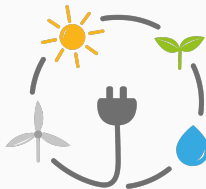


Optimisation aérodynamique avec IA d'un avion électrique

Aly Gamaleddyn*, Frédéric Alizard, Marc Buffat

Juillet 2024

Université Claude Bernard Lyon 1, LMFA, **Master M1 mécanique*



Avion électrique : concept

Avion classique

à propulsion électrique



- Moteur électrique certifié (SAFRAN 2025)
- Problème de puissance

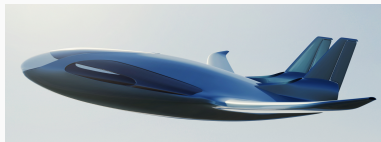
Avion électrique : concept

Avion classique à propulsion électrique



- Moteur électrique certifié (SAFRAN 2025)
- Problème de puissance

Nouveau design aérodynamique

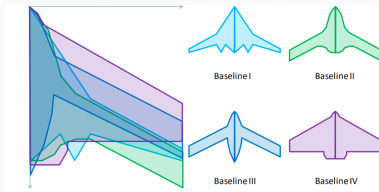


- Cellule portante
↗ finesse = $\frac{C_L}{C_D}$
- Société EENUE (gain x10)
19 passagers, 500km
(Clermont/Nantes)

Avion électrique à fuselage intégré

BWB = Blended Wing Body

- concept étudié depuis 2000
- mais pour des vols à $Ma=0.8$



objectif du stage

- étude d'une première optimisation aérodynamique dans des conditions de vol d'un avion électrique
- écoulement quasi 2D stationnaire à faible incidence

Problème d'optimisation pour un avion électrique

Optimisation de forme en aérodynamique

- problème classique dans l'aviation, l'éolien
 \rightsquigarrow ailes d'avion, pâles d'éoliennes très optimisées
- mais la solution dépend des conditions aérodynamiques
- et nécessite de très gros calculs aero (CFD)

Problème d'optimisation pour un avion électrique

Optimisation de forme en aérodynamique

- problème classique dans l'aviation, l'éolien
~> ailes d'avion, pâles d'éoliennes très optimisées
- mais la solution dépend des conditions aérodynamiques
- et nécessite de très gros calculs aero (CFD)

Contrainte aérodynamique spécifique

- faible vitesse $Ma=0.2$ (≈ 250 km/h)
- fuselage portant : blended wing body (BWB)

Utilisation d'un modèle de machine learning

- diminution du coût des calculs aero.

Problématique

→ **Optimisation de forme d'une aile pour maximiser la finesse (C_L/C_D)**

- Hypothèses : écoulement bidimensionnel, Re constant
- Solveurs NS : cher dans une optimisation
 - Machine learning pour prédiction de l'écoulement

1) Construction
base de données



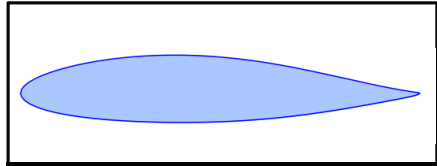
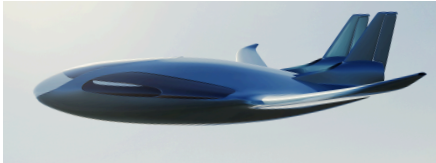
2) Entrainement
modèle IA



3) Algorithme
d'optimisation
sous contraintes



I. Paramétrisation de la base de données



- Approche de modélisation des profils : méthode CST (Class-Shape Transformation)

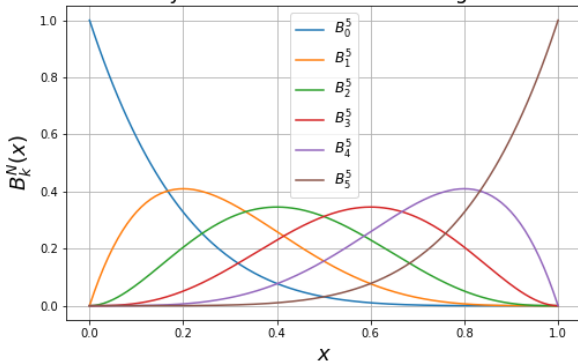
$$y(x) = C(x) \cdot S(x), \quad x \in [0, 1]$$

$$S(x) = \sum_{k=0}^N A_k \cdot B_k^N(x) \quad \text{avec} \quad B_k^N(x) = \binom{N}{k} x^k (1-x)^{N-k}$$

$$C(x) = x^{N_1} (1-x)^{N_2}$$

Polynômes de Bernstein

Polynômes de Bernstein de degré 5



$$B_k^N(x) = \binom{N}{k} x^k (1-x)^{N-k}$$

Valeurs aux bornes

- $B_0^N(0) = 1$, les autres sont nuls en $x = 0$
- $B_N^N(1) = 1$, les autres sont nuls en $x = 1$

