

Projet 5

*Méthodes d'interpolation et d'intégration/
Spline cubique*

Group 2 - Team 2

Manager : mthioliere

Secretary : lguerin005

Programmers : felosrouti, tpitault, alamhamdi001

Résumé : Le but de ce projet est de présenter différentes méthodes d'interpolation et d'intégration et de les implémenter afin de résoudre des problèmes physiques et industriels.

Introduction

Les méthodes d'interpolations et d'intégrations sont diverses et permettent d'obtenir des approximations de calculs complexes. Tout d'abord, nous présenterons la méthode du spline cubique qui permet, à partir d'une fonction réduite à un ensemble de points, d'interpoler des points intermédiaires de cette fonction. Ensuite nous détaillerons différentes méthodes d'intégrations dans la partie suivante pour calculer la longueur d'une courbe à partir d'un ensemble de points. Pour finir nous appliquerons ces différentes méthodes à un problème industriel qui permet de déterminer la portance d'une aile d'avion.

1 Méthode d'interpolation

1.1 Algorithme d'interpolation

Le but de cette partie est d'implémenter en Python un algorithme permettant de calculer une interpolation d'un tableau de points donné en entrée, en une fonction réelle, polynomiale de degré 3 par morceaux et de classe \mathcal{C}^2 . La méthode utilisée nous est fournie par le chapitre de *Numerical Recipes in C*.

Nous avons écrit une première fonction `spline` prenant en entrée un tableau $\mathbf{X} = (x_1, \dots, x_n)$ de valeurs d'abscisses, et un tableau $\mathbf{Y} = (y_1, \dots, y_n)$ de valeurs d'ordonnées, représentant les points à interpoler. Elle retourne un tableau contenant les valeurs (y_1'', \dots, y_n'') de la dérivée seconde de la fonction d'interpolation en les abscisses des points à interpoler.

En effet, ces valeurs de dérivée secondes existent par l'hypothèse de continuité de la dérivée seconde de la fonction d'interpolation. De plus, la continuité de la dérivée

première nous fournit un système linéaire vérifié par les y_k'' . Ce système est de solution unique si l'on suppose de plus que la dérivée seconde est nulle aux extrémités de l'intervalle étudié. C'est l'hypothèse que nous avons choisie.

Finalement, ce système est assez simple pour que l'on puisse se passer de méthodes générales comme le pivot de Gauss afin de calculer ces valeurs plus rapidement : l'algorithme fourni par *Numerical Recipes in C* n'utilise que deux boucles successives sur les valeurs des points à interpoler, et est donc de complexité temporelle linéaire en `len(X)`.

Ces valeurs calculées pour les y_k'' nous seront utiles pour le calcul de l'image par notre fonction d'interpolation en un point x quelconque. *Numerical Recipes in C* nous donne en effet, avec une méthode semblable à l'interpolation de Lagrange, une formule pour calculer la valeur en une abscisse $x \in [x_k, x_{k+1}]$ de la fonction d'interpolation, à partir des valeurs de y_k , y_{k+1} , y_k'' et y_{k+1}'' . Nous avons donc écrit une fonction `splint` calculant cette valeur.

La fonction `splint` prend en entrée les tableaux `X` et `Y` représentant les points à interpoler, et une abscisse `x`. Elle renvoie la valeur de $f(x)$, où f est la fonction d'interpolation calculée. Elle commence par appeler la fonction `spline` pour récupérer les valeurs des y_k'' , puis elle calcule, par dichotomie, la valeur de k telle que $x \in [x_k, x_{k+1}]$, et enfin elle applique la formule mentionnée plus tôt.

Au total, cet algorithme est de complexité temporelle $O(\text{len}(X))$. Notons de plus que le langage Python nous procure des avantages par rapport au langage C (comme par exemple la capacité de renvoyer des tableaux). Nous avons donc profité du langage Python pour simplifier les prototypes des fonctions `spline` et `splint`, par rapport à ceux fournis par *Numerical Recipes in C*.

