

AER8270 - Aérodynamique

Guide d'utilisation pour le code HSPM



**POLYTECHNIQUE
MONTREAL**

UNIVERSITÉ
D'INGÉNÉRIE



Hess and Smith Panel Method (HSPM)

Description

Le code HSPM est un outil numérique qui résout un écoulement potentiel linéarisé en 2D, en d'autres mots, l'écoulement est non-visqueux, incompressible et irrotationnel. L'équation à résoudre se simplifie à l'équation de Laplace

$$\nabla^2 \phi = 0 \quad (1)$$

qui est linéaire. Par conséquent, le théorème de superposition est valide, ce qui permet de représenter l'écoulement par plusieurs éléments (source, vortex, doublet...). L'approche HSPM utilise des sources et des vortex pour modéliser l'épaisseur du profil et générer une circulation (portance) respectivement.



Implémentation Python

Fichiers

- *HSPM.py* : Fichier principal à exécuter qui contient les fonctions pour construire et résoudre le système linéaire.
- *geometryGenerator.py* : Contient des fonctions pour lire les points du profil contenu dans un fichier ou pour générer automatiquement un profil NACA 4-DIGIT choisi par l'utilisateur.
- *sourcePanel.py* : Module qui contient les fonctions pour calculer l'influence (vitesse induite) des sources.
- *Vector3.py* : Module qui définit un objet de type vecteur (x,y,z).

Attention

Tous les fichiers doivent être dans le même répertoire pour que le code fonctionne correctement.



Utilisation

Exemple #1

Vous pouvez exécuter directement dans un terminal le fichier *HSPM.py* **python HSPM.py**

À la fin de ce-dernier, vous pouvez modifier les paramètres de la simulation. Dans cet exemple, c'est le générateur de profils NACA qui est utilisé pour générer la géométrie.

```
1 if __name__ == '__main__':  
2     panels = geometryGenerator.GenerateNACA4digit(maxCamber=0.0,  
3                                                     positionOfMaxCamber=0.0,  
4                                                     thickness=12.0,  
5                                                     pointsPerSurface=50)  
6  
7     prob = HSPM(listOfPanels = panels,  
8                 alphaRange = [0.0,5.0,10.0,20.0,25.0],  
9                 referencePoint=[0.0,0.0,0.0])  
10  
11     prob.run()
```

execRun.py



Utilisation

Exemple #2

Vous pouvez également créer votre propre script python et importer la *class* HSPM du fichier *HSPM.py*. Dans cet exemple, c'est le générateur de profils NACA qui est utilisé pour générer la géométrie.

```
1 import HSPM
2 import geometryGenerator as geoGen
3 panels = geoGen.GenerateNACA4digit(maxCamber=2.0,
4                                     positionOfMaxCamber=4.0,
5                                     thickness=12.0,
6                                     pointsPerSurface=50)
7 prob = HSPM.HSPM(listOfPanels = panels,
8                   alphaRange = [0.0,5.0,10.0,20.0,25.0])
9 prob.run()
```

run_example_NACA2412.py



Utilisation

Exemple #3

Vous pouvez également importer une liste de coordonnées provenant d'un fichier externe pour générer la géométrie. Le fichier contient les coordonnées x et y, en deux colonnes, écrits dans l'ordre des aiguilles d'une montre à partir du bord de fuite. Le fichier *airfoil_points.dat* inclus avec le code vous montre un exemple.

```
1 import HSPM, geometryGenerator
3 panels = geometryGenerator.ReadPoints("airfoil_points.dat")
  prob = HSPM.HSPM(listOfPanels = panels, alphaRange =
    [0.0, 5.0, 10.0, 20.0, 25.0])
5 prob.run()
```