Somme constante

$$\sum_{k=0}^N B_k^N(x) = 1 \quad \text{pour tout } x \in [0, 1]$$

Positivité

$$B_k^N(x) \geq 0 \quad \text{sur } [0, 1]$$

Symétrie

$$B_k^N(x) = B_{N-k}^N(1-x)$$

Maximum en

$$x = \frac{k}{N}$$

Expression matricielle du problème :

$$\mathbf{Y}_{\text{CST}} = \mathbf{C} \cdot \mathbf{B} \cdot \mathbf{A}$$

$$\mathbf{Y}_{\text{CST}} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} \in \mathbb{R}^M \quad \mathbf{C} = \begin{bmatrix} C(x_1) & 0 & \cdots & 0 \\ 0 & C(x_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C(x_M) \end{bmatrix} \in \mathbb{R}^{M \times M} \quad \mathbf{A} = \begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_5 \end{bmatrix} \in \mathbb{R}^6$$

$$\mathbf{B} = \begin{bmatrix} B_0^5(x_1) & B_1^5(x_1) & B_2^5(x_1) & B_3^5(x_1) & B_4^5(x_1) & B_5^5(x_1) \\ B_0^5(x_2) & B_1^5(x_2) & B_2^5(x_2) & B_3^5(x_2) & B_4^5(x_2) & B_5^5(x_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ B_0^5(x_M) & B_1^5(x_M) & B_2^5(x_M) & B_3^5(x_M) & B_4^5(x_M) & B_5^5(x_M) \end{bmatrix} \in \mathbb{R}^{M \times 6}$$

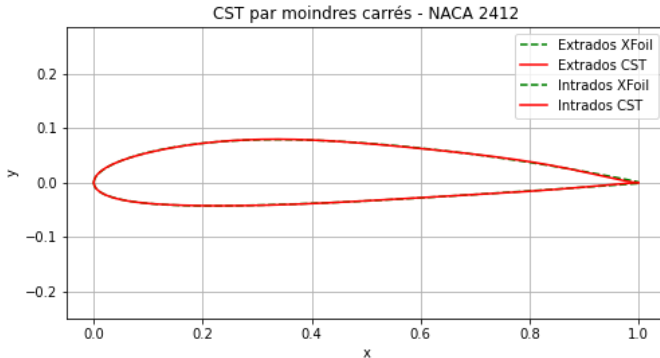
Détermination de A par méthode des moindres carrés :

- On cherche donc à minimiser l'écart : $\min_{\mathbf{A}} \|\mathbf{Y_P} - \mathbf{C} \cdot \mathbf{B} \cdot \mathbf{A}\|_2^2$
- A est solution d'un système linéaire
- Numérique (Python) : `A = np.linalg.lstsq(C @ B, Y, rcond=None)[0]`
- Coefficients → Points : $\mathbf{Y_{CST}} = \mathbf{C} \cdot \mathbf{B} \cdot \mathbf{A} \quad \mathbf{Y_P} \approx \mathbf{Y_{CST}}$

Exemple NACA 2412 : 425 points \rightarrow 12 coefficients

Coefficients CST extrados : [0.19, 0.17, 0.23 0.04, 0.19]

Coefficients CST intrados : [-0.17, -0.06, -0.12, -0.01, -0.08]



Erreur norme 2
relative extrados : 1,8%

Erreur norme 2
relative intrados : 1,9%

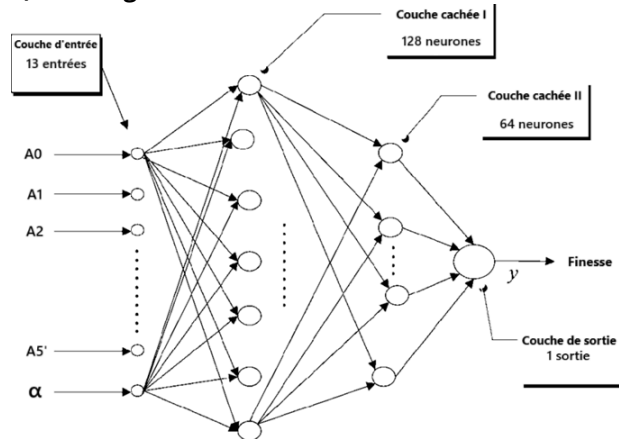
Construction de la base de données

- **1722 profils** NACA stockés
- Paramétrisation appliquée aux 1722 profils
- Calcul des coefficients aérodynamiques pour **221 angles** avec XFOil
 - **380 562 calculs aérodynamiques** { Géométrie , Angle, C_L/C_D }
- Temps de calcul : 1h30 ~ 3 min par profil ~ 0,9 s par angle
(5000 x plus rapide que calcul complet NS 2D)



