la vitesse du fluide est maximale, chose qui maximise son énergie cinétique.

## Q10- Coefficient de pression $C_p$

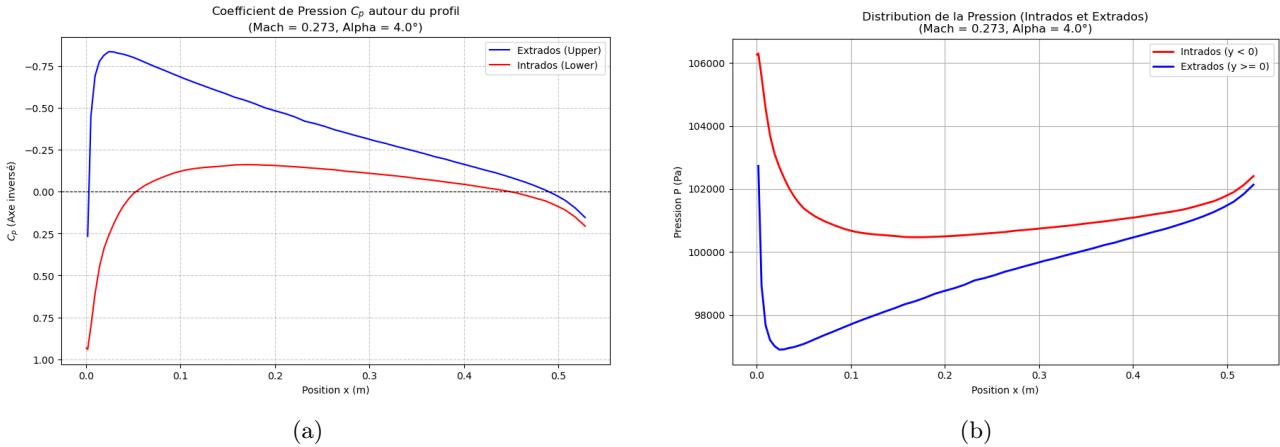


FIGURE 3

Ces deux figures valident la physique de l'écoulement obtenu :

- **Génération de Portance (Figure  $C_p$ )** : La distribution du coefficient de pression montre clairement le mécanisme de portance pour  $\alpha = 4^\circ$ . L'extrados (courbe bleue) présente un fort pic de succion au bord d'attaque ( $C_p \approx -0.85$ ), zone où le fluide accélère fortement. L'intrados (courbe rouge) reste en surpression relative sur la majorité de la corde. La surface entre les deux courbes représente la charge aérodynamique qui porte le profil.
- **Validation de la Convergence (Figure Pression)** : Le graphique de la pression absolue apporte la preuve formelle que la simulation a convergé vers une solution physique valide. Au bord de fuite ( $x \approx 0.53m$ ), les courbes d'intrados et d'extrados se rejoignent parfaitement (pas de saut de pression).

## Q11a- Influence de local time step

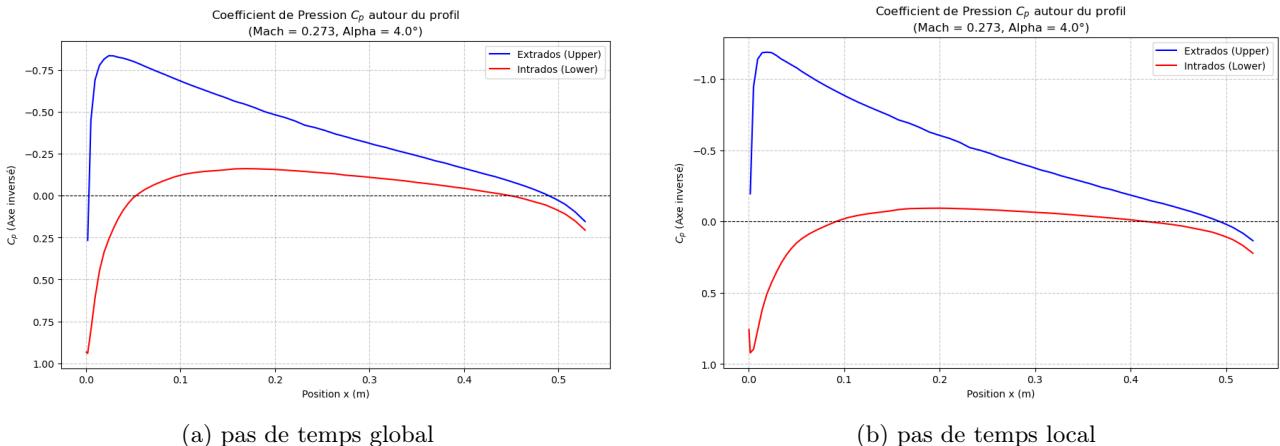


FIGURE 4 – Evolution du coefficient de portance pour deux pas de temps différents

