# Banco de dados muti-plataforma de coeficientes aerodinâmicos de baixo Reynolds para ângulos de ataque estendidos com dependência de Mach

**Daniel Ribeiro Santiago**

**Lucas Otaviano Alves**

Universidade Federal de Uberlândia

Abril/2020

## Resumo

Processos modernos de otimização, meta-modelagem e analise de dados para os mais diversos fins se beneficiam enormemente da disponibilidade de uma grande massa da dados a ser analisada. ZHANG, Yao et al. [1] demonstrou a possibilidade da utilização de um grande volume de dados gerados por programas de modelagem física computacional para o treinamento de redes neurais convolucionais (CNN) no desenvolvimento de um metamodelo capaz de gerar os mesmos resultados dos dados em que foi baseado mas com custo computacional significativamente reduzido. Usos similares de dados em processos de meta-modelagem de modelos físicos também foram obtidos por LI, Jichao et al. [2].

Especificamente no desenvolvimento de elementos aerodinâmicos de rotores, não são incomuns regimes de operação com ângulos de ataque acima dos usuais em projetos de asas fixas, com condições que podem incluir a região de pós-estol e alcançar números de Mach consideráveis. Torna-se então de grande utilidade a disposição de dados de aerofólios para uma grande variedade de números de Mach, Reynolds e ângulos de ataque. No entanto, dados aerodinâmicos nessas condições são escassos sendo os disponíveis, como aqueles em *airfoiltools.com* [3], em formatos de difícil busca e acesso através de linguagens de programação.

Este artigo tem como objetivo a estruturação e criação de um banco de Dados de aerofólios denominado *Aerodynamic Structured Query Language DataBase* (*AeroSQLDB* ou apenas *AeroSQL*) que seja multi-plataforma, de confiabilidade rastreável, englobe diversas condições de operação e que possa ser amplamente acessado e populado pela comunidade de pesquisadores além de pré-preenchido com valores de Cl, Cd e Cm para diversos números de Reynolds e Mach de todas as geometrias de aerofólios disponíveis no banco de dados *UIUC Airfoil Coordinates Database* [4] . O Banco de dados deve contar também com a disponibilidade de dados em ângulos de ataque para além da região de estol, obtidos a partir de métodos de baixo custo computacional.

## 1 Introdução

Muitos trabalhos publicados buscam obter dados aerodinâmicos de aerofólios em baixos números de Reynolds, como o compilado de publicações de testes em tuneis de vento feitos por SELIG, M.S et al. [5–9]. No entanto, tais publicações se encontram espalhadas em vários documentos de formatos diversos e de difícil acesso.

O cenário em que alguém, desejando obter informações contidas nestes documentos, acaba por exaustivamente recorrer a busca e digitação manual de entradas se repete constantemente entre grupos de pesquisa e equipes de competição. Mais comum ainda são as situações onde pessoas em diversos locais diferentes direcionam suas capacidades de processamento a execução de programas de CFD ou que utilizam métodos potenciais como o XFOIL [10] em condições iguais ou similares. Estes entes poderiam salvar tempo de processamento ao utilizar uma abordagem de banco de dados, onde poderiam buscar os mesmos dados, sendo o custo computacional de sua geração gasto apenas uma vez.

Se torna interessante então o desenvolvimento de um banco de dados que possa abrigar as informações vindas de diversas fontes diferentes e que possa ser atualizado e acessado por pessoas da comunidade. Para garantir a confiabilidade dos dados presentes neste banco é importante que haja alguma forma de rastrear a fonte por qual foram gerados e o usuário que os postou.

Desta forma, alguém que pretende acessar os dados no banco pode, seletivamente, obter apenas os dados de uma fonte ou usuário que confia plenamente.

Neste trabalho o Sistema de Gerenciamento de Banco de Dados Relacionais (RDMS) MySQL foi escolhido como a plataforma de gerenciamento de banco de dados por ser de software livre, portátil, padronizado e de excelente desempenho.

Com o intuito de incentivar o uso da plataforma criada, o presente trabalho também se dedica ao esforço de pré popular o banco de dados com 1.546 geometrias de aerofólios presentes em [4], além de cerca de 264.601 Polares, geradas através do software XFOIL, com base nestes aerofólios para diversas condições de operação.

# 2 Metodologia

## 2.1 Estruturação do Banco de Dados


Figura 1: Diagrama EER do banco de dados AeroSQL

A primeira etapa do processo de criação do banco de dados é a sua estruturação, isso é, quais serão as tabelas que o irão compor, quais as colunas presentes nestas tabelas e como elas irão se relacionar. Além disso, é importante ainda desenvolver mecanismos de segurança na plataforma para prevenir eventuais ataques de negação de serviço, como usuários mal intencionados com permissão de escrita deletando dados, ou realizando tarefas que ocupam o servidor de forma a inviabiliza-lo.