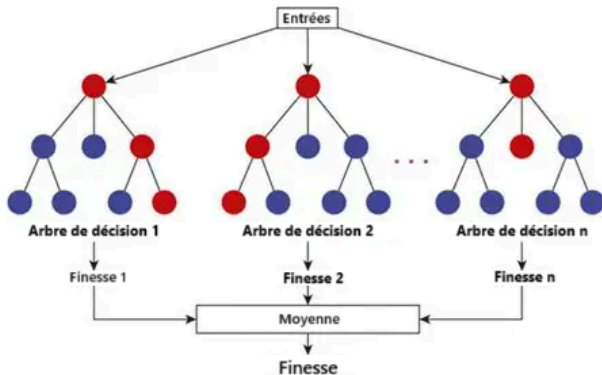
II. Modèles de machine learning

A) MLP Regressor :



- Chaque neurone calcule :
$$z = w_0 A_0 + w_1 A_1 + \dots + w'_5 A'_5 + b$$
- Fonction d'activation ReLU
$$a = \max(0, z)$$
- Rétropropagation
- Minimisation d'Adams

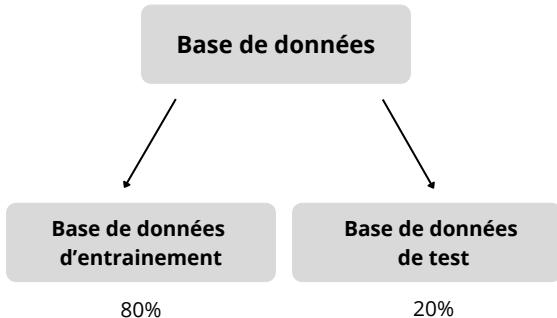
B) Random Forest Regressor :



- Choix du meilleur seuil
- Choix du meilleur paramètre
- Calcul des MSE
- Réduction de la variance

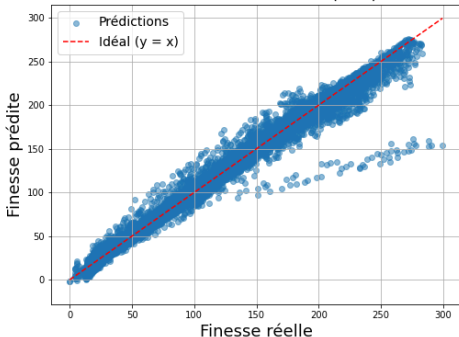
Entraînement des modèles IA

- **MPL Regressor** : 10 min
- **RF Regressor** : 5 min



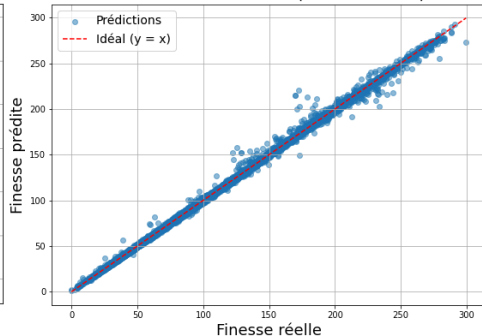
Erreur des modèles sur la base de données de test

Prédiction de la finesse (MLP)



Erreur moyenne quadratique = 23,3

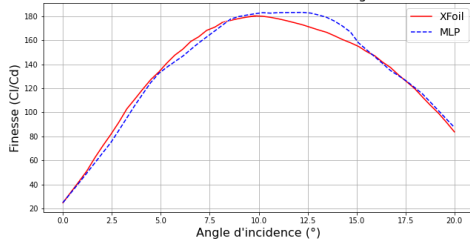
Prédiction de la finesse (Random Forest)