Nous observons que dans la simulation avec un pas de temps local, le minimum de pression sur l'extrados est beaucoup plus marqué que pour la simulation avec le pas de temps global (-1.2 au lieu de -0.8). On peut en déduire, donc, que dans la simulation "Global", le fluide n'a pas encore eu le temps d'accélérer suffisamment sur le dessus de l'aile pour créer la dépression attendue. La simulation "Local", en revanche, capture bien cette forte accélération au bord d'attaque, essentielle pour générer la portance.

**Performance et Temps de Calcul** : Bien que le temps de calcul par itération soit identique ( $\simeq 18$  secondes pour 200 itérations) pour les deux méthodes, la stratégie à **pas de temps local** accélère considérablement la convergence vers l'état stationnaire.

Cette efficacité s'explique par la disparité des tailles de cellules dans le maillage non structuré.

- Dans l'approche **Globale**, le pas de temps  $\Delta t$  est unique et limité par la condition de stabilité de la plus petite cellule du domaine ( $\Delta t_{global} = \min_i \Delta t_i$ ). Cela "bride" inutilement les grandes cellules qui pourraient avancer plus vite.
- Dans l'approche **Locale**, chaque cellule  $i$  avance avec son propre pas de temps maximal admissible  $\Delta t_i$ . L'information se propage donc beaucoup plus rapidement à travers le domaine, réduisant le nombre d'itérations nécessaires.

Cependant, cette accélération est limitée par la condition de **stabilité CFL** (Courant-Friedrichs-Lowy). Pour un schéma explicite, le pas de temps doit respecter strictement le domaine de dépendance physique. Si le nombre CFL choisi est trop élevé (par exemple, ici la simulation diverge à CFL=0.5 ou plus ce qui est assez bas quand même), les oscillations numériques s'amplifient exponentiellement. Cela conduit à l'apparition de valeurs non physiques (densités ou pressions négatives), provoquant l'arrêt du code par une erreur d'exécution . Donc pour des CFL trop élevés, L'information numérique va plus vite que l'information physique ce qui fait diverger notre simulation.

## Q11b- Validation de la simulation (Preuve de convergence)

Paramètres de la simulation : - maillage naca0012.msh d'aile d'avion, CFL=0.5 (pour assurer la stabilité), M=0.273

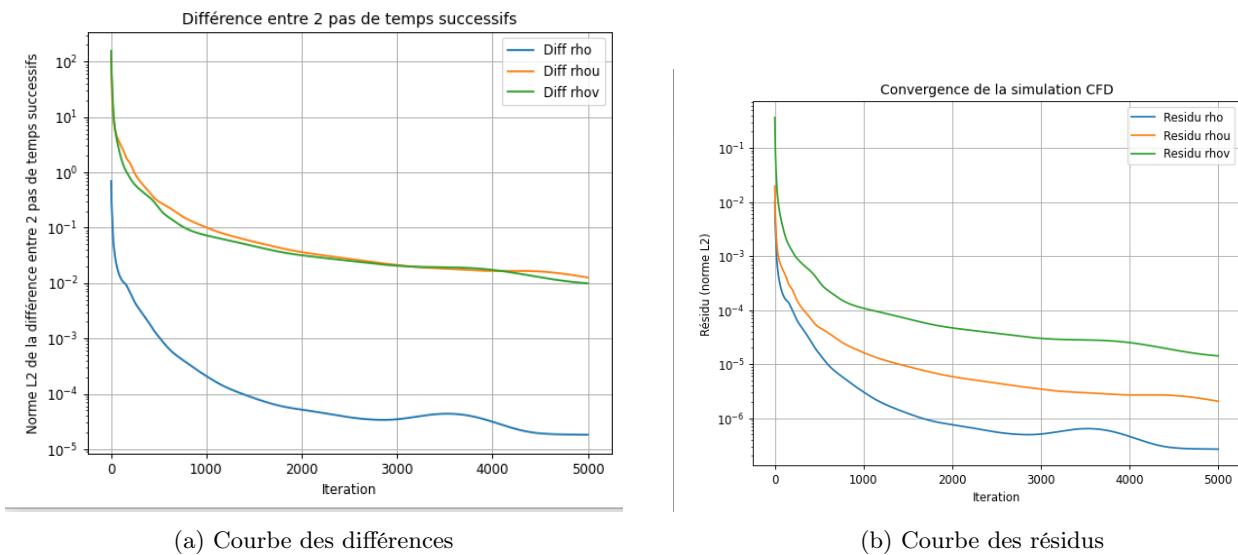


FIGURE 5 – Comparaison des deux courbes

Les figures ci-dessus illustrent l'historique de convergence de la simulation. La première représente la différence brute des variables conservatives entre deux pas de temps successifs, tandis que la seconde trace l'évolution de la norme  $L_2$  des résidus  $R^n$ , définie par la formule suivante :

$$R^n = \|\vec{U}^n - \vec{U}^{n-1}\|_2 = \sqrt{\sum_K (\vec{U}_K^n - \vec{U}_K^{n-1})^2} \quad (14)$$