Finalement, comme le calcul de $f'(x)$ (en conservant les notations précédentes) sera, comme nous le verrons, nécessaire pour mener le calcul de la longueur d'un graphe, il nous faut une fonction pour calculer cette quantité. *Numerical Recipes in C* nous fournit une formule afin de calculer $f'(x)$ à partir des mêmes quantités. Il nous suffit donc de modifier l'instruction `return` finale de `splint` afin de l'adapter en une fonction `splint_der` qui, à partir des mêmes paramètres, renvoie la dérivée de la fonction d'interpolation à l'abscisse `x`.

1.2 Méthode de test

Afin de tester la fonction `splint`, nous l'avons appliquée à l'interpolation de plusieurs fonction usuelles sur l'intervalle $[0.1, \frac{\pi}{2} - 0.1]$. En effet, pour chacune de ces fonctions f , nous avons construit, grâce à `splint`, une interpolation \tilde{f} à partir de n valeurs de f . Nous avons calculé \tilde{f} sur un nombre représentatif de points (ici 1000), et nous avons étudié l'écart relatif en norme infinie entre f et \tilde{f} au fur et à mesure que n grandit. Par exemple, l'écart relatif en norme infinie entre deux fonctions f et \tilde{f} s'écrit $\frac{\|f - \tilde{f}\|_\infty}{\|f\|_\infty}$. La figure 1 montre bien que plus on considère de points d'interpolation, plus l'interpolation est précise, si bien que l'écart relatif semble converger vers 0.

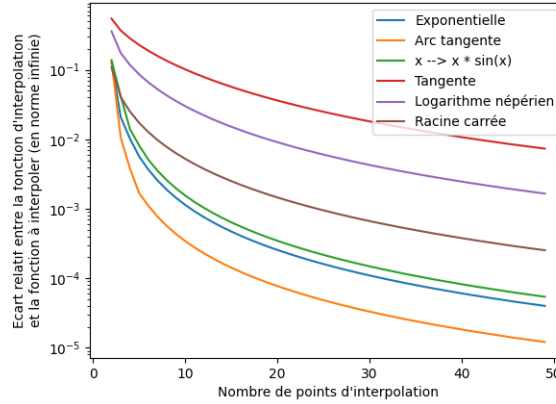


FIGURE 1 – Écarts relatifs d'interpolation de différentes fonctions usuelles en fonction du nombre de points d'interpolation

Nous avons procédé de même pour comparer les dérivées de f et de \tilde{f} . Comme cela est montré en figure 2, les écarts relatifs pour les dérivées semblent tout aussi bien converger vers 0, et il est alors justifiable d'utiliser notre interpolation pour remplacer f' par \tilde{f}' dans les calculs des parties suivantes.

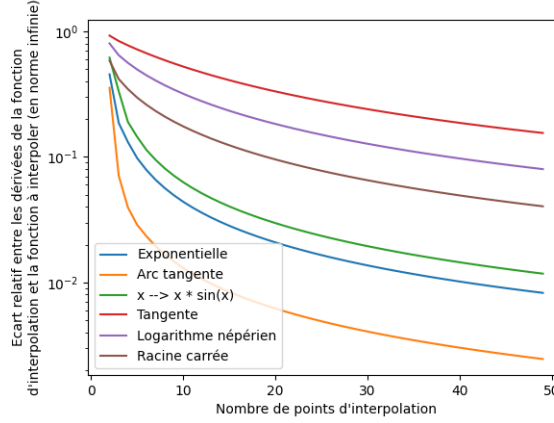


FIGURE 2 – Écarts relatifs d’interpolation de différentes fonctions usuelles, en fonction du nombre de points d’interpolation (comparaison des dérivées)

2 Méthodes d’intégration

Dans cette partie nous allons détailler trois méthodes d’intégration que nous avons implémentées. Pour commencer nous avons codé la méthode des rectangles à droite, puis la méthode des trapèzes et pour finir la méthode de *Simpson*.

2.1 Méthode des rectangles à droite

La méthode des rectangles à droite consiste à prendre un ensemble d’abscisses α_i comprises dans l’ensemble $[a, b]$ sur lequel on souhaite intégrer. Ensuite l’intégrale I_{RD} est calculée à partir de l’aire des rectangles construits à partir des points $(\alpha_i, 0)$, $(\alpha_i, f(\alpha_i))$, $(\alpha_{i+1}, 0)$ et $(\alpha_{i+1}, f(\alpha_i))$. Ainsi cette méthode donne une approximation de l’intégrale à partir d’un ensemble de fonctions constantes donc c’est une méthode d’ordre 0. L’équation 1 donne l’expression de l’intégrale calculée à partir de la méthode des rectangles.

$$I_{RD} = \int_a^b f(x)dx \approx \sum_{i=0}^{n-1} (\alpha_{i+1} - \alpha_i) \times f(\alpha_i) \quad (1)$$

La figure 3 illustre la méthode des rectangles à droites appliquée à la fonction $f : x \mapsto x^2 + x^3 + 1000$.

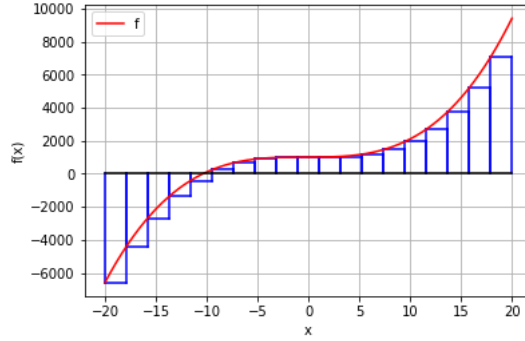
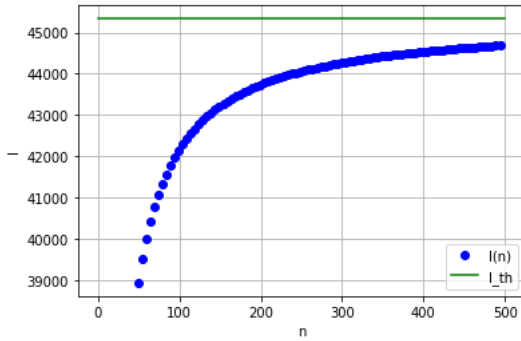
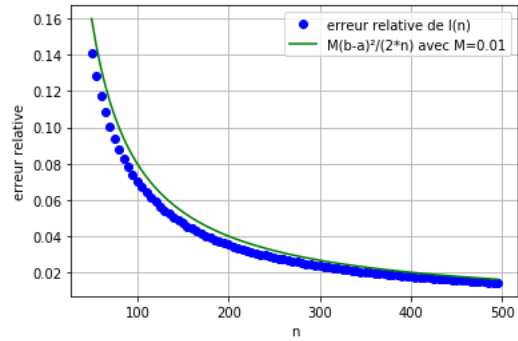


FIGURE 3 – Intégration avec la méthode des rectangles à droite

Cette méthode permet de calculer l'intégrale d'une fonction avec une erreur relative ϵ_n telle que $|\epsilon_n| \leq \frac{M(b-a)^2}{2n}$ avec M une constante et n le nombre de subdivisions pour un pas régulier h . La figure 4a montre la valeur de l'intégrale pour la fonction f en fonction de n et la figure 4b illustre l'erreur relative en fonction de n . On peut remarquer que la convergence est bien en $O\left(\frac{1}{n}\right)$.



(a) Valeur de l'intégrale en fonction de n



(b) Erreur relative en fonction de n

FIGURE 4 – Efficacité de la méthode des rectangle à droite en fonction de n

2.2 Méthode des trapèzes

La méthode des trapèzes consiste à prendre un ensemble d'abscisses α_i comprises dans l'ensemble $[a, b]$ sur lequel on souhaite intégrer. Ensuite l'intégrale I_T est calculée à partir de l'aire des trapèzes construits à partir des points $(\alpha_{i-1}, 0)$, $(\alpha_{i-1}, f(\alpha_{i-1}))$, $(\alpha_i, 0)$ et $(\alpha_i, f(\alpha_i))$. Ainsi, cette méthode donne une approximation de l'intégrale à

partir d'un ensemble de fonctions affines, donc c'est une méthode d'ordre 1. L'équation 2 donne l'expression de l'intégrale calculée à partir de la méthode des trapèzes.

$$I_T = \int_a^b f(x)dx \approx \sum_{i=1}^n (\alpha_i - \alpha_{i-1}) \times \frac{f(\alpha_{i-1}) + f(\alpha_i)}{2} \quad (2)$$

La figure 5 illustre la méthode des trapèzes appliquée à la fonction $f : x \mapsto x^2 + x^3 + 1000$.