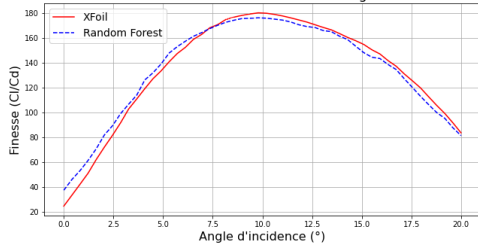
Erreur moyenne quadratique = 0,9

Exemple de prédiction de polaire

Polaire NACA 23016 : XFoil vs MLP Regressor



Polaire NACA 23016 : XFoil vs RF Regressor



III. Optimisation de forme du profil

Méthode d'optimisation nécessite de nombreux calculs aérodynamiques
(méthode de gradient avec approximation numérique)

Entrée :



$$\mathbf{V} = [A_0, \dots, A_5, A'_0, \dots, A'_5, \alpha]$$



Fonction objective :

$$J(\mathbf{v}) = -\text{Finesse}$$

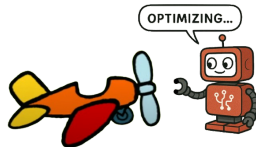
- Appel du modèle IA pour calculer la finesse
- Minimisation sous contraintes



Sortie :



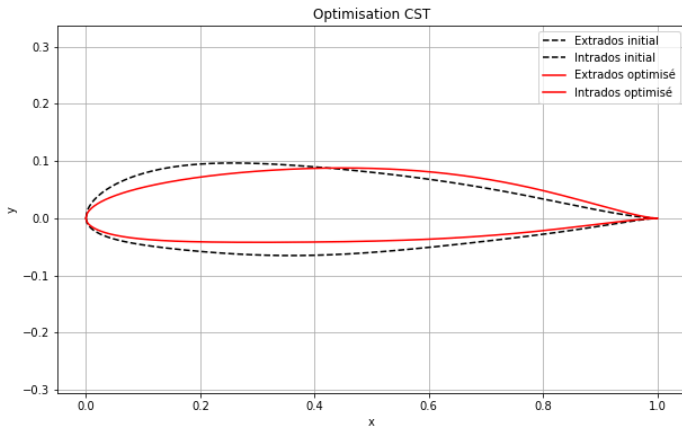
$$\mathbf{V}_{\text{OPT}}$$



Contraintes imposées sur :

- Angle d'incidence α : $\alpha_{min} \leq \alpha \leq \alpha_{max}$
- L'aire : $\int_0^1 (y_{ext}(x) - y_{int}(x))dx < \text{Aire}_{max}$
- L'épaisseur : $\max (y_{ext}(x) - y_{int}(x)) > e_{min}$
- La courbure : $\max |A_{k+2} - 2A_{k+1} + A_k| \leq C_{max}$
- Le bord d'attaque : $A_0 > S$, $A'_0 < -S$

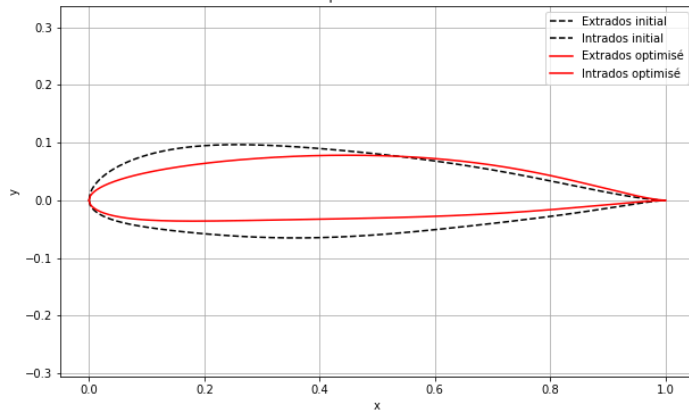
Résultats préliminaires



$$\begin{aligned}\alpha_{min} &= 0 \\ \alpha_{max} &= 15 \\ e_{min} &= 0.1 \\ Aire_{max} &= 0.1 \\ C_{max} &= 0.1 \\ S &= 0.1\end{aligned}$$

- Finesse initiale à 3° : 69
- Finesse finale : 169
- Gain : 146%
- Angle optimal : $11,7^\circ$

Optimisation CST



$$\alpha_{min} = 0^\circ$$

$$\alpha_{max} = 15^\circ$$

$$e_{min} = 0.1$$

$$Aire_{max} = 0.08$$

$$C_{max} = 0.1$$

$$S = 0.1$$

- Finesse initiale à 3° : 69
- Finesse finale : 168
- Gain : 143%
- Angle optimal : $10,9^\circ$

Bilan et perspectives

- **Bilan :**

- Modèle IA proche de XFoil
- Calculs plus rapides
- Paramétrisation efficace
- Améliorations à apporter

- **Perspectives :**

- Géométries 3D
- Paramétrisation 3D
- Choix du solveur