On observe une décroissance monotone globale des résidus pour les trois variables conservatives ( $\rho, \rho u, \rho v$ ). Les résidus de densité, par exemple, passent d'environ  $10^{-1}$  à  $10^{-6}$ . Une telle chute de 5 à 6 ordres de grandeur constitue un excellent critère de convergence pour un schéma d'Euler explicite d'ordre 1.

Après 5000 itérations, les courbes atteignent un plateau et s'aplatissent. Cela indique que les variations temporelles de la solution numérique deviennent négligeables, confirmant ainsi que **l'état stationnaire est atteint**.

D'un point de vue physique, on remarque une disparité dans la convergence des variables :

- La densité  $\rho$  (courbe bleue) présente la meilleure convergence.
- La quantité de mouvement verticale  $\rho v$  (courbe verte) converge plus difficilement.

Ce comportement s'explique par la nature de l'écoulement, qui est principalement horizontal (selon l'axe X). Les vitesses  $v$  sont de faible amplitude, générées uniquement par la portance et le contournement du profil. Par conséquent, elles sont proportionnellement plus sensibles aux erreurs numériques et plus délicates à stabiliser que le flux principal. D'après la figure des résidus, nous observons la convergence qu'on considère à  $10^{-4}$

### Stabilité des coefficients (CL,CD)

## Q11c- Polaire de portance du profil ( $C_L$ vs Incidence)

### Calcul de la polaire de portance

Dans cette sous-partie, nous nous sommes intéressés au calcul de la polaire de portance du profil, c'est-à-dire l'évolution du coefficient de portance ( $C_L$ ) en fonction de l'angle d'incidence  $\alpha$ .

#### Cadre théorique

La force aérodynamique s'exerçant sur le profil résulte de l'intégration des forces de pression sur la surface. Dans notre simulation (fluide non-visqueux), nous négligeons les forces de frottement pariétal.

Les coefficients de force selon les axes du maillage ( $C_x$  selon la corde,  $C_y$  selon la normale) sont obtenus par intégration du coefficient de pression  $C_p$  sur le contour du profil  $\mathcal{S}$  :

$$C_x = \frac{1}{c} \oint_{\mathcal{S}} -C_p n_x dS \quad \text{et} \quad C_y = \frac{1}{c} \oint_{\mathcal{S}} -C_p n_y dS \quad (15)$$

où  $\vec{n} = (n_x, n_y)$  est la normale sortante à la surface et  $c$  la corde du profil.

Cependant, la portance ( $L$ ) et la traînée ( $D$ ) sont définies dans le repère aérodynamique, lié à la vitesse de l'écoulement infini  $V_\infty$ . Comme notre maillage reste fixe et que nous faisons varier l'angle du vecteur vitesse incident  $\alpha$ , il est nécessaire d'effectuer un changement de repère (une rotation) pour obtenir  $C_L$  et  $C_D$  :

$$\begin{pmatrix} C_D \\ C_L \end{pmatrix} = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} C_x \\ C_y \end{pmatrix} \quad (16)$$

Ce qui nous donne pour la portance :

$$C_L = C_y \cos \alpha - C_x \sin \alpha \quad (17)$$

#### Protocole numérique

Pour construire la polaire, nous avons réalisé une série de simulations en faisant varier l'incidence  $\alpha$ . Les paramètres de simulation sont les suivants :

- **Nombre de Mach** :  $M = 0.273$  (identique aux cas précédents).
- **Plage d'incidence** :  $\alpha \in [-2^\circ, 18^\circ]$  (discrétisée en 10 points).
- **Convergence** : Utilisation du pas de temps global.
- **Critère d'arrêt** : La simulation pour un angle donné s'arrête lorsque le résidu sur la densité  $\rho$  devient inférieur à  $10^{-4}$ .

