# Introdução

Perfis aerodinâmicos definem as formas das seções que compõe as asas e superfícies de controle de aeronaves. Alguns exemplos de perfis:

{FIGURA}

As coordenadas geométricas desses perfis estão armazenadas no site {https://m-selig.ae.illinois.edu/ads/coord\_database.html}.

Para a execução desse trabalho escolhi os seguintes perfis aerodinâmicos utilizandos 3 letras iniciais do meu nome (sendo elas: R, D e A):

*rg8.dat - Rolf Girsberger RG 8 airfoil*

*arad13.dat - Aeronautical Research Association / Bocci - Dowty Rotol ARA - D13% thick propeller airfoil*

*drgnfly.dat - Dragonfly Canard airfoil*

Cada um dos perfis contém dois arquivos; o .dat, que traz as as coordenadas dos perfis superior e inferior, e o .gif, que traz uma visualização desses dados.

A partir da escolha dos perfis aerodinâmicos, iremos então analisar esses dados de maneira a encontrar o melhor polinômio de ordem 3 que aproxima as coordenadas de cada perfil. Faremos essa tarefa através do método *cubic splines.* Ao encontrar cada um desses polinômios, iremos interpolar as splines de acordo com os pontos gerados pela distribuição x = ½ \* (1 – cos(theta)), onde theta = [0, pi], e então verificaremos se as splines geram um gráfico próximo dos dados reais.

# Desenvolvimento

Para realização deste trabalho, foi utilizada a linguagem Python e suas bibliotecas Numpy e Matplotlib, para, respectivamente, tratamento de vetores e arrays e geração de gráficos a partir de um conjunto de dados.

Considerando as letras iniciais do meu nome, foram escolhidos os seguintes perfis aerodinâmicos:

*rg8.dat - Rolf Girsberger RG 8 airfoil*

*arad13.dat - Aeronautical Research Association / Bocci - Dowty Rotol ARA - D13% thick propeller airfoil*

*drgnfly.dat - Dragonfly Canard airfoil*

A partir deles, retiramos os dados necessários para o início da nossa análise. Por exemplo, do arquivo rg8.dat foram recolhidas as coordenadas (x, y) dos perfis aerodinâmicos superiores e interiores (na aeronáutica, são chamados de extradorso e intradorso, respectivamente). No arquivo rg8.gif, temos uma visualização das coordenadas (x, y) em um gráfico cartesiano. Nas figuras as seguir podemos ver como estão organizados todos os arquivos .dat utilizados neste trabalho e suas respectivas visualizações no gráfico cartesiano.