O banco de dados foi constituído em cinco tabelas, estruturadas conforme exemplificado na Figura 1:

1. Airfoils
2. Geometries
3. Runs
4. Polars
5. PolarProperties

A tabela **Airfoils** Figura 2 contém as informações que caracterizam os aerofólios na plataforma, como nome, espessura e cambra, enquanto a tabela **Geometries** Figura 3 contém suas coordenadas X e Y correspondentes.

| AirfoilID | File_name | Name | Thickness | X_Thickness | Camber | X_Camber | Source | Creator_us |
|---|---|---|---|---|---|---|---|---|
| 1 | 2032c | 20-3... | 8.00 | 18.18 | 7.00 | 10.31 | UIUC Airfoil C... | daanrsan... |
| 2 | a18 | A18 (... | 7.35 | 29.29 | 5.04 | 8.34 | UIUC Airfoil C... | daanrsan... |
| 3 | a18sm | A18 (... | 7.28 | 26.26 | 3.85 | 6.99 | UIUC Airfoil C... | daanrsan... |
| 4 | a63a108c | NASA... | 7.74 | 31.31 | 0.55 | 3.41 | UIUC Airfoil C... | daanrsan... |
| 5 | ag03 | AG03... | 6.24 | 25.25 | 2.02 | 5.09 | UIUC Airfoil C... | daanrsan... |

Figura 2: Exemplo de resultados contidos em Airfoils

| AirfoilID | X | Y | Side |
|---|---|---|---|
| 1 | 1.000000 | 0.001600 | Top |
| 1 | 0.950000 | 0.012400 | Top |
| 1 | 0.900000 | 0.022900 | Top |
| 1 | 0.800000 | 0.042800 | Top |
| 1 | 0.700000 | 0.061000 | Top |
| 1 | 0.600000 | 0.077100 | Top |
| 1 | 0.500000 | 0.090500 | Top |

Figura 3: Exemplo de resultados contidos em Geometries

Cada conjunto de valores que determinam uma condição de operação única, como Aerofólio, número de Reynolds, número de Mach e fonte dos dados, compõe uma linha da tabela **Runs** Figura 4. Cada Run é única e identificada por sua *RunID*, valor que relaciona as informações da Run ao seus correspondentes nas tabela Polars e PolarProperties.

| RunID | AirfoilID | N_Point | Alpha_m | Alpha_m | Ncrit | Mach | Reynolds | Source | Status | RunDate | Creator_ | AdditionalData |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 202 | -18.000 | 18.500 | 9.00 | 0.000 | 20000 | Xfoil | Done | 2020-02... | daanr... | N_Panels = 200... |
| 2 | 1 | 196 | -18.000 | 18.500 | 9.00 | 0.000 | 35000 | Xfoil | Done | 2020-02... | daanr... | N_Panels = 200... |
| 3 | 1 | 198 | -18.000 | 18.500 | 9.00 | 0.000 | 50000 | Xfoil | Done | 2020-02... | daanr... | N_Panels = 200... |
| 4 | 1 | 198 | -18.000 | 18.500 | 9.00 | 0.000 | 65000 | Xfoil | Done | 2020-02... | daanr... | N_Panels = 200... |
| 5 | 1 | 168 | -18.000 | 18.500 | 9.00 | 0.000 | 80000 | Xfoil | Done | 2020-02... | daanr... | N_Panels = 200... |

Figura 4: Exemplo de resultados contidos em Runs

Os coeficientes aerodinâmicos correspondentes a cada conjunto de condições obtido em uma linha da tabela Runs são inseridos na tabela **Polars** Figura 5.Para cada polar calculada, suas propriedades como $Cl_{max}$, $Cd_{min}$, e $\alpha_{stall}$ são inseridas na tabela **PolarProperties** Figura 6

As colunas *Creator_username* são preenchidas automaticamente com o nome do usuário logado no banco de dados durante a inserção de linhas na tabela e não podem ser modificadas. Adicionalmente, linhas inseridas por um usuário não podem ser deletadas por outro. Isso cria uma camada de proteção contra eventuais abusos de usuários com privilégio de escrita, evitando que se passem por outros e que deletem conteúdo no banco.