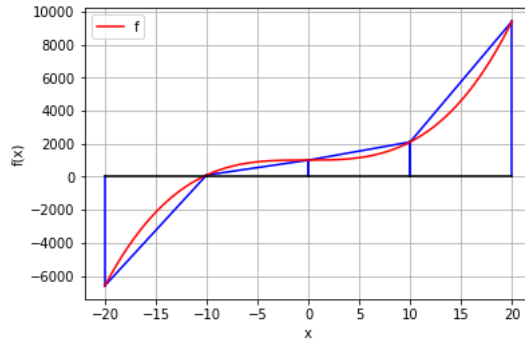


FIGURE 5 – Intégration avec la méthode des trapèzes

Cette méthode permet de calculer l'intégrale d'une fonction avec une erreur relative ϵ_n telle que $|\epsilon_n| \leq \frac{M(b-a)^3}{12n^2}$ avec M une constante et n le nombre de subdivisions pour un pas régulier h . La figure 6a montre la valeur de l'intégrale pour la fonction f en fonction de n et la figure 6b illustre l'erreur relative en fonction de n . On peut remarquer que la convergence est bien en $O\left(\frac{1}{n^2}\right)$.

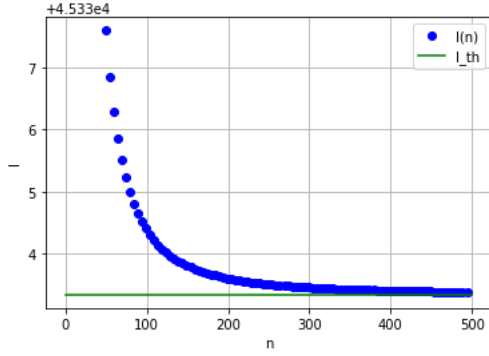
2.3 Méthode de *Simpson*

La méthode de *Simpson* appliquée sur l'intervalle $[a, b]$ consiste à le diviser en un nombre n fini d'intervalles $[\alpha_{i-1}, \alpha_i]$ tel que le pas est $h = \frac{b-a}{n}$. En posant $m_i = \frac{\alpha_{i-1} + \alpha_i}{2}$, on obtient l'intégrale suivante sur l'intervalle $[\alpha_{i-1}, \alpha_i]$:

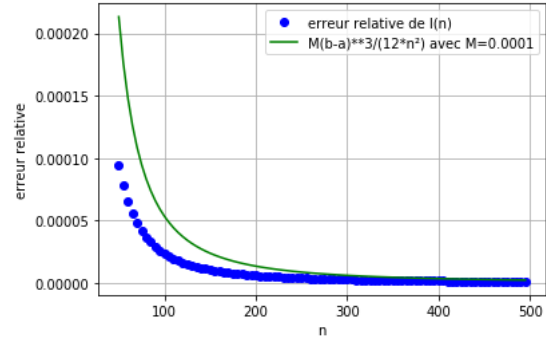
$$I_s = h \left(\frac{1}{6} f(\alpha_{i-1}) + \frac{2}{3} f(m_i) + \frac{1}{6} f(\alpha_i) \right) \quad (3)$$

Par la suite, on obtient sur l'intervalle $[a, b]$ par sommation :

$$\int_a^b f(t)dt \approx h \times \left(\frac{f(a) + f(b)}{6} + \frac{1}{3} \sum_{i=1}^{n-1} f(a + ih) + \frac{2}{3} \sum_{i=0}^{n-1} f(a + ih + \frac{h}{2}) \right) \quad (4)$$



(a) Valeur de l'intégrale en fonction de n



(b) Erreur relative en fonction de n

FIGURE 6 – Efficacité de la méthode des trapèzes en fonction de n

La figure 7 illustre la méthode de *Simpson* appliquée à la fonction $f : x \mapsto x^2 + x^3 + 1000$.

