

# PRJ-23 Projeto Preliminar de Aeronave

## Tarefa 03 - Projeto de Perfil

### Atividade em Grupo

Maj Eng **Ney** Rafael Sêcco  
ney@ita.br

28 de agosto de 2025

## Instructions

- Due date: 9 de setembro de 2025. 11:59 PM.
- Upload your solutions using Google Classroom. ALL YOUR FILES SHOULD BE UPLOADED IN A SINGLE .zip FILE. The filename must be <group name>\_PRJ23\_Lab0X.zip.
- Send the source code as well.
- You may discuss solution approaches with other groups, but solutions and codes cannot be shared.
- Late delivery penalty: 2.0 points from the total score per day.
- The way you present your results will be taken into account during the grading.

## 1 Introdução

Nessa tarefa, usaremos o um código que analisa aerofólios segundo a equação de Euler para otimizar aerofólios para o regime transônico.

## 2 Descrição das códigos fornecidos

### 2.1 Módulos

Os seguintes módulos são fornecidos:

1. **eulerblock.exe**: código em Fortran que resolve as equações de Euler em uma malha-O estruturada bidimensional ao redor de um aerofólio. Esse programa utiliza uma metodologia de volumes finitos de segunda ordem, com fluxos calculados por meio da média de Roe. A convergência é feita por meio de Runge-Kutta de ordem 5 com passo de tempo local baseado em CFL. Esse código espera que a malha em formato PLOT3D esteja presente no arquivo **airfoil.xyz**. As condições da simulação devem estar escritas no arquivo **settings.txt**. Após a simulação, o código gravará os resultados nos arquivos **wall.dat** (informações das fronteiras do domínio) e **solution.vtk** (informações de campo em formato compatível com o software Paraview ou VisIt). Esse código aplica o método adjunto para calcular as derivadas dos coeficientes aerodinâmicos com respeito aos nós da malha e condições de contorno. Tais derivadas são guardadas no arquivo **derivatives.dat** após a execução do código;
2. **gridgen2d.py**: Gerador de malhas 2D estruturadas por meio de equações hiperbólicas. Esse código recebe a geometria do perfil para gerar a malha no espaço;
3. **gridwarp.py**: Deformador de malha 2D pelo método de ponderação pelo inverso da distância. Esse código atualiza a malha do espaço em função de deformações nos nós da superfície do aerofólio;
4. **euler\_mod.py**: módulo em Python que realiza a interface com o código em Fortran. Esse módulo chama o gerador de malha para escrever o arquivo de malha (**airfoil.xyz**) e também escreve o arquivo com as condições de simulação do código Euler (**settings.txt**);
5. **airfoil\_mod.zip**: Módulo de parametrização de aerofólios por meio da formulação CST e NACA. Esse módulo Python precisa ser instalado separadamente por meio do *pip*.

## 2.2 Pacotes externos

O módulo em Python `airfoil_mod`, utilizado para parametrização de aerofólios, precisa ser instalado separadamente por meio do `pip`. Isso pode ser feito por meio das seguintes etapas:

1. Descompacte o arquivo `airfoil_mod.zip` em algum diretório de seu computador (ou clone o repositório a partir desse link: [airfoil\\_mod](#)).
2. Abra o explorador de arquivos e navegue até a pasta descompactada;
3. Copie o endereço da pasta (clique na barra de endereço e aperte `CTRL+C`);
4. Abra o Menu Iniciar e procure por “anaconda prompt” ou “anaconda powershell”;
5. No terminal, use o comando `cd` para navegar até a pasta dos códigos. Basta digitar “`cd`” e depois clicar no ícone do topo esquerdo da janela e clicar em “Editar” > “Colar”. Aperte `<ENTER>` em seguida.
6. Agora execute o comando de instalação:

```
pip install .
```

## 2.3 Scripts

As principais tarefas desse laboratório poderão ser resolvidas por meio da execução dos seguintes scripts contidos na pasta `run_cases`:

- **01\_airfoil\_plotter:** Esse script permite analisar aerofólios de forma interativa. Ao executar o código, o usuário observará uma janela que facilita a interface com o código Euler.