Além disso, os valores de *N_Points, Alpha_min e Alpha_max* na tabela Runs, que correspondem respectivamente ao número de pontos, alfa mínimo e alfa máximo dos dados contidos na tabela Polars para o Run correspondente são atualizados automaticamente após uma inserção em Polars.

| RunID | AirfoilID | Alpha | Cl | Cd | Cm |
|---|---|---|---|---|---|
| 1 | 1 | -18.000 | -0.53590 | 0.23470 | 0.02500 |
| 1 | 1 | -17.900 | -0.53020 | 0.23154 | 0.02460 |
| 1 | 1 | -17.800 | -0.52560 | 0.22955 | 0.02410 |
| 1 | 1 | -17.700 | -0.52150 | 0.22808 | 0.02360 |
| 1 | 1 | -17.600 | -0.51770 | 0.22691 | 0.02290 |
| 1 | 1 | -17.500 | -0.51430 | 0.22598 | 0.02220 |
| 1 | 1 | -17.400 | -0.51120 | 0.22529 | 0.02140 |

Figura 5: Exemplo de resultados contidos em Polars

| RunID | AirfoilID | Cl_max | Cl_0 | Cl_alpha | Cd_min | Cd_max | Cl_Cd_max | Cm_0 | Alpha_stall | Alpha_0_Cl | Alpha_Cl_Cd_max | Creator_use |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0.000 | 0.139 | 0.138 | 0.050 | 0.271 | 7.673 | -0.055 | 0.000 | -1.007 | 2.750 | daanrsant... |
| 2 | 1 | 0.949 | 0.374 | 0.205 | 0.042 | 0.228 | 18.992 | -0.142 | 4.000 | -1.823 | 2.250 | daanrsant... |
| 3 | 1 | 1.217 | 0.638 | 0.130 | 0.026 | 0.231 | 32.278 | -0.146 | 6.000 | -4.900 | 2.000 | daanrsant... |
| 4 | 1 | 1.565 | 0.672 | 0.118 | 0.021 | 0.232 | 42.749 | -0.140 | 9.500 | -5.676 | 9.000 | daanrsant... |
| 5 | 1 | 1.565 | 0.697 | 0.107 | 0.018 | 0.223 | 49.098 | -0.136 | 9.500 | -6.503 | 8.750 | daanrsant... |

Figura 6: Exemplo de resultados contidos em PolarProperties

O campo *AdditionalData* na tabela Runs foi adicionado com a intenção de cobrir eventuais casos específicos de plataforma que caracterizam como a Run foi gerada, como o número de painéis e iterações em uma chamada ao XFOIL. Este campo também pode ser usado para detalhar falhas durante o processo de geração das polares correspondentes, para que fique, então, documentado potenciais problemas na constituição desta polar. É recomendado que se este campo for preenchido por com múltiplos valores, estes sejam separados por um token como ";"para que possa ser analisado facilmente por ferramentas já existentes como leitores de formato CSV.

Um usuário do AeroSQLDB com as credenciais contidas na Tabela 1 foi criado com permissão apenas de leitura para uso geral da comunidade que deseja acessalo:

| | |
|---|---|
| User: | Reader |
| Password: | |

Tabela 1: Credenciais com permissão de leitura para uso geral (sem senha)

## 2.2 Geração de Curvas pelo XFOIL

No objetivo de popular a plataforma com dados prontos para uso, foi escrito um código em Python chamado *Gera_AeroSQL_DB_mysql.py* esquematizado na Figura 7.

O *Gera_AeroSQL_DB_mysql.py* funciona consultando o banco de dados a procura de Runs com as colunas *Status* e *Source* com os valores 'ToDo' e 'Xfoil' respectivamente. Após a aquisição dos parâmetros necessários a partir da linha obtida, este atualiza o valor de *Status* para 'Doing' e executa o XFOIL.

Para preencher o *AeroSQLDB* com as *Runs* pretendidas neste trabalho, foi escrito em script em *Matlab* chamado *Gera_runs_AeroSQL_DB.m* que realiza esta função. Para isso foi necessário decidir quais as condições de operação, como Reynolds e Machs, seriam executadas por cada aerofólio no banco de dados.

Os valores de Mach e Reynolds foram escolhidos com base em [11], [12] e [13], moderando o tempo computacional para a obtenção das curvas e a resolução dos coeficientes. As regiões em que há maior variação dos coeficientes com relação aos números Mach e Reynolds receberam uma maior quantidade de pontos para análise, por meio da redução do passo.