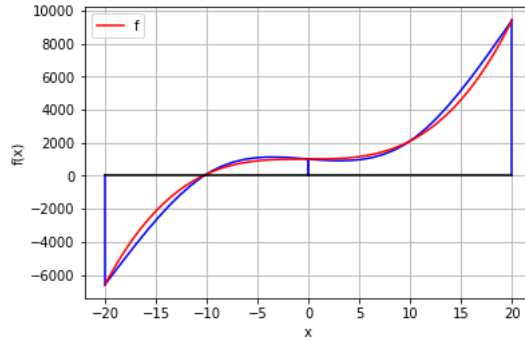


FIGURE 7 – Intégration avec la méthode de *Simpson*

Cette méthode permet de calculer l'intégrale d'une fonction avec une erreur relative ϵ_n telle que $|\epsilon_n| \leq K_s h^4 \|f^{(4)}\|_\infty$ où h est le pas régulier et K_s est une constante. La figure 8a montre la valeur de l'intégrale pour la fonction f en fonction de n le nombre de subdivisions et la figure 8b illustre l'erreur relative en fonction de n . On peut remarquer que la convergence est bien en $O(\frac{1}{n^4})$ donc en $O(h^4)$. On remarque que, en comparaison avec les deux autres méthodes, la méthode de *Simpson* donne la meilleure approximation pour un même nombre de subdivisions.

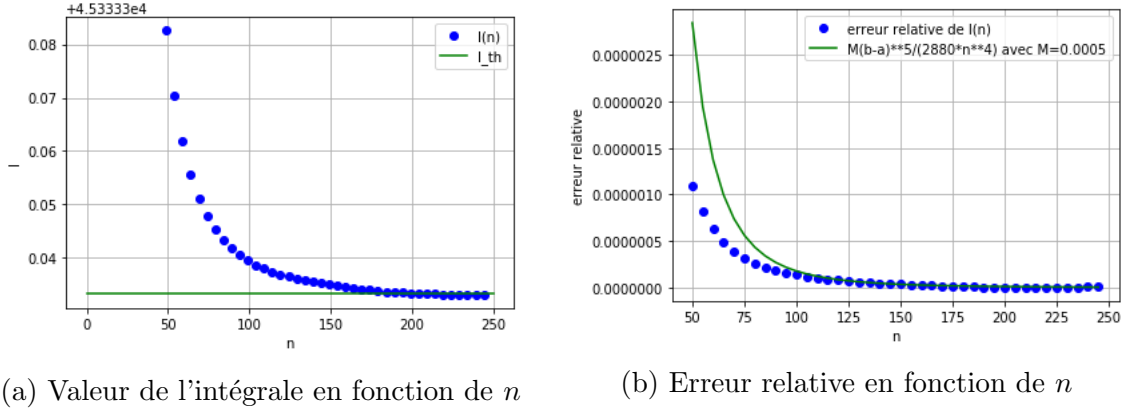


FIGURE 8 – Efficacité de la méthode de *Simpson* en fonction de n

2.4 Calcul de la longueur d'une courbe représentative

Le calcul de la longueur d'une courbe représentative est nécessaire pour la partie qui suit. Afin de déterminer cette longueur, nous utilisons l'équation 5, pour f une fonction définie et dérivable sur l'intervalle $[0, T]$.

$$L([0, T]) = \int_0^T \sqrt{1 + f'(x)^2} dx \quad (5)$$

Pour implémenter l'algorithme qui calcule cette longueur, nous avons utilisé la fonction `splint_der` vue dans la partie 1, afin d'extraire une bonne approximation de la dérivée à partir d'un nombre fini de points de f . Ensuite, nous utilisons l'une des méthodes d'intégration mentionnées précédemment pour calculer la longueur $L([0, T])$.

3 Application au problème de la portance

Dans cette partie on s'intéresse à appliquer les méthodes présentées plus haut à un problème industriel. Le problème choisi est le calcul de la pression dynamique pour un profil d'aile d'avion donné afin de déterminer sa portance. Nous avons utilisé une base de données représentant les profils d'ailes par un ensemble de points. Nous avons utilisé la méthode du spline cubique pour interpoler ces derniers afin de déterminer et dessiner le profil de l'aile. La figure 9 présente le profil du modèle **NACA M25 AIRFOIL**.