run_example_ImportedGeo.py



Utilisation

Exemple #4

Vous pouvez également exporter la vitesse tangentielle (U_e) sur l'intrados et l'extrados du profil, séparés par le point de stagnation, à l'aide des méthodes *getLowerVtangential* et *getUpperVtangential* qui ressortent les coordonnées et les vitesses sous la forme d'une liste de deux vecteurs. Cela vous sera utile pour les calculs de couche limite.

```
1 import HSPM
2 import geometryGenerator as geoGen
3 panels = geoGen.GenerateNACA4digit(maxCamber=0.0,
4                                     positionOfMaxCamber=0.0,
5                                     thickness=12.0,
6                                     pointsPerSurface=50)
7 prob = HSPM.HSPM(listOfPanels = panels, alphaRange = [2.5])
8 prob.run()
9
10 (lowerCoords, lowerV) = prob.getLowerVtangential()
11 (upperCoords, upperV) = prob.getUpperVtangential()
```

run_example_extractVTangential.py



Utilisation

Exemple #5

Vous pouvez consulter les ΔC_P associés aux angles d'attaques en entrée. Avec suffisamment d'angles en entrée, le C_{lmax} se trouve avec la fonction *findAlphaMaxCIMax* qui interpole la liste de *deltaCPvalarezo* selon un critère *valarezoCriterion*

```

1 import HSPM
2 import geometryGenerator as GeoGen
3 panels = GeoGen.GenerateNACA4digit(maxCamber=0.0,
4                                     positionOfMaxCamber=0.0,
5                                     thickness=12.0,
6                                     pointsPerSurface=120)
7 prob = HSPM.HSPM(listOfPanels = panels, alphaRange = [0.0, 5.0,
8               10, 15])
9 prob.run()
10
11 print(prob.deltaCPvalarezo)
12 (alphaMax, ciMax)=prob.findAlphaMaxCIMax(valarezoCriterion=5.0)
13 print("alphaMax: %.3lf, ciMax: %.3lf" % (alphaMax, ciMax))

```

run_example_Valarezo.py



Paramètres

Class HSPM : Input

- *listOfPanels* : Liste d'objets *sourcePanel* qui représentent la géométrie. Cette liste est créée automatiquement par le module *geometryGenerator*.
- *alphaRange* : Une liste qui contient les angles d'attaque à calculer.
- **retourne** le système HSPM.

Class HSPM : Méthodes

- *run* : Exécute le code sur l'ensemble des angles d'attaques donnés en résolvant le système de panneaux créés et retourne un fichier *CPsol_...* pour chaque angle d'attaque et contenant la solution du coefficient de pression sur le profil.
- *getUpperVtangential* : **Retourne** une liste de coordonnées (liste de Vector3-XYZ) et une liste de vitesses tangentielles (U_e) de l'extrados à partir du point de stagnation jusqu'au bord de fuite.
- *getLowerVtangential* : **Retourne** une liste de coordonnées (liste de Vector3-XYZ) et une liste de vitesses tangentielles (U_e) de l'intrados à partir du point de stagnation jusqu'au bord de fuite.



Paramètres

Fonction `geometryGenerator.GenerateNACA4digit`

- *maxCamber* : Cambrure maximale.
- *positionOfMaxCamber* : Position de la cambrure maximale.
- *thickness* : Épaisseur du profil.
- *pointsPerSurface* : Nombre de points désirés sur l'intrados et l'extrados.
- **retourne** la liste d'objets *sourcePanel* à donner à l'objet *HSPM*.

Fonction `geometryGenerator.ReadPoints`

- *inputfile* : Le nom du fichier contenant les points du profil.
- **retourne** la liste d'objets *sourcePanel* à donner à l'objet *HSPM*.



Paramètres

Class HSPM : Valarezo

- *deltaCPvalarezo* : Liste des ΔC_P correspondant à chaque angle d'attaque en entrée

Class HSPM : Méthodes pour Valarezo

- *checkIfStall_valarezo* : **Retourne** un booléen (True/False) indiquant si le critère de Valarezo donné en entrée est atteint, signifiant un décrochage. Le booléen devient vrai (True) dès qu'un des angles d'attaques génère un ΔC_P suffisamment grand.
- *findAlphaMaxCImax* : **Retourne** l'angle d'attaque maximal et le coefficient de portance (C_l) associé selon le critère de Valarezo donné en entrée. Cette méthode **interpole** la liste *deltaCPvalarezo*. Il faut donc avoir plus d'un angle d'attaque α bornant le C_{lmax}