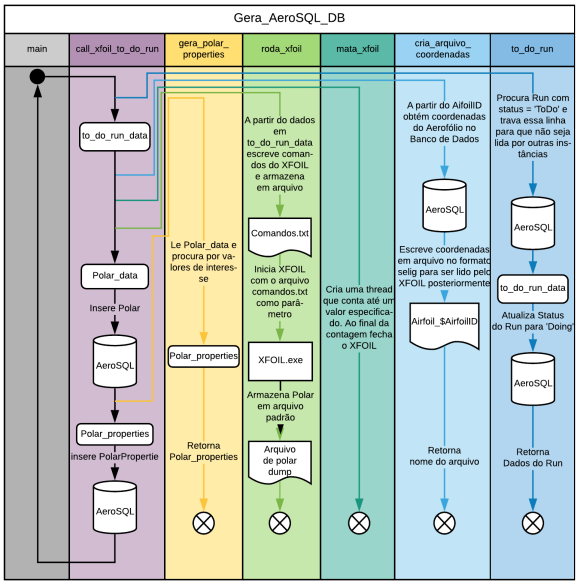


Figura 7: Fluxograma Gera_AeroSQL_DB

Os números de Reynolds escolhidos podem ser observados na Tabela 2, enquanto os números de Mach utilizados estão dispostos no vetor: [0, 0.3, ,0.6, 0.7, 0.75, 0.8].

| Lim. Mínimo | Lim. Máximo | Passo |
|---|---|---|
| 2.000 | 100.000 | 15.000 |
| 125.000 | 250.000 | 25.000 |
| 300.000 | 800.000 | 50.000 |
| 900.000 | 1.500.000 | 100.000 |

Tabela 2: Valores de Reynolds usados na geração das polares

Devido ao funcionamento do XFOIL quanto a convergência da camada limite, foram necessárias duas execuções do programa por Run, uma varrendo os valores positivos de $\alpha$, de $0^o$ a $18.5^o$, e outra os negativos, de $0^o$ a $-18^o$, utilizando passos relativamente pequenos entre alfas calculados.

Os parâmetros de números de painéis e número máximo de iterações foram padronizados como 200 e 150 respectivamente para quase todos os Runs realizados.

Durante a chamada da função *Gera_polar_properties* um algoritmo é utilizado na procura do valor de $Cl_{max}$. Como os valores de $\alpha$ utilizados na produção das polares podem ultrapassar a região de

estol, obtendo um $Cl_{max}$ global após esta região ou, então, sequer atingi-la por falhas de convergência, providências foram tomadas para garantir a aquisição do valor correto deste parâmetro. Quando o algoritmo não encontra nenhum valor aceitável para $Cl_{max}$, este apenas preenche o campo com o valor 0.

O código está disponível em um repositório público no github, acessível através do link `https://github.com/daniel-tucano/AeroSQL-DB.git`. A única modificação feita é que as credenciais para o acesso por uma conta com privilégio de escrita foram removidas dos códigos.

## 2.3 Provisionamento de VMs

Os códigos desenvolvidos foram projetados de forma a tornar possível o uso de múltiplos computadores paralelamente durante para a execução da tarefa, método conhecido como *Computação distribuída*. Qualquer computador ligado a rede poderia executar o script e acelerar a execução do trabalho. Isso permitia que poder de computação fosse adicionado ou removido moduladamente, além de isolar as falhas à apenas unidades específicas, sem comprometer todo o processo.

Através do serviço de computação em cloud Amazon Elastic Computing Cloud (E2C) foram providenciadas até 64 (em pico) instâncias de VMs do tipo t2.medium, operando no Sistema Operacional (SO) Windows Server 2019, totalizando 128 Virtual Cores (vCores). Cada instância foi configurada para inicializar executando um script de PowerShell (PS) personalizado que instala as dependências do código, como o python e as bibliotecas necessárias para a operação do Gera_AeroSQL_DB.

Os scripts de python e outros arquivos de dependências como o próprio XFOIL foram armazenados em um Bucket fornecido pelo serviço Amazon Simple Storage Service (S3) e foram obtidos pelas VMs via AWS CLI, também instalado pelo script em PS.

Para evitar processos entediantes de controle de versão das dezenas de VMs durante a etapa de desenvolvimento, uma VM operando o SO Linux foi utilizada para a execução de um playbook personalizado de Ansyble, que executa um script de PS, atualizando os códigos para suas versões mais recentes.

## 2.4 Extrapolação das Polares

Em [14], pesquisadores da University of Florence e Imperial College realizaram ensaios em túnel de vento em aerofólios distintos para valores de $\alpha$ em todo o espectro, então compararam os resultados com os obtidos por diversos modelos de extrapolação de polares. Dentre os cinco métodos estudados, os que obtiveram os melhores resultados foram os propostos por MONTGOMERIE, Bjorn em [15] e SPERA, David A. em [16].

Este trabalho focou no uso do procedimento proposto por Montgomerie, por sua relativa facilidade de implementação e qualidade nos resultados. Algumas modificações foram feitas com relação ao modelo original para a utilização das informações extrapoladas juntamente com as obtidas *a priori*, de forma que os dados extrapolados completam os dados coletados, preservando a continuidade da polar original.