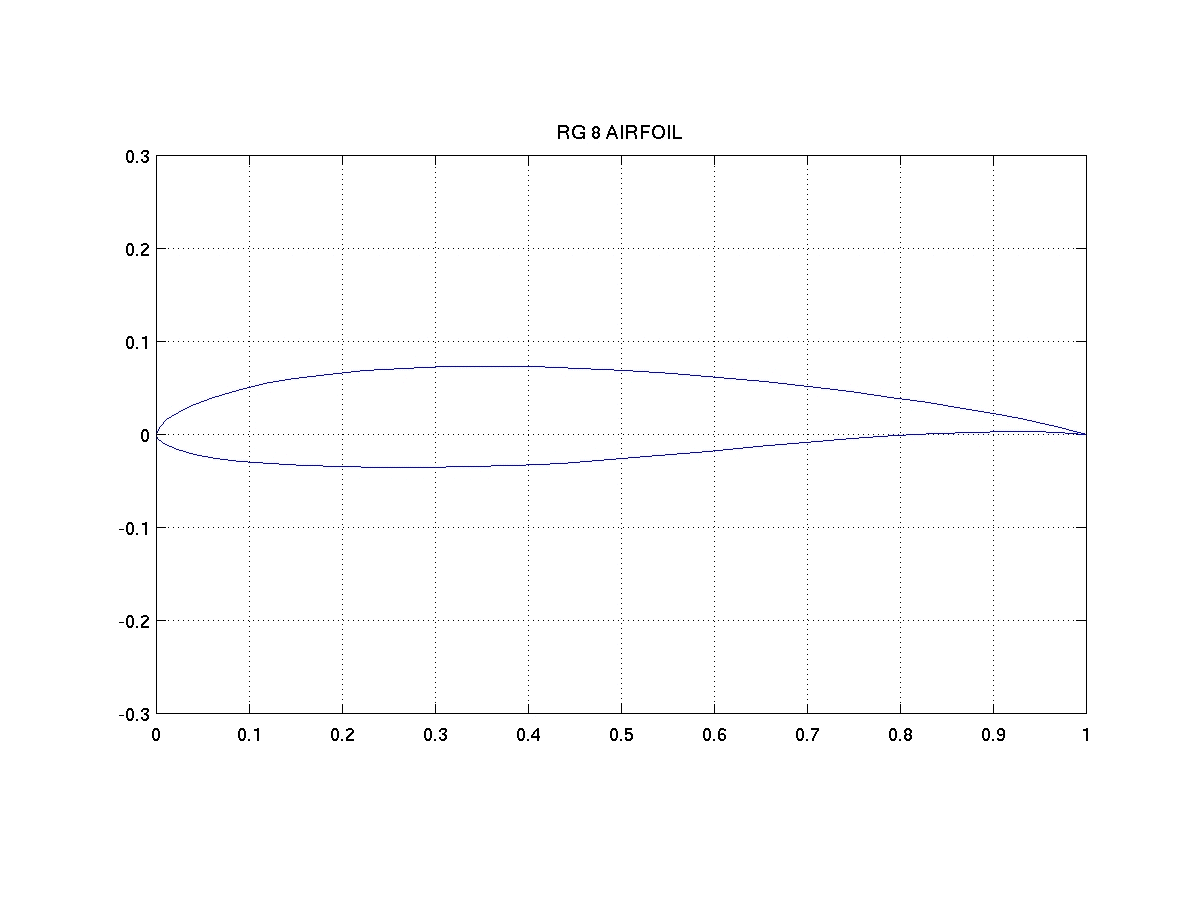
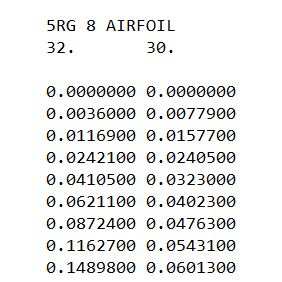


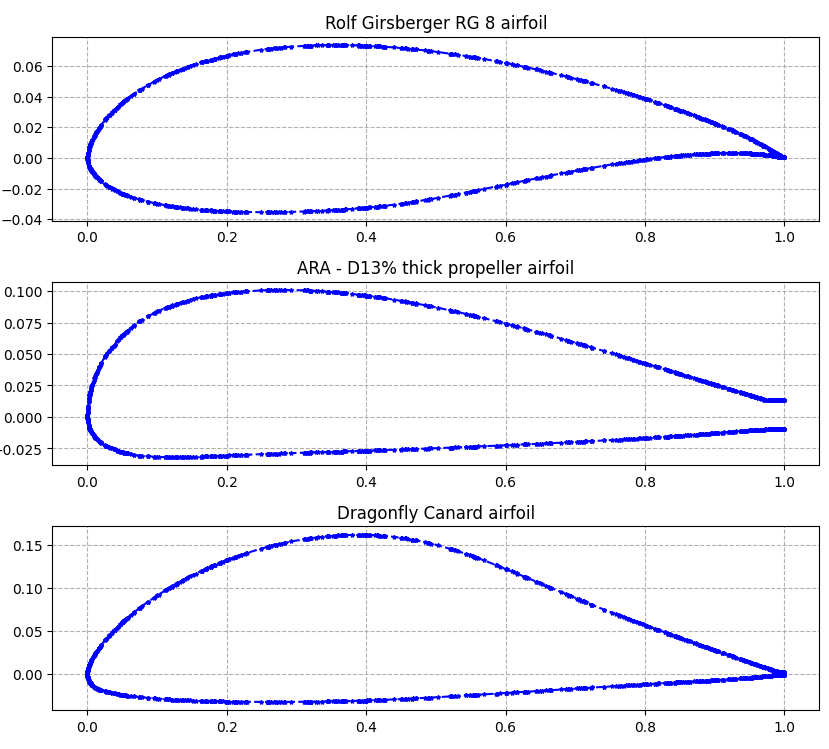
Figura 1. Arquivo rg8.dat Figura 2. Visualização dos dados

A rotina utilizada para obter esses dados dentro do programa está contida na função *read\_data*, que tem como parâmetros o nome do arquivo em questão e o modo de execução da função (neste trabalho, só utilizaremos o modo ‘r’ de leitura de arquivos).