Ensuite pour simuler la portance de l'aile nous devons déterminer l'écoulement de l'air et les lignes d'écoulement. Pour cela nous supposons que l'écoulement est laminaire.

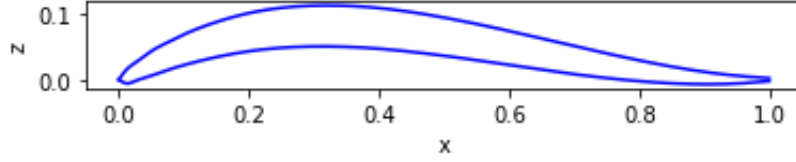


FIGURE 9 – Profil du modèle NACA M25 AIRFOIL

Pour déterminer les lignes d'écoulement nous utilisons l'équation 6 et l'équation 7 afin d'obtenir les points appartenant aux lignes d'écoulement $f_{\lambda_{sup}}$ et $f_{\lambda_{inf}}$, correspondant respectivement à l'extrados et l'intrados de l'aile. Pour restreindre le calcul du flux d'air autour de l'aile nous choisissons de nous placer dans un intervalle $[3h_{min}; 3h_{max}]$, correspondant aux ordonnées maximale et minimale de l'aile. En dehors de cet intervalle l'écoulement de l'air est supposé rectiligne.

$$f_{\lambda_{sup}} = (1 - \lambda_{sup})f_{sup}(x) + \lambda_{sup} \times 3h_{max}, \forall \lambda_{sup} \in [0, 1] \quad (6)$$

$$f_{\lambda_{inf}} = (1 - \lambda_{inf})f_{inf}(x) + \lambda_{inf} \times 3h_{min}, \forall \lambda_{inf} \in [0, 1] \quad (7)$$

La figure 10 illustre le flux d'air laminaire pour le profil NACA M25 AIRFOIL. On peut constater que plus les lignes d'écoulement sont éloignées de l'aile plus leur allure est rectiligne ce qui confirme notre intervalle de restriction.

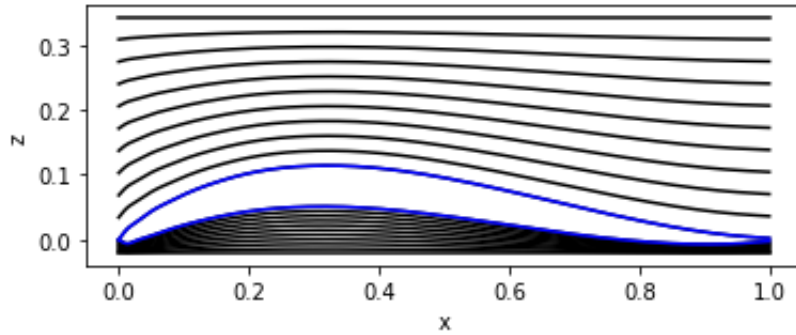


FIGURE 10 – Écoulement de l'air pour le profil NACA M25 AIRFOIL

La force de portance dépend de la différence de pression statique de l'air entre la partie inférieure et supérieure de l'aile. Cette force est dirigée de la pression statique la plus élevée vers la moins élevée. Son intensité est d'autant plus forte que la différence

entre les deux pressions est grande. Pour déterminer cette pression statique nous supposons la pression de l'air totale comme étant constante. La pression totale peut être calculée à l'aide de la loi de *Bernoulli* présentée par l'équation 8.

$$P_{air} = P_{statique} + P_{dynamique}, \text{ où } P_{dynamique} = \frac{1}{2}\rho V^2 \quad (8)$$

où V est la vitesse locale de l'air. Comme l'écoulement est laminaire, les molécules d'air mettent le même temps pour parcourir l'entièreté de chaque ligne de champ. Ainsi la vitesse de l'air peut être déterminée en fonction de la longueur de la ligne d'écoulement et de la vitesse de l'aile en vol. La longueur des lignes de champ est calculée par une méthode d'intégration, celle de *Simpson* dans notre cas, à partir de l'interpolation des points de chaque ligne. Une fois les longueurs calculées nous pouvons calculer la pression dynamique en chaque point de l'espace et afficher la carte de pression dynamique. La figure 11 décrit la carte de pression dynamique pour le profil d'aile NACA M25 AIRFOIL avec une corde de profil aérodynamique de $3.50m$, une vitesse de croisière de $900km.h^{-1}$ et pour 50 lignes d'écoulement.