Por questões de brevidade estas modificações não serão documentas neste trabalho.

O procedimento de extrapolação foi implementado no módulo *Extrapola_polares.py*. Os resultados obtidos pela execução do modelo foram, então, comparados com os obtidos por OSTOWARI, C. em [17]. A Figura 8 demonstra as curvas de $Cl \times \alpha$ do aerofólio NACA 4415, obtidas em $Re = 0.5 \times 10^6$ por Ostowari e pelo XFOIL respectivamente, antes de receberem qualquer alteração.

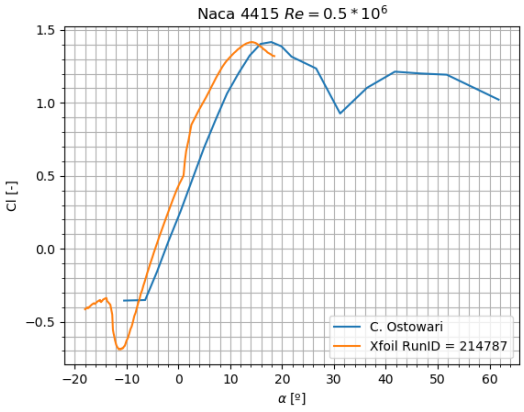


Figura 8: Comparação $Cl \times \alpha$ NACA 4415 $Re = 0.5 \times 10^6$ – Ostowari e XFOIL

A Figura 9 mostra uma comparação entre o procedimento de Montgomorie 'puro' (sem as modificações propostas pelo trabalho) e os dados experimentais obtidos por Ostowari. Já a Figura 10 demonstra a mesma comparação mas com as modificações propostas.

Podemos ainda aplicar o processo de extrapolação na curva obtida pelo XFOIL para entendermos como este irá se comportar com alguns dados presentes no banco. Na Figura 11 é possível ver que a resposta do modelo com entradas vindas de uma fonte menos confiável passa a perder acuracidade.

Isso se dá devido a dependência do valor de $C_L(0)$ na amplitude da região de pós estol, dada pela Equação 1 em [15]:

$$A = 1 + \frac{C_L(0)}{\sin 45^o} \sin \alpha \tag{1}$$

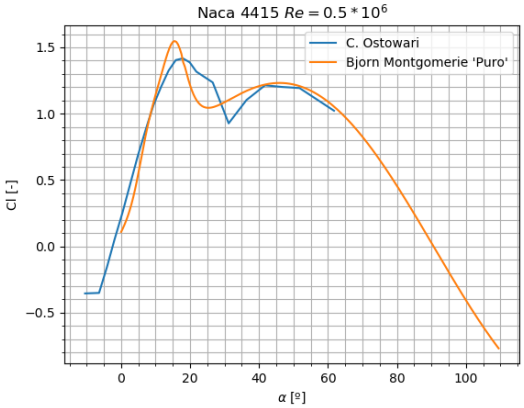


Figura 9: Comparação $Cl \times \alpha$ NACA 4415 $Re = 0.5 \times 10^6$ – Ostowari e Ostowari + Montgomerie 'Puro'

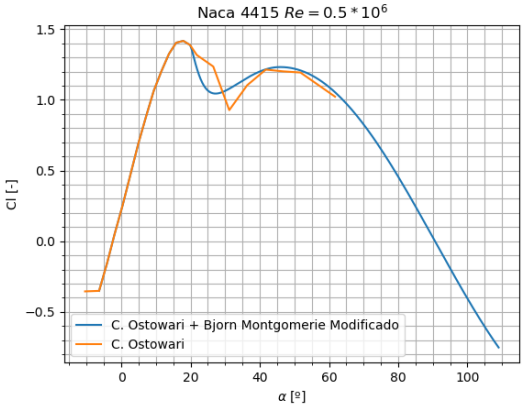


Figura 10: Comparação $Cl \times \alpha$ NACA 4415 $Re = 0.5 \times 10^6$ – Ostowari e Ostowari + Montgomerie Modificado

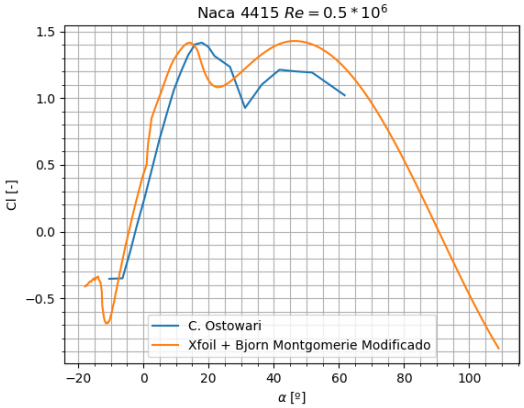


Figura 11: Comparação $Cl \times \alpha$ NACA 4415 $Re = 0.5 \times 10^6$ – Ostowari e Xfoil + Montgomerie Modificado

Portanto erros da entrada neste valor escalam na região que segue o estol.