Na lateral esquerda da janela estão campos para selecionar o Mach da simulação, o número máximo de iterações, o CFL (número de Courant) para controle do passo no tempo e a tolerância dos resíduos desejada.

No canto inferior da janela, o usuário pode escolher “no” para que o processo iterativo seja inicializada assumindo escoamento uniforme, ou escolher “yes” para utilizar as informações de campo já guardadas no arquivo `solution.vtk` para reinicializar a solução a partir de um caso anterior (isso pode acelerar a convergência).

No centro da janela há barras interativas para ajustar os parâmetros CST do intradorso (`Au`) e do extradorso (`A1`) do aerofólio, bem como o ângulo de ataque da simulação.

Na parte inferior da janela há três botões: 1) *Export* para criar o arquivo `airfoil.dat` com as coordenadas do perfil no formato esperado pelo Xfoil, 2) *Run* para analisar o aerofólio com o código Euler e 3) *Reset* para colocar as barras ajustáveis de volta aos valores padrão.

- **02\_single\_run:** Script que mostra como executar o código Euler sem a interface gráfica. Esse script automaticamente gera o arquivo `airfoil.dat` com as coordenadas do perfil no formato esperado pelo Xfoil.

- **03\_optimize:** Script que realiza a otimização de aerofólio para uma dada condição de escoamento fornecida pelo usuário. A definição do problema de otimização é:

$$\begin{aligned} \min \quad & c_d \\ \text{w.r.t.} \quad & \text{CST coefficients}(A_l, A_u), \alpha \\ \text{s.t.} \quad & c_\ell = c_{\ell\text{ref}} \\ & (t/c) \geq (t/c)_{\text{ref}} \\ & A_{l,\text{lower}} \leq A_l \leq A_{l,\text{upper}} \\ & A_{u,\text{lower}} \leq A_u \leq A_{u,\text{upper}} \end{aligned} \tag{1}$$

Os valores de  $c_{\ell\text{ref}}$  e  $(t/c)_{\text{ref}}$  são definidos pelo usuário no início script. O otimizador utilizado é o SLSQP, com gradientes estimados por meio do método adjunto. **Esse script pode demorar horas para executar.**

Sugere-se que os scripts sejam executados por meio do prompt de comando para permitir o monitoramento das iterações do código Euler. Siga os seguintes passos:

1. Abra o explorador de arquivos e navegue até a pasta que contém os códigos fornecidos;
2. Copie o endereço da pasta (clique na barra de endereço e aperte `CTRL+C`);
3. Abra o menu iniciar e procure por “anaconda prompt” ou “anaconda powershell”;

4. No terminal, use o comando `cd` para navegar até a pasta dos códigos. Basta digitar “`cd` ” e depois clicar no ícone do topo esquerdo da janela e clicar em “Editar” ↗ “Colar”. Aperte ↵ENTER ↵ em seguida.
5. Agora execute o script chamando o Python. Por exemplo, para executar o script `single_run.py` digite:  
`python single_run.py`

O script `optimize.py` pode demorar horas até convergir. Enquanto esse script roda em um prompt de comando, é possível acompanhar o progresso da otimização abrindo um novo prompt no mesmo diretório e executando o script `plot_history.py`. Esse script também gera o arquivo `airfoil.dat` que pode ser analisado no Xfoil.

## 2.4 Interface com o Xfoil

Ao executar os scripts `airfoil_plotter.py`, `single_run.py` ou `plot_history.py`, haverá a atualização do arquivo `airfoil.dat` com as coordenadas do último aerofólio analisado já dispostas no formato esperado pelo Xfoil. Dessa forma, é possível analisar o comportamento do perfil projetado no regime subsônico levando em conta efeitos viscosos, uma vez que a formulação Euler usada na otimização é não-viscosa. No entanto, recomenda-se fazer o tratamento da distribuição dos painéis para aprimorar a qualidade e a convergências das simulações. Ao abrir o Xfoil, utilize os seguintes comandos para ajustar a geometria:

• <code>load airfoil.dat</code>	Carrega o aerofólio
• <code>ppar</code>	Menu para ajuste dos painéis
• <code>t</code>	Ajustar concentração de painéis no bordo de fuga
• <code>0.45</code>	Fator de concentração
• <code>n</code>	Ajustar número de painéis
• <code>200</code>	Número de painéis
• <code>&lt;ENTER&gt;</code>	Retornar ao menu anterior
• <code>&lt;ENTER&gt;</code>	Retornar ao menu anterior
• <code>oper</code>	Abrir menu de análise
• <code>iter</code>	Definir número de iterações viscosas
• <code>100</code>	Número máximo de iterações viscosas
• <code>v</code>	Ativas análise viscosa
• <code>20e6</code>	Coloque o número de Reynolds desejado
• <code>a</code>	Analizar ângulo de ataque
• <code>5</code>	Ângulo de ataque desejado (em graus)

Na sequência, o usuário pode executar demais operações que achar pertinente. **Lembre-se de ajustar o número de Reynolds para o valor adequado.** O valor mostrado acima trata-se apenas de um exemplo.

## 3 Atividades

1. Primeiramente precisamos determinar um ponto de projeto (altitude, Mach e peso) para otimizar o aerofólio. Uma vez que esse ponto for estabelecido, calcule o coeficiente de sustentação ( $C_L$ ) correspondente para a aeronave. Em seguida, tome a corda média aerodinâmica da asa como corda de referência ( $c_{ref}$ ) e, para fins desse exercício, assuma que o coeficiente de sustentação dessa seção é igual ao  $C_L$  da aeronave ( $c_{lref} = C_L$ ). Utilize a espessura média da aeronave da equipe para determinar o valor de espessura mínima para a otimização ( $t/c$ )<sub>ref</sub>. Utilize tais dados para preencher a Tab. 1.

Mostre no relatório o procedimento de cálculo dos itens presentes na planilha.

2. Utilize os scripts da pasta `02_single_run` para determinar qual o ângulo de ataque (variável `alpha` da linha 19) faz com que o perfil NACA 1411 atinja o  $c_l$  de projeto. Lembre de atualizar também o número de Mach.

Table 1: Dados para a seção de referência

Parâmetro	Valor	Descrição
$h$		Altitude do ponto de projeto [ft]
$M_\infty$		Mach do ponto de projeto
$W$		Peso da aeronave no ponto de projeto [kgf]
$S_{ref}$		Área de referência da aeronave [ $m^2$ ]
$\rho_\infty$		Densidade do ar no ponto de projeto [ $kg/m^3$ ]
$a_\infty$		Velocidade do som no ponto de projeto [m/s]
$c_{ref}$		Corda da seção de referência [m]
$c_{\ell ref}$		Coeficiente de sustentação da seção de referência
$Re_{ref}$		Reynolds para a seção de referência
$(t/c)_{ref}$		Espessura relativa base da seção de referência

Table 2: Dados dos aerofólios otimizados

Parâmetro	NACA 1411	$A_{min} = 0.05$	Descrição
$A_{l1}$	-0.14894		Parâmetro CST do intradorso
$A_{l2}$	-0.10330		Parâmetro CST do intradorso
$A_{l3}$	-0.10305		Parâmetro CST do intradorso
$A_{l4}$	-0.10515		Parâmetro CST do intradorso
$A_{u1}$	0.16146		Parâmetro CST do extradorso
$A_{u2}$	0.18349		Parâmetro CST do extradorso
$A_{u3}$	0.14126		Parâmetro CST do extradorso
$A_{u4}$	0.18194		Parâmetro CST do extradorso
$\alpha$			Ângulo de ataque [graus]
$c_\ell$			Coeficiente de sustentação
$c_d$			Coeficiente de arrasto
$c_m$			Coeficiente de momento
$(t/c)_{max}$			Espessura máxima
$x_{t/c,max}$			Posição da espessura máxima
$(h/c)_{max}$			Arqueamento máximo
$x_{h/c,max}$			Posição do arqueamento máximo

3. Utilize os scripts da pasta `03_optimize` para obter um aerofólio otimizado para as condições de projeto. O ponto de partida da otimização será o aerofólio NACA 1411. Utilize como batentes para parâmetros do CST os valores  $A_{l,upper} = -0.05$  e  $A_{u,lower} = 0.05$ . Preencha a Tab. 2 com os resultados da otimização.