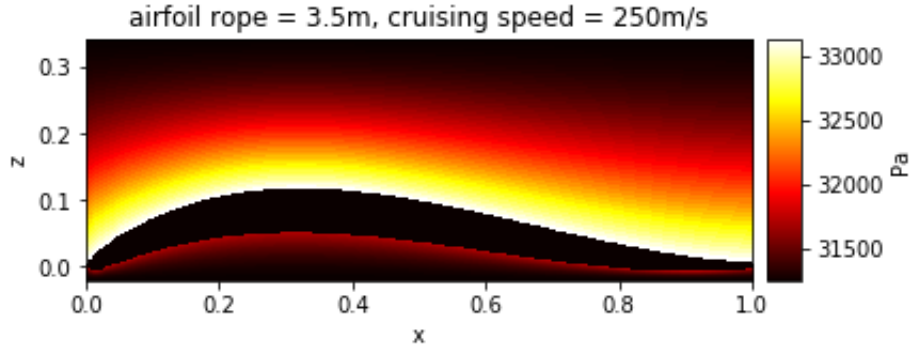


FIGURE 11 – Carte de pression dynamique pour le profil NACA M25 AIRFOIL.

On peut observer que la pression dynamique de l'air est moins élevée pour l'intrados de l'aile que pour l'extrados de l'aile donc la pression statique de l'air est plus élevée pour l'intrados que pour l'extrados de l'aile pour que la pression globale reste constante. Ainsi la force de portance est dirigée vers le haut, ce qui est l'effet recherché. Un défaut de cette simulation est qu'elle fait intervenir des approximations sur l'écoulement de l'air et la pression. La figure 12 donne le nombre de Reynolds pour l'air pour une vitesse de croisière de $900km.h^{-1}$ pour différentes longueurs caractéristiques, qui dans notre cas est la distance entre la ligne d'écoulement et l'aile. On remarque que l'écoulement

de l'air est laminaire seulement pour des longueurs caractéristiques inférieures à $10^{-4}m$. Ainsi notre hypothèse n'est valable que pour un flux très proche de l'aile.

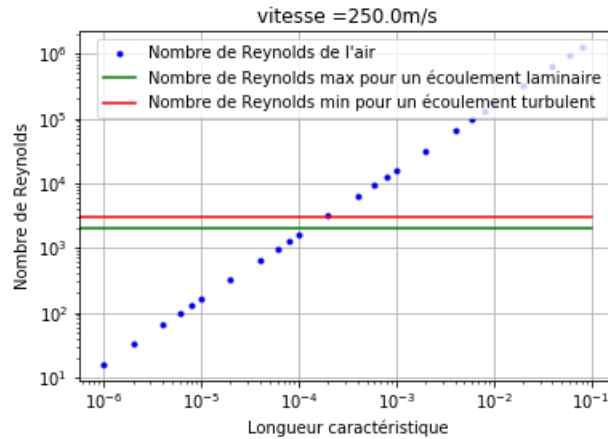


FIGURE 12 – Nombre de Reynolds de l'air pour une vitesse de croisière d'un avion de ligne pour différentes longueurs caractéristiques

4 Conclusion

Pour conclure les méthodes numériques d'interpolation et d'intégration ont des enjeux scientifiques et industriels importants. Elles permettent d'avoir une bonne approximation de la réalité. Ces méthodes appliquées à des problèmes industriels, comme la portance d'une aile d'avion, peuvent offrir des gains de temps et une réduction de coût non négligeables.

Bibliographie

- Mathieu Faverge, Projet 5, http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/?page_id=304, 23/04/2021
- University of Illinois, UIUC Airfoil Data Site, https://m-selig.ae.illinois.edu/ads/coord_database, 23/04/2021
- Numerical Recipes in C, https://www.cec.uchile.cl/cinetica/pcordero/MC_libros/NumericalRecipes, 1992