Comparações das curvas $Cd \times \alpha$ são abordadas na Figura 12 e Figura 13.

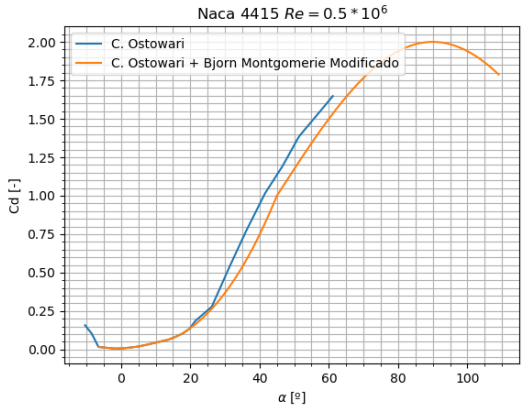


Figura 12: Comparação $Cd \times \alpha$ NACA 4415 $Re = 0.5 \times 10^6$ – Ostowari e Ostowari + Montgomerie Modificado

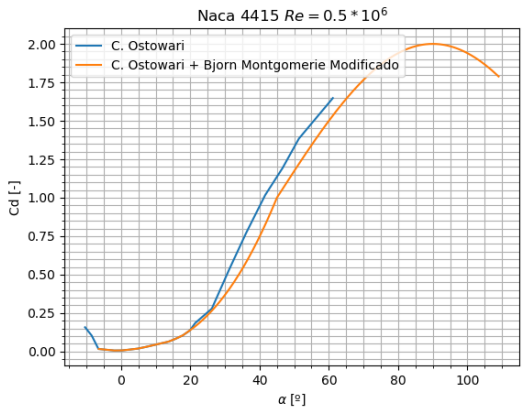


Figura 13: Comparação $Cd \times \alpha$ NACA 4415 $Re = 0.5 \times 10^6$ – Ostowari e Xfoil + Montgomerie Modificado

Após selecionado o método e geradas as polares, um código em *Matlab* denominado *Gera_Runs_extrapoladas.m* foi chamado para que inserisse linhas na tabela Runs com as polares a serem entendidas. Para este processo foram selecionadas somente as Runs que cumpriam com os critério de:

1. $Reynolds \geq 80000$
2. $Mach \leq 0.6$
3. $Cl_{max} \neq 0$
4. $Source = Xfoil$

O script *Extrapola_Polares.py* desenvolvido foi, en-

tão, executado de maneira similar ao *Gera_Aero_SQL_mysql.py*, de forma a estender as polares das Runs selecionadas.

## 2.5 Scripts de acesso ao AeroSQLDB

Juntamente com os arquivos usados para o desenvolvimento do AeroSQL, estão presentes no repositório em github os arquivos *AeroSQLDB_Utilities.py* e *Aero_SQL_Polar_data.m* que fornecem ferramentas em *python* e *Matlab* para o acesso ao banco de dados sem a necessidade da familiaridade do usuário com a linguagem *SQL*. A descrição do funcionamento dos módulos mencionados com suas respectivas função se darão nos próximos tópicos:

### 2.5.1 AeroSQLDB_Utilities.py

O *AeroSQLDB_Utilities.py* é um modulo de python que contém diversas funções que facilitam o acesso ao banco de dados. Para seu uso são necessários os pacotes *matplotlib*, *numpy* e, para a criação de uma conexão com o AeroSQL capaz de fornecer um cursor, o pacote *pymysql*. A Figura 14 sumariza as funções no módulo e seus retornos.

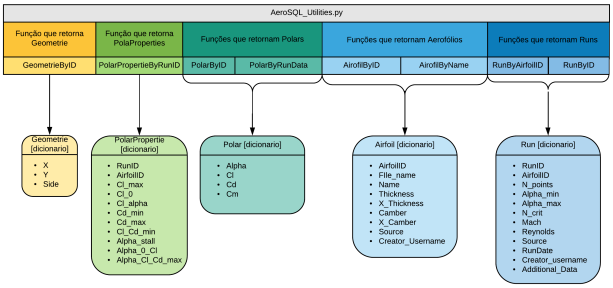


Figura 14: Funções em *AeroSQL_Utilities.py* e seus retornos

O cursor, necessário para o funcionamento das funções, pode ser obtido pelos comandos:

aerosqldb = sql.connect(host='aerosqlrestore.c4co31zewwzk.us-east-1.rds.amazonaws.com', user='Reader', password='', db='AeroSQLDB')

cursor = aerosqldb.cursor()

### 2.5.2 Aero_SQL_Polar_data.m

O arquivo *Aero_SQL_Polar_data.m* é uma função em matlab que permite o usuário acessar as polares presentes no AeroSQLDB a partir de um nome ou *AirfoilID* e número de Reynolds fornecido. A Tabela 3 lista as variáveis e informações relevantes para a utilização da função.