ATENÇÃO: Cada otimização pode demorar horas, por isso guarde bem o resultado.

4. Plote o histórico de convergência de cada otimização (pode usar o script `plot_history.py`) e discuta quanto foi o aprimoramento obtido pelo otimizador. Indique também o tempo total de otimização e como as restrições afetaram o resultado.
5. Desenhe 1) gráficos sobrepondo a geometria, 2) gráficos sobrepondo curvas de  $C_p$  ao longo da corda e 3) gráficos sobrepondo curvas de Mach ao longo da corda dos aerofólios da Tab. 2 na condição de projeto transônica. O script da pasta `02_single_run` pode ser útil para isso.
6. Use os gráficos gerados para explicar a abordagem aplicada pelo otimizador para chegar na configuração ótima.
7. Adapte o script `02_single_run.py` para gerar uma polar de arrasto para o perfil otimizado na condição transônica. Para isso, você precisará simular o perfil para uma sequência de ângulos de ataque e depois plotar os valores de  $c_\ell$  e  $c_d$ . Sugiro usar um intervalo de  $-2/+2$  graus em relação ao ângulo de ataque do ponto ótimo. Lembre-se de destacar o ponto de projeto ( $c_\ell$  da otimização) na polar. Sobreponha no mesmo gráfico uma curva correspondente ao aerofólio transônico RAE2822 (os dados desse aerofólio estão no arquivo `single_run.py`). Discuta se a polar do aerofólio otimizado tem alguma peculiaridade e se isso é bom ou ruim para o projeto de uma aeronave.
8. Exporte os aerofólios iniciais e finais da otimização para o formato Xfoil e gere curvas  $c_\ell \times \alpha$ ,  $c_\ell \times c_d$  e  $c_m \times \alpha$  para os dois aerofólios em regime subsônico (pode desconsiderar o número de Mach para essa análise). Utilize a condição de decolagem da aeronave para determinar o número de Reynolds para essa simulação (lembre de indicar o número de Reynolds no relatório). Sobreponha as curvas dos dois aerofólios em cada plot e discuta as vantagens e desvantagens dos perfis no regime analisado.

9. Verifiquem a possibilidade de alterar a definição da otimização para obter um aerofólio de melhor característica no regime subsônico.

## 4 Paraview

Quem estiver com o Paraview (<https://www.paraview.org/download/>) instalado no computador pode observar a solução no domínio carregando o arquivo `paraview_state_windows.py` com a opção “Load State”. Esse arquivo sempre carregará o arquivo `solution.vtk` da pasta `02_single_run`.

## 5 Outras dicas

- Quando você carrega um perfil no Xfoil, ele dá informações sobre espessura e arqueamento, bem como sobre as posições dos máximos desses valores. Você também pode obter essas informações com a função `cstfoil` presente no arquivo `airfoil_mod.py` ou nos resultados impressos pelo script `single_run.py`.
- Caso a otimização seja interrompida por algum motivo, você pode reinicializá-la aproveitando o histórico anterior com o seguinte procedimento:
  1. Faça uma cópia do arquivo `opt_results.pickle` presente na pasta do código e renomeie essa cópia para `opt_results_copy.pickle`;
  2. Abra o arquivo `optimize.py` e, na linha 91, substitua:

```
reload_opt = None
```

por:

```
reload_opt = 'opt_results_copy.pickle';
```
  3. Execute novamente o script `optimize.py`.
- Tentem alterar o parâmetro `ftol` do arquivo `optimize.py` para uma tolerância menor (por exemplo  $1.0 \cdot 10^{-5}$ ) para obter aerofólios mais otimizados. No entanto, isso pode aumentar o tempo de otimização. Vocês também podem alterar os batentes da otimização alterando as variáveis `Al_lower`, `Al_upper`, `Au_lower`, `Au_upper`.