#### Implémentation pratique

Dans le code de calcul, l'intégrale continue est remplacée par une somme discrète sur l'ensemble des faces appartenant à la frontière WALL. Pour chaque face  $k$  de longueur  $L_k$  et de normale  $\vec{n}_k$ , nous calculons la contribution locale à la force :

$$F_{x,\text{total}} = \sum_{k \in \text{WALL}} -C_{p,k} \cdot n_{x,k} \cdot L_k \quad (18)$$

Nous projetons ensuite ces forces globales dans le repère du vent comme décrit dans la partie théorique pour obtenir le  $C_L$  final de chaque angle. Le code source complet réalisant cette boucle est disponible dans le fichier `main.py`

## Analyse des résultats

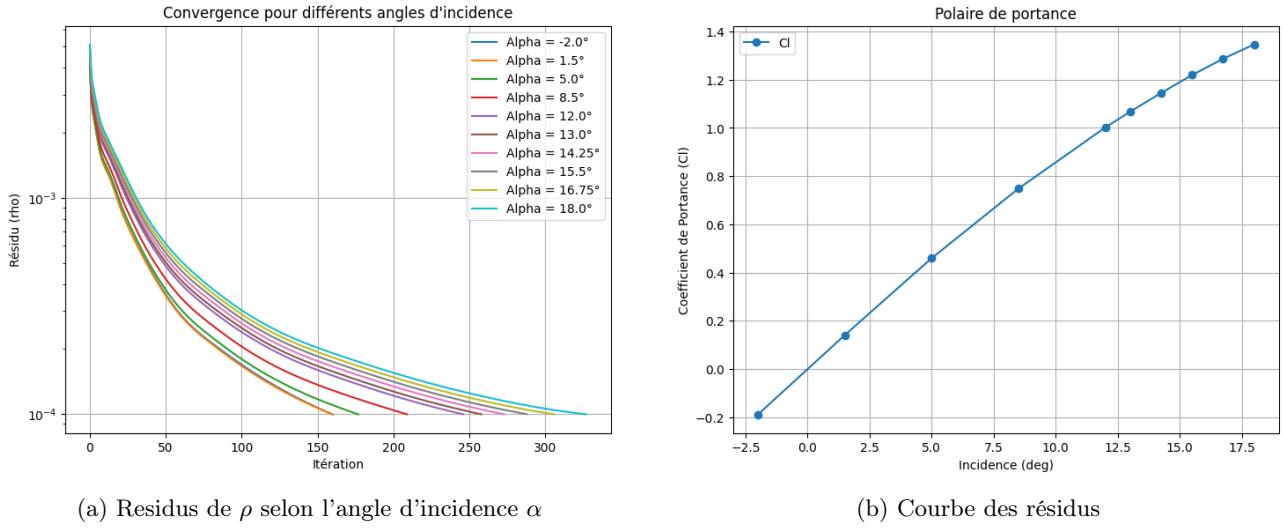


FIGURE 6 – Polaire de portance

### Analyse de la convergence

La Figure 6a montre l'évolution du résidu de la densité ( $\rho$ ) en échelle logarithmique.

Nous observons que pour tous les angles testés, le résidu décroît de manière monotone jusqu'au critère d'arrêt ( $10^{-4}$ ). Cependant, on remarque une corrélation directe entre l'incidence et le coût de calcul : plus l'angle d'incidence  $\alpha$  augmente, plus le nombre d'itérations nécessaire pour converger est élevé (de  $\approx 160$  itérations pour  $\alpha = -2^\circ$  à plus de 300 pour  $\alpha = 18^\circ$ ). Cela s'explique par la perturbation plus importante de l'écoulement à forte incidence, qui demande plus de temps au solveur pour se stabiliser.

### Analyse de la polaire de portance ( $C_L \propto \alpha$ )

La Figure 6b présente l'évolution du coefficient de portance.

- Zone linéaire : Pour les faibles incidences ( $\alpha < 10^\circ$ ), la courbe est quasi parfaitement linéaire. Cela valide notre simulation par rapport à la théorie des profils minces (où  $C_L \approx 2\pi\alpha$ ).
- Comportement aux grands angles : Au-delà de  $10^\circ$ , la pente diminue légèrement (la courbe s'arrondit), mais le  $C_L$  continue d'augmenter de façon monotone jusqu'à  $\alpha = 18^\circ$ .
- Absence de décrochage : Contrairement à une expérience réelle ou une simulation Navier-Stokes, nous n'observons pas de chute brutale de portance (décrochage ou stall). Cela est dû à la nature inviscide (fluide parfait) du modèle Euler : en l'absence de viscosité, la couche limite ne peut pas se décoller. Le modèle prédit donc une portance qui continue de croître artificiellement là où la physique réelle imposerait un décrochage.

