

# AER8270 - Aérodynamique

## Guide d'utilisation pour le code Blasius



**POLYTECHNIQUE  
MONTREAL**

UNIVERSITÉ  
D'INGÉNÉRIE



## Couche Limite Plaque Plane Blasius

### Description

Le code fourni est l'implémentation numérique décrite dans *Blasius\_integration\_numerique\_V2.pdf* disponible sur le site du cours. Le programme résout l'équation de Blasius:

$$f f'' + 2f''' = 0 \quad (1)$$

décrivant le profil de vitesse dans une couche limite **laminaire** sur une **plaque** plane **sans gradient de pression**.



# Implémentation Python

## Fichiers

- *libblasius.py* : Fichier contenant la classe Blasius et les fonctions permettant de tout résoudre
- *example.py* : Fichier d'exemple appelant la classe Blasius avec certains paramètres

## 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 *libblasius.py*  
***python libblasius.py***

À la fin de ce-dernier, vous pouvez modifier les paramètres de la simulation. Dans cet exemple, le Nombre de points d'intégration et le nombre d'itérations sont définis à 100. Les autres paramètres prennent une valeur par défaut définie dans la classe.

```
1 if __name__ == '__main__':  
3     prob = Blasius(N=100,numIter=100)  
    prob.run()
```

example\_main\_blasius.py



## Utilisation

### Exemple #2

Vous pouvez également créer votre propre script python et importer la *class Blasius* du fichier *libblasius.py*. Dans cet exemple, tous les paramètres sont définis et remplaceront les paramètres par défaut de la classe.

```
1 from libblasius import *
2
3 test = Blasius( N=20,
4                 numIter=10,
5                 x=1.0,
6                 etaStart=0.0,
7                 etaEnd=5.0,
8                 VInf=5.0,
9                 tol=-16.0)
10
11 test.initLinear()
12 test.solveBlasius()
13 test.printResults()
```

example.py



## Paramètres

### Class Blasius : Input

- $N$  : Nombres de points d'intégration en  $\eta$
- $numIter$  : Nombre d'itérations maximales (si rms ne converge pas)
- $x$  : Longueur de la plaque plane
- $etaStart$  : Début du domaine en  $\eta$
- $etaEnd$  : Fin du domaine en  $\eta$
- $VInf$  : Vitesse à l'infini
- $tol$  : Tolérance rms



## Paramètres

### Class Blasius : Méthodes

- *initLinear* : Initialise le problème avec une solution linéaire ( $f'(\eta) = \eta$ )
- *solveBlasius* : Résout Blasius jusqu'à la convergence rms désiré ou le nombre d'itérations maximal
- *printResults* : Imprime les résultats ( $\eta, U, F, f, f', f''$ ) dans un fichier définit par: *outputName="resultsBlasius.dat"*