Para acessar um banco de dados através do Matlab é necessária a instalação do Driver MySQL ODBC 5.3 disponível em https://dev.mysql.com/downloads/connector/odbc/5.3.html e, após isso, adicionar uma DNS de Sistema configurada conforme a Figura 15. Para configurar a conexão em um SO Windows são utilizados os seguintes passos: Ferramentas Administrativas> Administrador de Fonte de Dados ODBC> DNS de Sistema> Adicionar> MySQL ODBC 5.3 Unicode Driver> Concluir> Figura 15

Atualmente o AeroSQLDB está hospedado em um servidor provido pela Amazon Web Services (AWS) acessado pelo url aerosqlrestore.c4co31zewwzk.us-east-1.rds.amazonaws.com. no entanto este endereço está sujeito a mudanças, e deve ser checado no repositório do github para obtenção do endereço atualizado.

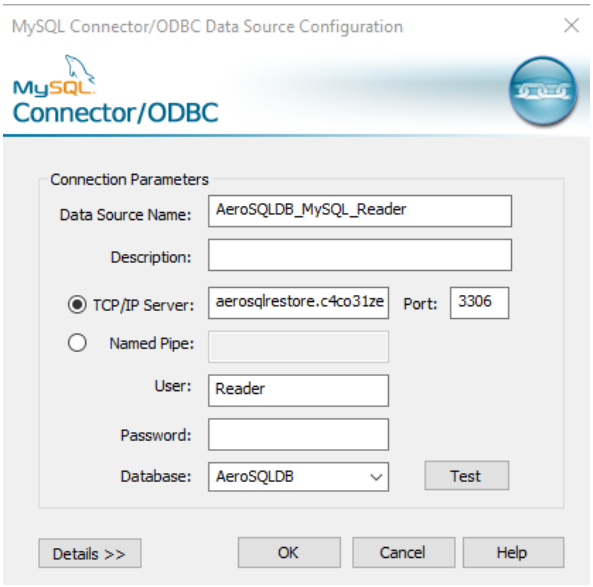


Figura 15: Informações MySQL Connector para uso do *Aero_SQL_Polar_data.m*

# 3 Resultados

Ao todo foram utilizados o equivalente a 7.300 horas de computação por em média 57 instâncias de VMs do tipo t2.medium AWS, capazes de gerar 264.601 Polares por meio do XFOIL das 1.546 geometrias presentes no banco, totalizando cerca de 41.718.805 entradas na tabela Polars. Em cima disso foram realizadas 97.733 extrapolações de polares.

| Variável | Valor Padrão | Valores Aceitos | Descrição |
|---|---|---|---|
| Airfoil-NameID | | Números inteiros e strings | Nome ou ID do aerofólio |
| Re | | Números inteiros maiores que 0 | Número de Reynolds da polar pretendida |
| Aditional-Output | false | 'Polar-Properties' | se 'True' a função passa também o Polar-Properties |
| N_Crit | 9 | Números maiores ou iguais a 0 | Parâmetro de transição de camada limite N_crit da polar pretendida |
| Mach | 0 | Números maiores ou iguais a 0 | Número de Mach da polar pretendida |
| Source | 'Xfoil' | strings | Fonte dos dados da polar pretendida |
| Creator-_username | 'daanr-santiago' | strings | Nome do usuário que inseriu os dados da polar pretendida |
| Output | 'Polar' | 'Polar', 'Alpha', 'Cl', 'Cd', 'Cm' | Output da polar retornada pela função |

Tabela 3: Parâmetros da função *Aero_SQL_Polar_data.m*

# 4 Conclusão

O AeroSQLDB foi estruturado e criado. Esforços foram realizados na tentativa de manter as propriedades de portabilidade, segurança e rastreabilidade dos dados nele inseridos.

Um código que permite computação distribuída foi executado sucedidamente, gerando uma grande quantidade de informações aerodinâmicas prontas para uso. No processo, foram provisionadas e gerenciadas dezenas de máquinas virtuais, que tornaram o processamento dessa imensa quantidade de dados possível em um tempo hábil.

Dentre as polares geradas, milhares foram selecionadas e tiveram suas informações estendidas para compor um maior alcance de ângulos de ataque. Um procedimento que permite a utilização das informações extrapoladas em conjunto com as coletadas pela fonte quais foram baseadas foi utilizado na fabricação destas polares.

Por fim, foram desenvolvidas ferramentas para facilitar o acesso a plataforma por usuários de python e *Matlab*, de forma a incentivar seu uso por quem não tem experiência em SQL.

# Trabalhos Futuros

Visando tornar o AeroSQLDB ainda mais acessível a comunidade, futuros trabalhos incluem o desenvolvimento de ferramentas que tornem o acesso ao banco de dados ainda mais amigável, com maneiras de interação a partir de interfaces gráficas.

Esforços para a inclusão dos dados experimentais presentes em [5–9] e [17] também estão entre as tarefas pretendidas.

# Referências


[1] Yao Zhang, Woong Je Sung, and Dimitri N Mavris. Application of convolutional neural network to predict airfoil lift coefficient. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 1903, 2018.

[2] Jichao Li, Mohamed Amine Bouhlel, and Joaquim RRA Martins. Data-based approach for fast airfoil analysis and optimization. *AIAA Journal*, 57(2):581–596, 2019.

[3] Airfoil Tools. Disponível em< http://airfoiltools.com>. *Acesso em*, 2, 2020.

[4] Michael Selig. Uiuc airfoil coordinates database. *UIUC Applied Aerodynamics Group, Urbana, IL, accessed May*, 18:2016, 2016.

[5] Michael S. Selig. *Summary of low speed airfoil data*. SoarTech Publications, 1995.

[6] Michael S. Selig. *Summary of low-speed airfoil data. Vol. 2*. SoarTech Publications, 1996.

[7] Michael S. Selig. *Summary of low speed airfoil data Vol. 3*. SoarTech Publications, 1997.

[8] Michael S Selig and Bryan D. McGranahan. Wind tunnel aerodynamic tests of six airfoils for use on small wind turbines. In *Collection of ASME Wind Energy Symposium Technical Papers AIAA Aerospace Sciences Meeting and Exhibit*, Collection of ASME Wind Energy Symposium Technical Papers AIAA Aerospace Sciences Meeting and Exhibit, pages 558–576, 7 2004.

[9] Gregory A Williamson, Bryan D McGranahan, Benjamin A Broughton, Robert W Deters, John B Brandt, and Michael S Selig. Summary of low-speed airfoil data, vol. 5. *Department of Aerospace Engineering, University of Illinois at Urbana-Champaign*, 2012.

[10] Mark Drela. Xfoil: An analysis and design system for low reynolds number airfoils. In *Low Reynolds number aerodynamics*, pages 1–12. Springer, 1989.

[11] Eastman N Jacobs and Albert Sherman. Airfoil section characteristics as affected by variations of the reynolds number. *NACA report*, 586:227–264, 1937.

[12] Tousif Ahmed, Md Tanjin Amin, SM Rafiul Islam, and Shabbir Ahmed. Computational study of flow around a naca 0012 wing flapped at different flap angles with varying mach numbers. *Global Journal of Research In Engineering*, 2014.

[13] Authors Malmuth, R Crites, Joel Everhart, P Newman, and W Sickles. *TRANSONIC WIND TUNNEL WALL INTERFERENCE*, pages 5–2 through 5–107. 10 1998.

[14] Alessandro Bianchini, Francesco Balduzzi, John M Rainbird, Joaquim Peiro, J Michael R Graham, Giovanni Ferrara, and Lorenzo Ferrari. An experimental and numerical assessment of airfoil polars for use in darrieus wind turbines—part ii: Post-stall data extrapolation methods. *Journal of Engineering for Gas Turbines and Power*, 138(3), 2016.

[15] Björn Montgomerie. Methods for root effects, tip effects and extending the angle of attack range to±180 deg, with application to aerodynamics for blades on wind turbines and propellers. *FOI, Swedish Defence Research Agency, Stockholm, Sweden, Report No. FOI*, 2004.

[16] David A Spera. Models of lift and drag coefficients of stalled and unstalled airfoils in wind turbines and wind tunnels. 2008.

[17] C Ostowari and D Naik. Post-stall wind tunnel data for naca 44xx series airfoil sections. Technical report, Texas A and M Univ., College Station (USA). Dept. of Aerospace Engineering, 1985.

## Contato do Autor

**Autor**
Daniel Ribeiro Santiago
email: danielribeirosantiago.tucano@gmail.com

**Co-Autor**
Lucas Otaviano Alves
email: lucasalves.tucano@gmail.com

**Professor Orientador**
Odenir de Almeida
email: odenir.almeida@ufu.br

UNIVERSIDADE FEDERAL DE UBERLÂNDIA