## Q11e- Intégration Temporelle Implicite

Ceci est la réponse théorique pour la question 11. Nous n'avons pas implémenté la méthode. L'intégration temporelle explicite utilisée dans la partie principale du TP impose une contrainte stricte de stabilité (condition  $CFL < 1$ ). Cette contrainte limite fortement la taille du pas de temps  $\Delta t$  et augmente considérablement le nombre d'itérations nécessaires pour atteindre l'état stationnaire. Pour s'affranchir de cette limite et accélérer la convergence, une méthode implicite est nécessaire.

### Formulation Mathématique

Le principe de la méthode implicite réside dans l'évaluation des flux numériques à l'instant futur  $t^{n+1}$  (inconnu) plutôt qu'à l'instant actuel  $t^n$ . L'équation semi-discretée de conservation pour une cellule  $K$  s'écrit alors :

$$\frac{\vec{U}_K^{n+1} - \vec{U}_K^n}{\Delta t} + \frac{1}{|K|} \sum_{e \in \partial K} \vec{\Phi}(\vec{U}_K^{n+1}, \vec{U}_{K_e}^{n+1}, \vec{n}_e) \cdot L_e = 0 \quad (19)$$

Ce système est non-linéaire car le flux numérique  $\vec{\Phi}$  dépend de l'inconnue  $\vec{U}^{n+1}$ . On ne peut donc pas isoler directement  $\vec{U}^{n+1}$ . Pour résoudre ce problème, nous utilisons une linéarisation de type Newton-Raphson.

### Linéarisation du Flux

Nous posons  $\Delta\vec{U} = \vec{U}^{n+1} - \vec{U}^n$  la variation de la solution entre deux pas de temps (la correction à calculer). En supposant que cette variation est petite, nous effectuons un développement de Taylor du flux au premier ordre autour de l'état connu  $t^n$  :

$$\vec{\Phi}^{n+1} \approx \vec{\Phi}(\vec{U}_K^n, \vec{U}_{K_e}^n) + \frac{\partial \vec{\Phi}}{\partial \vec{U}_K} \Delta \vec{U}_K + \frac{\partial \vec{\Phi}}{\partial \vec{U}_{K_e}} \Delta \vec{U}_{K_e} \quad (20)$$

Dans cette expression, les termes  $\frac{\partial \vec{\Phi}}{\partial \vec{U}}$  sont les matrices Jacobiennes ( $3 \times 3$ ) du flux numérique par rapport à l'état de la cellule gauche ( $K$ ) et de la cellule droite voisine ( $K_e$ ).

### Système Linéaire Résultant

En injectant cette linéarisation dans l'équation de bilan et en regroupant les termes inconnus ( $\Delta\vec{U}$ ) dans le membre de gauche, nous obtenons un grand système linéaire de la forme  $\mathbf{AX} = \mathbf{B}$  à résoudre à chaque pas de temps :

$$\underbrace{\left[ \frac{|K|}{\Delta t} \mathbf{I} + \sum_{e \in \partial K} \frac{\partial \vec{\Phi}}{\partial \vec{U}_K} L_e \right]}_{\text{Bloc Diagonal (Implicite)}} \Delta \vec{U}_K + \sum_{e \in \partial K} \underbrace{\left[ \frac{\partial \vec{\Phi}}{\partial \vec{U}_{K_e}} L_e \right]}_{\text{Bloc Extra-diagonal}} \Delta \vec{U}_{K_e} = - \underbrace{\sum_{e \in \partial K} \vec{\Phi}(\vec{U}_K^n, \vec{U}_{K_e}^n) \cdot L_e}_{\text{RHS : Résidu Explicite}} \quad (21)$$

### Algorithme de Résolution

L'implémentation de ce schéma implicite modifie la boucle de temps principale :

1. **Construction du membre de droite (RHS)** : Calcul du résidu explicite (somme des flux de Roe). C'est exactement le même calcul que dans le code explicite actuel.
2. **Assemblage de la matrice (LHS)** : Construction d'une matrice creuse par blocs contenant les Jacobiens des flux.
3. **Résolution** : Inversion du système linéaire (généralement via un solveur itératif type GMRES pour les grandes matrices creuses) pour trouver le vecteur correction  $\Delta\vec{U}$ .
4. **Mise à jour** :  $\vec{U}^{n+1} = \vec{U}^n + \Delta\vec{U}$ .

Cette approche permet théoriquement d'utiliser des nombres de CFL très élevés ( $CFL \gg 1$ ), accélérant considérablement la convergence vers l'état stationnaire malgré le coût de calcul supplémentaire par itération dû à l'inversion de la matrice.