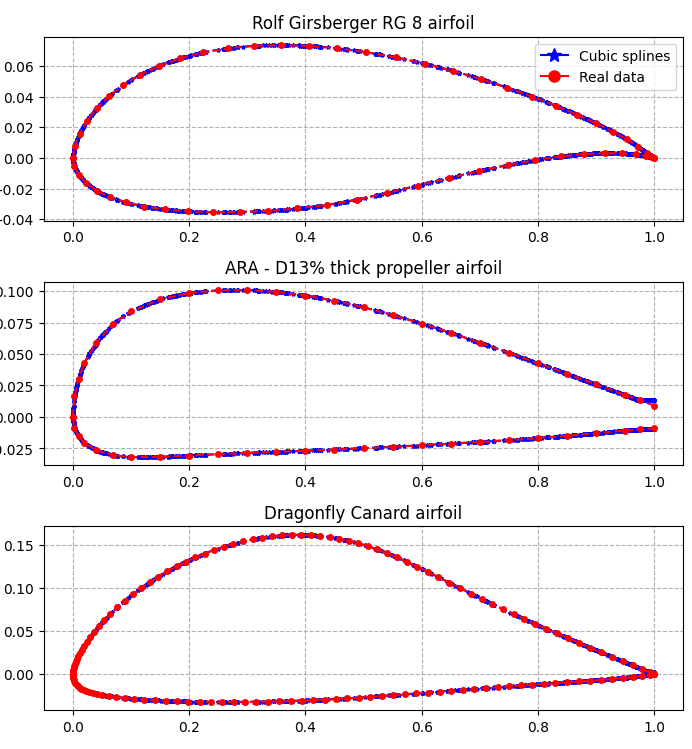
A função *read\_data* irá ler cada linha do programa, separar como colunas os dados que estão separados por um espaço branco e irá retornar em dois arrays tais dados. Para diferenciar onde terminam os dados do extradorso e onde começam os dados do intradorso foi colocada uma linha em branco. O índice dessa linha é guardado numa variável. Ao fim da função, fazemos o fatiamento dos dados de acordo com índice que separa os dados dos perfis superiores e inferiores e os guardamos em dois arrays distintos: *extradorso* e *intradorso*.

Com os dados arranjados, podemos agora iniciar o procedimento para encontrar as splines cúbicas que vão aproximar nossos dados através de polinômios de ordem 3. Para facilitar a reutilização, leitura e compreensão do código, nós quebramos o algoritmo em diversas pequenas funções: *catch\_values*, *diff\_x*, *find\_alpha*, *create\_l\_u\_z* e *coeff\_b\_c\_d*. Como o algoritmo desenvolvido neste trabalho está explicado na Seção 3.5 (Algoritmo 3.4) do Capítulo 3 – Interpolação e Aproximação Polinomial da referência deste curso, a saber, o livro Numerical Analysis, de Richard L. Burden, não iremos nos prolongar em sua explicação. Ao fim do algoritmo, teremos em mãos os coeficientes de todos os polinômios interpoladores.

Para interpolar as splines encontradas geramos 500 pontos originados de uma distribuição definida por x = 0.5 \* (1 – cos(theta)), onde theta = [0, pi], através da função *dist\_theta.* Ao interpolar esses pontos, obtivemos os seguintes gráficos:



Olhando esses gráficos de maneira isolada, eles já parecem assumir a forma dos perfis aerodinâmicos escolhidos. Agora vamos plotar os gráficos das splines e dos dados originais juntos para compararmos seus desenhos diretamente e de maneira mais clara.



Como podemos ver, as splines (em azul) obtidas através do algoritmo disponível no livro Numerical Analysis se aproximaram muito bem dos dados originais (em vermelho), demonstrando o caráter preciso do método das cubic splines para aproximação de dados.

Além disso, foi pedido que identificássemos o ponto de espessura máxima do perfil aerodinâmico e seu valor de espessura. Como ferramenta para execução dessa tarefa utilizei a distância entre dois pontos, por ser um método muito simples de implementar e que respondeu melhor ao problema apresentado. Dois pontos que têm a maior distância entre si comparada às distâncias dos demais pontos são identificados como os pontos extremos da reta que representa o “diâmetro” do perfil. Sendo assim, a coordenada x desses pontos (que deve ser a mesma para ambos os pontos) é a posição de espessura máxima do perfil aerodinâmico em questão. A distância calculada entre esses dois pontos é o próprio valor de espessura. No gráfico, os pontos de espessura máxima e suas posições ficam dessa maneira:

