`amiet_tools`: um pacote em Python para predição de ruído de interação turbulência-aerofólio

EM CONSTRUÇÃO

Fabio Casagrande Hirono
`fchirono@gmail.com`

Fevereiro 2021

Resumo

O pacote `amiet_tools` (`AmT`), escrito em linguagem Python, é uma implementação do modelo analítico de Amiet [1] para predição de ruído de interação turbulência-aerofólio, com extensões. As funções permitem o cálculo da flutuação de pressão na superfície do aerofólio (i.e. a distribuição da fonte acústica) gerado em resposta à turbulência incidente, e do campo acústico radiado por esta interação. A turbulência incidente pode ser uma única rajada senoidal, ou uma soma incoerente de rajadas com amplitudes definidas por um espectro energético. Outras funções includem a modelagem de efeitos de convecção e refração na propagação sonora para modelar medições acústicas realizadas em túneis de vento abertos ou fechados. Uma publicação de referência está disponível [2], e o pacote pode ser encontrado através do link https://github.com/fchirono/amiet_tools.

1 Introdução

1.1 Contexto

O ruído gerado pela interação entre um escoamento turbulento e uma superfície sólida, tal como um aerofólio, é uma das principais fontes de ruído banda larga na indústria de aviação moderna. Apesar dos mecanismos físicos fundamentais desta fonte de ruído estarem bem estabelecidos, métodos eficientes para redução de ruído de interação turbulência-aerofólio ainda são objeto de estudos e pesquisas. Devido ao seu baixo custo computacional, modelos analíticos são frequentemente adotados para analisar os comportamentos físicos fundamentais e escalas do problema. Entre os modelos analíticos mais populares para predição de ruído de aerofólio, o modelo de Amiet [1] para um aerofólio do tipo placa plana em campo livre é frequentemente considerado um dos mais importantes devido a sua simplicidade formal e extensibilidade.

Em resumo, o modelo de Amiet decompõe um escoamento turbulento em uma soma de Fourier, onde cada componente constitui uma "rajada" turbulenta senoidal, e calcula a flutuação de pressão — também denominada "salto de pressão" — sobre a superfície do aerofólio em resposta à interação com cada rajada. Seguindo a analogia acústica de Ffowcs Williams e Hawkings, a radiação acústica pode ser calculada usando uma distribuição de fontes tipo dipolo pontual sobre a superfície do aerofólio, cuja amplitude é dada pelo salto de pressão. O campo acústico radiado é então calculado através de uma integral de radiação, levando-se em conta os efeitos de convecção e/ou refração do som na propagação entre a fonte e o(s) observador(es).

Ao assumir que o aerofólio possui envergadura infinita (uma aproximação teórica para aerofólios de alta razão de aspecto, ou para envergaduras muito maiores que o comprimento de onda acústico), pode-se mostrar que uma única rajada senoidal será responsável por todo o ruído recebido por um observador distante. A expressão resultante desta simplificação relaciona

a densidade espectral de potência (*power spectral density* – PSD) do ruído visto pelo observador diretamente às propriedades do escoamento turbulento e à geometria do aerofólio, evitando assim o cálculo da distribuição de fonte acústica sobre a superfície do aerofólio e da integral de radiação completa. Esta forma simplificada do modelo de Amiet é a mais popular, e é geralmente suficiente para se obter as tendências gerais do ruído radiado.

Porém, tais aproximações podem ser simplórias demais para problemas mais avançados, tais como o cálculo explícito do campo acústico próximo ao aerofólio, ou a inclusão de efeitos de envergadura finita ou de rajadas múltiplas. Sob estas condições, o modelo de Amiet requer o cálculo separado da distribuição de fontes e da integral de radiação, realizados de forma probabilística no domínio das funções densidade espectral cruzada (*cross-power spectral density* – CSD), aumentando significativamente a complexidade das operações necessárias. Neste caso, é interessante buscar uma implementação de referência do modelo de Amiet completo, permitindo que os pesquisadores possam obter resultados confiáveis rapidamente e não perder tempo com detalhes de implementação.

Este documento busca introduzir o pacote `amiet_tools` (abreviado por "`AmT`"), uma implementação de referência do modelo de Amiet completo para ruído de interação turbulência-aerofólio [2], e explicar o seu uso. O pacote `AmT` é desenvolvido em linguagem Python e disponibilizado com uma licença de código aberto, permitindo a sua distribuição, modificação e uso por diversos usuários em múltiplos sistemas operacionais. O pacote encontra-se hospedado na plataforma GitHub através do link:

- https://github.com/fchirono/amiet_tools .

1.2 Referências teóricas recomendadas

Este documento não busca explicar completamente a teoria por trás do modelo de Amiet. Para mais detalhes e informações sobre modelo analítico completo, recomendamos que o leitor consulte outras referências na literatura, entre as quais recomendamos:

- R. Amiet, "*Acoustic radiation from an airfoil in a turbulent stream*", Journal of Sound and Vibration, Vol. 41, No. 4:407–420, 1975 [1];
- G. Reboul, "*Modélisation du bruit à large bande de soufflante de turboréacteur*", PhD Thesis, Laboratoire de Mécanique des Fluides et d'Acoustique – École Centrale de Lyon, Lyon – France, 2010 [3];
- M. Roger, "*Broadband noise from lifting surfaces: Analytical modeling and experimental validation*", in: R. Camussi (Ed.), "*Noise Sources in Turbulent Shear Flows: Fundamentals and Applications*", Springer-Verlag, 2013 [4];
- L. de Santana, "*Semi-analytical methodologies for airfoil noise prediction*", PhD Thesis, Faculty of Engineering Sciences – Katholieke Universiteit Leuven, Leuven, Belgium, 2015 [5];
- F. Casagrande Hirono, "*Far-Field Microphone Array Techniques for Acoustic Characterisation of Aerofoils*", PhD Thesis, Institute of Sound and Vibration Research, University of Southampton, Southampton – UK, 2018 [6].

1.3 O modelo de Amiet

Esta seção busca descrever os principais conceitos por trás do modelo de Amiet. O modelo estima as características do campo acústico radiado como função das condições experimentais do problema: as propriedades do escoamento turbulento, a geometria do aerofólio, e a localização dos observadores no campo acústico. Todos os cálculos são realizados no domínio da frequência

(isto é, assume-se que todas as variáveis acústicas oscilam harmonicamente), e a cada frequência de interesse são realizadas três etapas:

1. Decomposição do escoamento turbulento em múltiplas rajadas senoidais, cada uma caracterizada por seu número de onda lateral k_ψ;
2. Cálculo do "salto" de pressão Δp sobre o aerofólio em resposta a cada rajada turbulenta;
3. Cálculo do campo acústico através de uma integral de radiação, levando em consideração efeitos de convecção e/ou refração na propagação do som.

As condições experimentais do problema estão indicadas na Figura 1: um aerofólio do tipo placa plana retangular, de espessura infinitesimal, está imerso em um escoamento turbulento subsônico de velocidade U_x e número Mach $M_x = U_x/c_0$ na direção $+x$. O aerofólio possui corda $c = 2b$ e envergadura $L = 2d$ (e portanto, razão de aspecto L/c), e a origem do sistema de coordenadas é posicionada no centro do aerofólio, com o eixo $+z$ perpendicular ao aerofólio (apontando "fora da página" na Fig. 1). Pontos no espaço ao redor do aerofólio são indicados por um vetor $\mathbf{r} = (x, y, z)$, e pontos na superfície do aerofólio são indicados com um caractere subescrito "$_s$", da forma $\mathbf{r}_s = (x_s, y_s, z_s)$. O aerofólio encontra-se no plano $z_s = 0$. A análise aqui apresentada encontra-se no domínio da frequência angular $\omega = 2\pi f$, e a dependência temporal implícita é da forma $e^{+j\omega t}$.

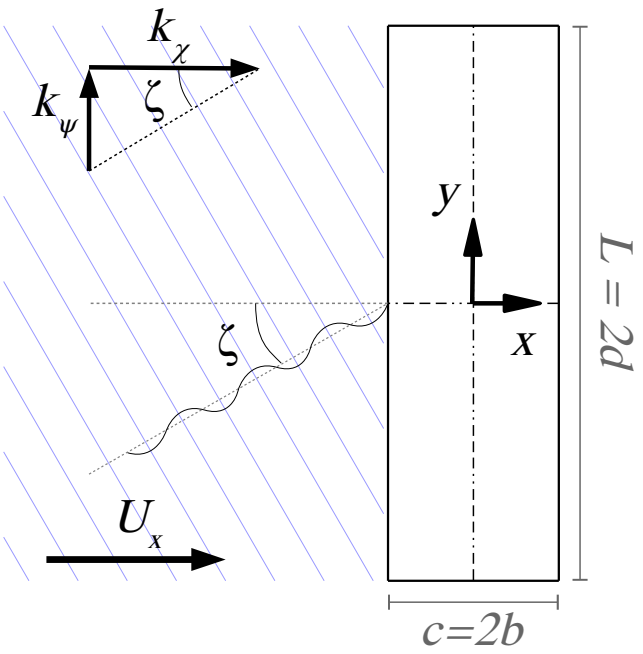


Figura 1: Diagrama das condições experimentais

1.3.1 Interação com uma rajada única

Um espectro contínuo de rajadas turbulentas senoidais "congeladas" move-se no plano $z = 0$ com velocidade de convecção U_x, e cada rajada é definida por um par de números de onda hidrodinâmicos (isto é, vorticais) longitudinal e lateral (k_χ, k_ψ). Uma rajada única representa uma perturbação de velocidade vertical, conforme indicado na Figura 2, e é da forma

$$w(x, y, \omega) = \hat{w}_0(\kappa_\chi, k_\psi) e^{-j\kappa_\chi x} e^{-jk_\psi y}, \tag{1}$$

onde $\kappa_\chi = \omega/U_x$ é o número de onda longitudinal da rajada, associado unicamente à frequência ω do ruído radiado, e $\hat{w}_0$ é a amplitude desta rajada.

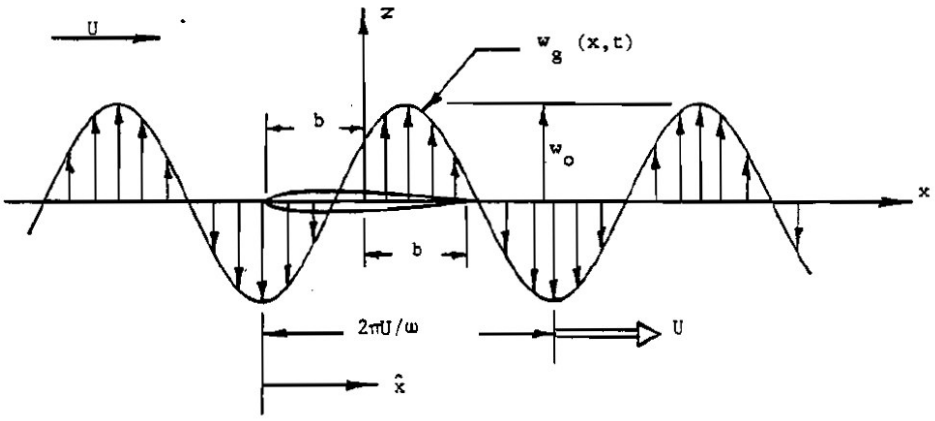


Figura 2: Diagrama 2D indicando uma perturbação de velocidade vertical $w(x,t)$ ("rajada" turbulenta) interagindo com um aerofólio. (Figura reproduzida a partir dos materiais da disciplina *AA-220 – Aerodinâmica Não Estacionária*, Prof. Dr. Roberto Gil Annes da Silva, ITA, Brasil – http://www.aer.ita.br/~gil/disciplinas/aa-220/aa2206.pdf).

Em resposta à incidência de uma única rajada turbulenta, haverá uma flutuação na pressão sobre a superfície do aerofólio, denominada aqui de "salto" de pressão Δp. Este salto de pressão será da forma

$$\Delta p(x_s, y_s, \omega) = 2\pi\rho_0 \hat{w}_0(\kappa_\chi, k_\psi)\ g(x_s, \kappa_\chi, k_\psi) e^{-jk_\psi y_s}, \tag{2}$$

onde g é o salto de pressão não-dimensional sobre a corda x_s devido a uma rajada (κ_χ, k_ψ).

Segundo a analogia de Ffowcs Williams e Hawkings, o ruído gerado por esta interação pode ser calculado usando uma distribuição equivalente de fontes pontuais tipo dipolo, com a flutuação de pressão Δp sobre o aerofólio exercendo o papel de amplitude dos dipolos equivalentes. Analiticamente, a pressão acústica $p(\mathbf{r}, \omega)$ recebida por um observador no ponto $\mathbf{r} = (x, y, z)$ é dada por

$$p(\mathbf{r}, \omega) = \int_{-d}^{+d} \int_{-b}^{+b} \Delta p(\mathbf{r}_s, \omega) \frac{\partial}{\partial z_s} G_{U_x}(\mathbf{r}|\mathbf{r}_s, \omega)\, \mathrm{d}x_s\, \mathrm{d}y_s, \tag{3}$$

onde o termo $\partial G_{U_x}(\mathbf{r}|\mathbf{r}_s, \omega)/\partial z_s$ representa a função de transferência de uma fonte tipo dipolo localizada em $\mathbf{r}_s$ e o observador em $\mathbf{r}$, onde ambos estão imersos em um escoamento uniforme de velocidade U_x. Esta função de transferência é dada por

$$\frac{\partial}{\partial z_s} G_{U_x}(\mathbf{r}|\mathbf{r}_s, \omega) = \left(jk_0 + \frac{1}{\overline{r}}\right) \frac{(\overline{z} - \overline{z}_s)}{\beta \overline{r}} \frac{e^{-jk_0 \overline{r}}}{4\pi\beta^2 \overline{r}} e^{jk_0 M_x (\overline{\overline{x}} - \overline{\overline{x}}_s)}, \tag{4}$$

onde a sobrelinha representa uma transformação de variáveis do tipo Prandtl-Glauert para descrever os efeitos do escoamento no campo acústico [7]:

$$\overline{\mathbf{r}} = (\overline{\overline{x}}, \overline{y}, \overline{z}) = \left(\frac{x}{\beta^2}, \frac{y}{\beta}, \frac{z}{\beta}\right), \quad \overline{r} = \|\overline{\mathbf{r}} - \overline{\mathbf{r}}_s\|, \quad \beta = \sqrt{1 - M_x^2}. \tag{5}$$

Se as condições experimentais adotam um túnel de vento aberto, a função de transferência dada pela Eq. 4 pode ser modificada para incluir efeitos de refração das ondas acústicas na camada cisalhante gerada entre o escoamento gerado pelo túnel e o ar estacionário ao redor deste. Mais detalhes podem ser encontrados nas referências [2, 6], por exemplo.

Conforme descrito acima, a flutuação de pressão Δp em resposta a uma única rajada turbulenta será completamente coerente — isto é, todos os pontos na superfície do aerofólio flutuarão

na mesma frequência e perfeitamente em fase, de forma estatisticamente *dependente*, e o campo acústico p gerado por esta interação exibirá efeitos de interferência construtiva e destrutiva entre as diferentes regiões do aerofólio.

1.3.2 Interação com múltiplas rajadas

Um escoamento turbulento isotrópico é um processo estocástico, e pode ser decomposto em um espectro contínuo de rajadas turbulentas incoerentes — isto é, estatisticamente *independentes* entre si. Portanto, a soma das contribuições de diferentes rajadas deve ser realizada de forma probabilística através das funções densidade espectral cruzada (CSD) da flutuação de pressão $S_{\Delta p\Delta p'}$ e do campo acústico $S_{pp'}$, que capturam as relações de magnitude e fase média destas grandezas entre dois pontos arbitrários.

A densidade espectral cruzada da flutuação de pressão $S_{\Delta p\Delta p'}(x_s, x'_s, y_s, y'_s, \omega)$ entre dois pontos (x_s, y_s) e (x'_s, y'_s) na superfície do aerofólio é dada por

$$S_{\Delta p\Delta p'}(x_s, x'_s, y_s, y'_s, \omega) = \lim_{T\to\infty}\left[\frac{\pi}{T}\mathrm{E}\left\{\Delta p(x_s, y_s, \omega)\Delta p^*(x'_s, y'_s, \omega)\right\}\right] \tag{6}$$

$$= (2\pi\rho_0)^2 U_x \int_{-\infty}^{+\infty} \Phi_{ww}(\kappa_\chi, k_\psi) g(x_s, \kappa_\chi, k_\psi) g^*(x'_s, \kappa_\chi, k_\psi) e^{-jk_\psi(y_s - y'_s)}\,\mathrm{d}k_\psi, \tag{7}$$

onde $\Phi_{ww}(k_\chi, k_\psi)$ é a densidade espectral de número de onda da turbulência incidente. Observe que esta CSD é obtida a partir da integração sobre todos os números de onda laterais k_ψ da turbulência, e portanto incorpora os efeitos de múltiplas rajadas.

Por sua vez, a densidade espectral cruzada do campo acústico $S_{pp'}(\mathbf{r}, \mathbf{r}', \omega)$ entre dois observadores em $\mathbf{r}$ e $\mathbf{r}'$ é dada por

$$S_{pp'}(\mathbf{r}, \mathbf{r}', \omega) = \lim_{T\to\infty}\left[\frac{\pi}{T}\mathrm{E}\left\{p(\mathbf{r}, \omega)p^*(\mathbf{r}', \omega)\right\}\right] \tag{8}$$

$$= \int_{-d}^{+d}\int_{-b}^{+b}\int_{-d}^{+d}\int_{-b}^{+b} S_{\Delta p\Delta p'}(\mathbf{r}_s, \mathbf{r}'_s, \omega)\frac{\partial}{\partial z_s}G_{U_x}(\mathbf{r}|\mathbf{r}_s, \omega)\frac{\partial}{\partial z'_s}G^*_{U_x}(\mathbf{r}'|\mathbf{r}'_s, \omega)\,\mathrm{d}x_s\,\mathrm{d}y_s\,\mathrm{d}x'_s\,\mathrm{d}y'_s. \tag{9}$$

1.3.3 Densidade espectral de número de onda da turbulência

Para a densidade espectral de número de onda da turbulência $\Phi_{ww}(k_\chi, k_\psi)$, pode-se por exemplo adotar o espectro energético de número de onda de von Karman [1], definido a partir da velocidade vertical média quadrática $\overline{w^2}$ e da escala de comprimento integral Λ do escoamento. Esta densidade espectral possui a forma

$$\Phi_{ww}(k_\chi, k_\psi) = \frac{4}{9\pi}\frac{\overline{w^2}}{k_e^2}\frac{\check{k}_\chi^2 + \check{k}_\psi^2}{\left(1 + \check{k}_\chi^2 + \check{k}_\psi^2\right)^{7/3}}, \tag{10}$$

$$k_e = \frac{\sqrt{\pi}}{\Lambda}\frac{\Gamma(5/6)}{\Gamma(1/3)}, \tag{11}$$

onde $\check{k}_i = k_i/k_e$ e $\Gamma(\cdot)$ é a função gama. Outros modelos, tais como o de Liepmann [8], também são possíveis.

2 Usando o pacote

2.1 Requerimentos

O pacote `amiet_tools` foi desenvolvido em linguagem Python 3.x, e tem como dependências os pacotes `numpy` e `scipy`. Para plotar resultados, esse tutorial usa o pacote `matplotlib`, mas esta não é uma dependência do `AmT`. Estes pacotes estão inclusos na "*Anaconda Python Distribution*"[1], uma distribuição Python gratuita e de código aberto. A distribuição Anaconda é usada para desenvolver o pacote `AmT`, e é a distribuição recomendada para usar `AmT`.

2.2 Definindo as variáveis

2.2.1 Inserindo variáveis experimentais – método recomendado

O pacote `AmT` pode ler as variáveis relacionadas às condições do experimento e à geometria do aerofólio a partir de dois arquivos de texto. Usaremos como exemplo os dois arquivos de configuração do experimento "DARP2016", realizado no túnel de vento DARP da Universidade de Southampton, Reino Unido, e disponíveis no repositório GitHub do projeto. Para este exemplo, assumimos que ambos os arquivos encontram-se no diretório de trabalho atual do console Python.

O primeiro arquivo, chamado `DARP2016_TestSetup.txt`, e os seus conteúdos estão indicados na Listagem 1 abaixo; note que linhas vazias são ignoradas, e o caractere "#" indica comentários. O programa lerá os valores em cada linha do arquivo, e os interpretará conforme indicado nas respectivas linhas. O arquivo deve conter as seguintes variáveis, apresentadas nas seguintes unidades, e listadas em ordem:

- `c0`: velocidade do som c_0 [m/s];
- `rho0`: densidade do ar ρ_0 [kg/m^3];
- `p_ref`: pressão acústica de referência p_{ref} [Pa RMS];
- `Ux`: velocidade do escoamento U_x [m/s];
- `turb_intensity`: intensidade de turbulência $\sqrt{\overline{w^2}}/U_x$ [adimensional];
- `length_scale` escala de comprimento de turbulência Λ [m];
- `z_sl`: altura da camada cisalhante z_{sl} (para túnel de vento de jato aberto) [m].

```
# amiet_tools Test Setup file
#
# DARP2016 test setup, ISVR, Univ. of Southampton, UK

# Acoustic characteristics
340.     # c0          Speed of sound [m/s]
1.2      # rho0        Air density [kg/m**3]
20e-6    # p_ref       Ref acoustic pressure [Pa RMS]

# turbulent flow properties
60       # Ux                  flow velocity [m/s]
0.025    # turb_intensity      turbulence intensity = u_rms/U
0.007    # length_scale        turb length scale [m]
```

[1] Disponível em https://www.anaconda.com/products/individual.

```
# shear layer height
-0.075   # z_sl            Shear layer height (aerofoil is at z=0) [m]
```

Listing 1: Arquivo `DARP2016_TestSetup.txt`

Devido ao sistema de coordenadas adotado, camadas cisalhantes com $z_{sl} > 0$ estão *acima* do aerofólio, e indicam que os observadores também estão acima do aerofólio; similarmente, camadas cisalhantes com $z_{sl} < 0$ estão *abaixo* do aerofólio, e indicam que os observadores também localizam-se abaixo do aerofólio.

O segundo arquivo chama-se `DARP2016_AirfoilGeom.txt`, e contém variáveis relacionadas à geometria do aerofólio. Seus conteúdos estão indicados na Listagem 2 abaixo. Este arquivo deve conter as seguintes variáveis, listadas em ordem:

- `b`: semi corda do aerofólio b [m];
- `d`: semi envergadura do aerofólio d [m];
- `Nx`: número de pontos para amostragem da corda (não uniforme) N_x;
- `Ny`: número de pontos para amostragem da envergadura (uniforme) N_y.

```
# amiet_tools AirfoilGeom file
#
# DARP2016 flat plate airfoil, ISVR, Univ. of Southampton, UK

0.075    # b          airfoil half chord [m]
0.225    # d          airfoil half span [m]
100      # Nx         number of chordwise points (non-uniform sampl)
101      # Ny         number of spanwise points (uniform sampl)
```

Listing 2: Arquivo `DARP2016_AirfoilGeom.txt`

Para carregar os valores descritos nos arquivos de texto, recomenda-se usar os comandos em Python descritos na Listagem 3. Primeiramente, o pacote `numpy` e o pacote `AmT` são importados pelo script. A função `AmT.loadTestSetup` é usada para carregar as variáveis relacionadas às condições do experimento a partir do arquivo de texto fornecido como argumento, e cria um objeto — aqui chamado de `DARP2016Setup` — para armazenar estas variáveis. Este objeto é uma instância da classe `TestSetup`, definida no pacote `AmT`; mais informações sobre as classes usadas no pacote `AmT` podem ser encontradas na Seção 5.

Similarmente, usa-se a função `AmT.loadAirfoilGeom` para carregar variáveis relacionadas à geometria do aerofólio a partir do arquivo de texto fornecido como argumento. Esta função criará um objeto — aqui chamado de `DARP2016Airfoil` — para armazenar as dimensões do aerofólio e as configurações de amostragem de sua superfície. Este objeto é uma instância da classe `AirfoilGeom`, definida no pacote `AmT`; mais informações sobre as classes usadas no pacote `AmT` podem ser encontradas na Seção 5.

```
import numpy as np
import amiet_tools as AmT

# carrega as condicoes do experimento a partir de um arquivo de texto
DARP2016Setup = AmT.loadTestSetup('DARP2016_TestSetup.txt')

# carrega a geometria do aerofolio a partir de um arquivo de texto
DARP2016Airfoil = AmT.loadAirfoilGeom('DARP2016_AirfoilGeom.txt')
```

Listing 3: Método recomendado para importar pacotes e carregar variáveis

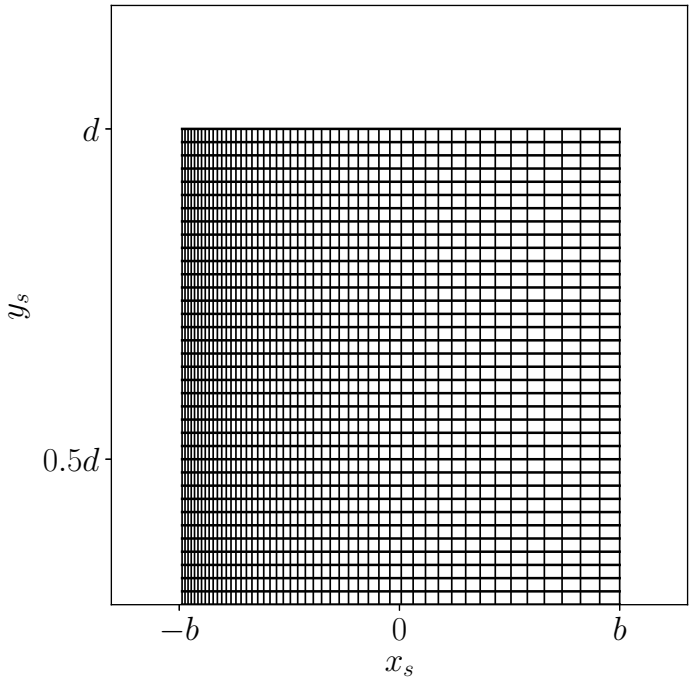


Figura 3: Exemplo de discretização da superfície do aerofólio próximo à ponta da envergadura, com $N_y = 101$ pontos distribuidos uniformemente ao longo da envergadura, e $N_x = 50$ pontos distribuidos não uniformemente ao longo da corda.

A amostragem não uniforme da corda adotada no pacote `AmT` utiliza uma maior concentração de pontos próximo ao bordo de ataque do aerofólio, de forma a melhor capturar o comportamento do salto de pressão g (Eq. 2) próximo ao bordo de ataque. Um exemplo desta amostragem está indicada na Figura 3, onde usamos $(N_x, N_y) = (50, 101)$ pontos para fins de ilustração do conceito. Para predições acústicas, recomenda-se adotar $(N_x, N_y) = (100, 101)$ pontos para obter convergência em altas frequências. Mais detalhes da amostragem podem ser encontrados nas referências [2, 6].

2.2.2 Calculando variáveis secundárias

Após carregar as variáveis a partir dos arquivos de texto, as funções `AmT.loadTestSetup` e `AmT.loadAirfoilGeom` calculam algumas variáveis secundárias e armazena-as como atributos dos objetos criados. Estas variáveis secundárias são:

- Função `AmT.loadTestSetup`:
  - `Mach`: número Mach $M_x = U_x/c_0$;
  - `beta`: parâmetro de Prandtl-Glauert $\beta = \sqrt{1 - M_x^2}$;
  - `flow_param = (flow_dir, Mach)`: tupla de parâmetros do escoamento, onde `flow_dir` é um caractere único indicando a direção do escoamento, e `flow_dir = 'x'` por definição;
  - `dipole_axis`: caractere único indicando a direção do eixo dos dipolos acústicos, e `dipole_axis = 'z'` por definição.
- Função `AmT.loadAirfoilGeom`:

- `XYZ_airfoil`: array `numpy` de dimensões $(3, N_x, N_y)$ contendo as coordenadas de cada amostra na superfície do aerofólio;
- `dx`: array `numpy` de dimensões $(N_x,)$ contendo os intervalos de amostragem (não-uniforme) da corda;
- `dy`: intervalo de amostragem (uniforme) da envergadura.

Estes cálculos são realizados internamente, e o usuário geralmente não precisa se preocupar com eles.

2.2.3 Inserindo variáveis experimentais – método alternativo

Alternativamente, estas funções podem ser chamadas sem nenhum argumento de entrada; os objetos resultantes conterão as configurações do teste "DARP2016" por padrão, e estes podem ser modificados posteriormente dentro do script. Um exemplo deste método alternativo encontra-se na Listagem 4, onde três variáveis são modificadas e recebem novos valores. Note, porém, que é necessário executar os métodos internos `MyTestSetup._calc_secondary_vars()` e `MyAirfoil._calc_grid()` para recalcular as variáveis secundárias depois das variáveis primárias serem redefinidas.

```
import amiet_tools as AmT

# Inicializando os objetos para armazenar variaveis
MyTestSetup = AmT.loadTestSetup()
MyAirfoil = AmT.loadAirfoilGeom()

# Redefinindo valores
MyTestSetup.Ux = 120
MyTestSetup.turb_intensity = 0.05
MyAirfoil.b = 0.15

# IMPORTANTE: recalcular as variaveis secundarias em ambos os objetos
MyTestSetup._calc_secondary_vars()
MyAirfoil._calc_grid()
```

Listing 4: Método alternativo para carregar variáveis

Desta forma, as diversas variáveis relacionadas ao setup experimental e ao aerofólio utilizado podem ser importadas e combinadas; por exemplo, pode-se simular dois aerofólios diferentes dentro das mesmas condições experimentais.

2.2.4 Acessando as variáveis experimentais

As variáveis lidas dos arquivos de texto podem agora ser acessadas como atributos dos objetos criados: por exemplo, usa-se o comando `DARP2016Setup.c0` para acessar a velocidade do som c_0 contida neste objeto. As variáveis também podem ser exportadas diretamente para o "namespace" atual utilizando o método `export_values` (pertencente às duas classes), conforme indicado na Listagem 5. Desta forma, as variáveis agora existem tanto como atributos dos objetos criados quanto como variáveis independentes dentro do namespace atual, facilitando certos cálculos e testes iterativos.

```
# exportando variaveis para o namespace atual
(c0, rho0, p_ref, Ux, turb_intensity, length_scale, z_sl, Mach, beta,
flow_param, dipole_axis) = DARP2016Setup.export_values()

(b, d, Nx, Ny, XYZ_airfoil, dx, dy) = DARP2016Airfoil.export_values()
```

Listing 5: Exportando as variáveis para o "namespace" atual

2.2.5 Definindo as variáveis temporais

As variáveis relacionadas à frequência temporal f (em Hz) são criadas dentro do próprio código, dependendo do interesse do usuário. Como exemplo, vamos definir uma única frequência f_0 a partir da frequência não dimensional normalizada pela corda $k_0c = 5$. Para tal, definimos o valor desejado de f_0 (a partir da frequência não dimensional k_0c), e passamos esta variável e o objeto `DARP2016TestSetup` como argumentos para a função `AmT.FrequencyVars`, resultando no objeto aqui chamado `FreqVars`. Este objeto é uma instância da classe `FrequencyVariables`, definida no pacote `AmT`; mais informações sobre as classes usadas no pacote `AmT` podem ser encontradas na Seção 5.

```
# frequency of operation
kc = 5                          # chordwise normalised frequency = k0*(2*b)
f0 = kc*c0/(2*np.pi*(2*b))      # approx 1.8 kHz

FreqVars = AmT.FrequencyVars(f0, DARP2016Setup)
(k0, Kx, Ky_crit) = FreqVars.export_values()
```

Listing 6: Criando variáveis relacionadas à frequência

Similarmente, os valores aqui calculados podem ser acessados como atributos do objeto `FreqVars`, ou exportados para o namespace atual através do método `FreqVars.export_values`. Esta função também calcula as seguintes variáveis secundárias para uma dada frequência $f0$:

- `k0`: número de onda acústico k_0 [rad/m];
- `Kx`: número de onda longitudinal da rajada $\kappa_\chi = \omega/U_x$;
- `Ky_crit`: número de onda crítico lateral de rajada k_ψ^{crit}.

Para cálculos envolvendo múltiplas frequências, recomenda-se realizar um loop sobre as frequências de interesse, e recriar o objeto `FreqVars` ao início de cada iteração.

3 Exemplos

Esta Seção irá descrever alguns aspectos dos scripts de teste contidos dentro do diretório `test_scripts`, armazenados na página do projeto `AmT` no GitHub. Este tutorial descreverá o uso das funções contidas no pacote `AmT` e seus resultados, mas não descreverá em detalhes outras partes dos códigos, tais como comandos usados para plotar uma figura.

3.1 Exemplo 1: Interação de Rajada Única

Como primeiro exemplo, vamos usar o pacote `AmT` para calcular a distribuição do "salto" de pressão $\Delta p(x_s, y_s)$ na superfície do aerofólio em resposta a uma única rajada turbulenta em uma única frequência, e as funções de diretividade do campo acústico nas direções da corda e da envergadura. Este exemplo corresponde ao script de teste `TestScript1_SingleGustPlots.py`.

A primeira parte do script está apresentada na Listagem 7. Primeiramente, seguimos as instruções descritas na Seção 2 para importar os pacotes `numpy`, `amiet_tools` e `matplotlib`, e carregar as variáveis experimentais a partir dos arquivos de texto `DARP2016_TestSetup.txt` e `DARP2016_AirfoilGeom.txt`. Após carregar as variáveis, estas são exportadas para o "namespace" atual.

```
import numpy as np
import amiet_tools as AmT

import matplotlib.pyplot as plt

# %% *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
# carrega as condicoes do experimento a partir de um arquivo de texto
DARP2016Setup = AmT.loadTestSetup('DARP2016_TestSetup.txt')

# exportar variaveis para o namespace atual
(c0, rho0, p_ref, Ux, turb_intensity, length_scale, z_sl, Mach, beta,
flow_param, dipole_axis) = DARP2016Setup.export_values()

# %% *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
# carrega a geometria do aerofolio a partir de um arquivo de texto
DARP2016Airfoil = AmT.loadAirfoilGeom('DARP2016_AirfoilGeom.txt')

(b, d, Nx, Ny, XYZ_airfoil, dx, dy) = DARP2016Airfoil.export_values()
XYZ_airfoil_calc = XYZ_airfoil.reshape(3, Nx*Ny)
```

Listing 7: Script de teste 1 - início

Aqui também realizamos um passo importante para os cálculos numéricos: redimensionamos o array `XYZ_airfoil` de coordenadas na superfície do aerofólio para criarmos um array `XYZ_airfoil_calc` de dimensões $(3, N_x N_y)$, que será útil para a realização de alguns cálculos posteriormente.

Em seguida, definimos a frequência temporal de interesse, conforme indicado na Listagem 8. Para este exemplo, iniciamos o cálculo com um valor de frequência não dimensional normalizada pela corda $k_0 c = 5$, o que resulta em uma frequência temporal $f \approx 1.8$ kHz. Por conveniência, exportamos as variáveis mais uma vez para o "namespace" atual.

```
# frequencia de operacao
kc = 5                          # freq normalizada pela corda = k0*(2*b)
f0 = kc*c0/(2*np.pi*(2*b))      # approx 1.8 kHz

FreqVars = AmT.FrequencyVars(f0, DARP2016Setup)
(k0, Kx, Ky_crit) = FreqVars.export_values()
```

Listing 8: Script de teste 1 - definindo a frequência

Para fins ilustrativos, escolhemos uma amplitude de rajada $\hat{w}_0 = 1$. Em seguida, escolhemos um valor para o número de onda lateral de rajada k_ψ (aqui escolhemos $k_\psi = 0.35 k_\psi^{crit}$ para fins ilustrativos), e calculamos a flutuação de pressão Δp sobre o aerofólio em resposta a esta rajada usando a função `AmT.delta_p`, conforme indicado na Listagem 9. O resultado é armazenada na variável `delta_p1`, um array `numpy` de dimensões `(Ny, Nx)` contendo os valores do salto de pressão Δp em cada ponto na superfície do aerofólio. Note que, como a função Δp não depende da altura do aerofólio z_s, passamos apenas as primeiras duas coordenadas (x_s e y_s) do array `XYZ_airfoil` como argumento para a função `AmT.delta_p`.

Ainda na Listagem 9, realizamos mais um passo importante para os cálculos numéricos: redimensionamos o array `delta_p1` para criar um array `delta_p1_calc` de dimensão $(N_x N_y,)$, e multiplicamos este array pelos intervalos de amostragem `dx` (um array) e `dy` (um número). Desta forma, cada elemento do array redimensionado `delta_p1_calc` conterá o valor da flutuação de pressão Δp calculada para o elemento localizado em (x_s, y_s), e ponderada pela área do elemento $\mathrm{d}x_s\,\mathrm{d}y_s$.

```
# amplitude da rajada
w0 = 1

# numero de onda lateral da rajada
ky = 0.35*ky_crit

# Calcula o salto de pressao sobre o aerofolio
delta_p1 = AmT.delta_p(rho0, b, w0, Ux, Kx, ky, XYZ_airfoil[0:2], Mach)

# redimensiona o array de salto de pressao, aplica pesos por area da amostragem
delta_p1_calc = (delta_p1*dx).reshape(Nx*Ny)*dy
```

Listing 9: Script de teste 1 - calculando o salto de pressão

A Figura 4 mostra a parte real $\mathrm{Re}\{\Delta p\}$ e a magnitude $|\Delta p|$ (em decibéis) do salto de pressão (normalizado) Δp obtido com os comandos listados acima. Note como a flutuação de pressão apresenta maior magnitude próximo ao bordo de ataque, e tende a zero no bordo de fuga. Note ainda a oscilação na direção lateral, devido ao número de onda lateral da rajada $k_\psi \neq 0$.

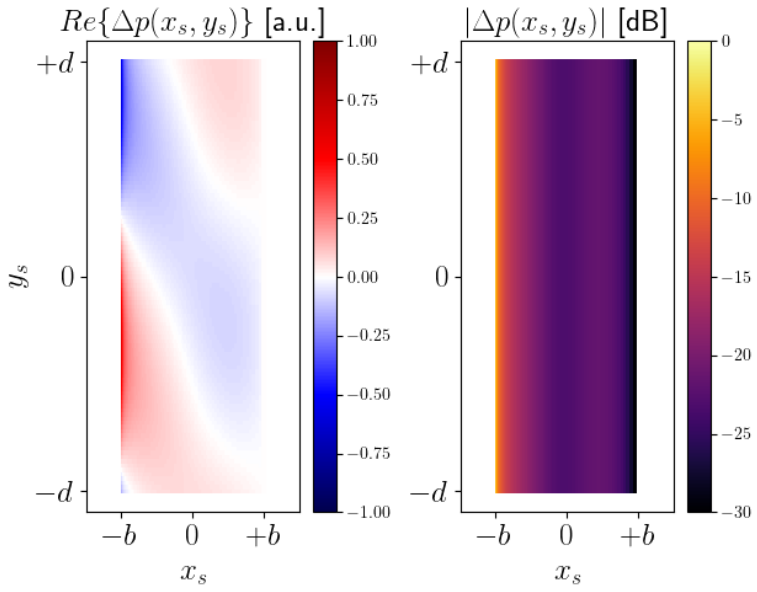


Figura 4: Parte real $\mathrm{Re}(\Delta p)$ e a magnitude $|\Delta p|$ (em decibéis) do salto de pressão (normalizado) Δp obtido com $k_\psi = 0.35 k_\psi^{crit}$.

Em seguida, calculamos as funções de diretividade do campo acústico nas direções da corda

(plano $y = 0$) e da envergadura (plano $x = 0$). Para isso, criamos uma série de pontos a uma distância `R_farfield` do centro do aerofólio, arranjados de forma semicircular, com espaçamento angular uniforme, e calculamos o campo acústico visto por esses observadores. Estes comandos estão indicados na Listagem 10, onde usamos $M = 181$ observadores em cada direção, e os arrays resultantes `XZ_farfield` e `YZ_farfield` possuem dimensões $(3, M)$.

Dadas os arrays `XZ_farfield` e `YZ_farfield` de coordenadas dos observadores distantes nas direções da corda e da envergadura, respectivamente, calculamos a matriz de funções de transferência para cada par fonte-observador através da função `AmT.dipole3D`. As matrizes resultantes `G_ffXZ` e `G_ffYZ` são arrays `numpy` de dimensões $(M, N_x N_y)$.

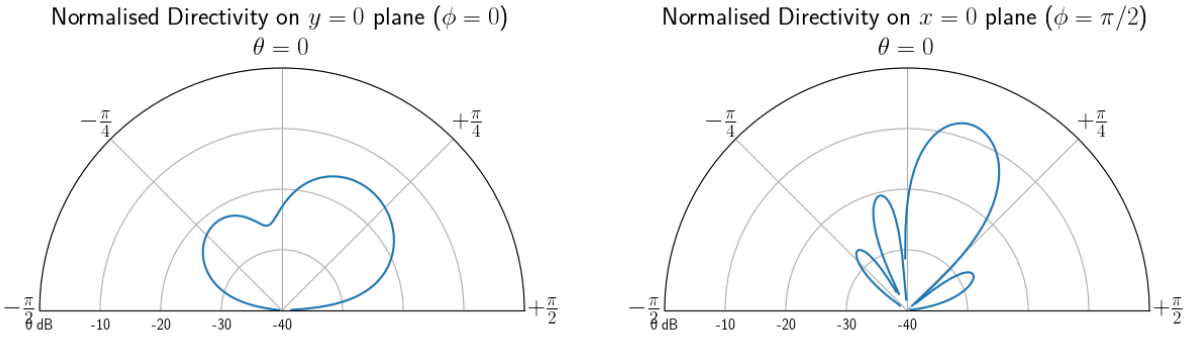
A pressão acústica p vista por cada observador é então calculada através do produto matricial `G_ffXZ @ delta_p1_calc`, onde o símbolo `@` é usado no pacote `numpy` para indicar produto escalar entre dois arrays. Note que como a variável `delta_p1_calc` já inclui uma ponderação pela área de cada elemento $\mathrm{d}x_s\,\mathrm{d}y_s$ da superfície do aerofólio, a operação aqui descrita pode ser entendida como o cálculo numérico da integral de superfície definida na Eq. 3:

$$
\begin{aligned}
p(\mathbf{r}, \omega) &= \int_{-d}^{+d} \int_{-b}^{+b} \Delta p(\mathbf{r}_s, \omega) \frac{\partial}{\partial z_s} G_{U_x}(\mathbf{r}|\mathbf{r}_s, \omega)\,\mathrm{d}x_s\,\mathrm{d}y_s \\
&\approx \sum_{N_x} \sum_{N_y} \frac{\partial}{\partial z_s} G_{U_x}(\mathbf{r}|\mathbf{r}_s, \omega)\;\left[\Delta p(\mathbf{r}_s, \omega)\,\mathrm{d}x_s\,\mathrm{d}y_s\right].
\end{aligned}
\tag{12}
$$

```
# Cria 181 observadores no campo distante
R_farfield = 50     # [m]
theta_farfield = np.linspace(-np.pi/2, np.pi/2, 181)
xy_farfield = R_farfield*np.sin(theta_farfield)
z_farfield = -R_farfield*np.cos(theta_farfield)

XZ_farfield = np.array([xy_farfield, np.zeros(xy_farfield.shape), z_farfield])
YZ_farfield = np.array([np.zeros(xy_farfield.shape), xy_farfield, z_farfield])

# Campo acustico gerado pelo aerofolio nos observadores
p_XZ_farfield = np.zeros(x_farfield.shape, 'complex')
p_YZ_farfield = np.zeros(x_farfield.shape, 'complex')

# *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
# Calcula matrizes de funcao de transferencia para dipolos com conveccao entre
# cada fonte e cada observador
G_ffXZ = AmT.dipole3D(XYZ_airfoil_calc, XZ_farfield, k0, dipole_axis,
flow_param)

G_ffYZ = AmT.dipole3D(XYZ_airfoil_calc, YZ_farfield, k0, dipole_axis,
flow_param)

# Calcula a pressao acustica no campo distante
p_XZ_farfield = G_ffXZ @ delta_p1_calc
p_YZ_farfield = G_ffYZ @ delta_p1_calc
```

Listing 10: Script de teste 1 - observadores no campo distante

As funções de diretividade da radiação acústica nas direções da corda e da envergadura estão demonstradas na Figura 5. Note o caráter “inclinado” da diretividade na direção da envergadura, característica típica da radiação em resposta a rajadas com $k_\psi \neq 0$.



(a) (b)

Figura 5: Funções de diretividade da radiação acústica em resposta a uma rajada única, com $k_0 c = 5$, $M_x \approx 0.17$. (a) diretividade na direção da corda; (b) diretividade na direção da envergadura.

4 Pseudocódigo

O pseudocódigo usado para calcular o espectro cruzado $S_{\Delta p \Delta p'}$ da pressão de superfície do aerofólio e/ou o espectro cruzado $S_{pp'}$ do ruído radiado é mostrado abaixo, retirado de [2]. A maior parte das instruções mostradas abaixo possuem funções Python equivalentes ou métodos associados no pacote `AmT`, e não precisam ser implementadas pelo usuário final. Para mais detalhes sobre o uso destas funções, veja a documentação e os scripts de exemplo no pacote.

Algorithm 1 Pseudocódigo para o modelo de predição de ruído de interação turbulência-aerofólio do pacote `amiet_tools`

Definir as variáveis do ambiente de teste: $c_0, \rho_0, p_{ref}, U_x, \overline{w^2}, \Lambda, z_{sl}, M_x, \beta$
Definir as variáveis da geometria do aerofólio: $b, d, N_x, N_y, \Delta x_s, \Delta y_s$
Calcular a amostragem da corda do aerofólio $x_s[n],\ n \in [1, \ldots, N_x]$
Calcular a amostragem da envergadura do aerofólio $y_s[n],\ n \in [1, \ldots, N_y]$
Definir as coordenadas dos observadores $\mathbf{r}_m = (x_m, y_m, z_m),\ m \in [1, \ldots, M]$
if usando correção de camada cisalhante **then**
    for all pares (m, n) de pontos observador-aerofólio $(\mathbf{r}_m|\mathbf{r}_s)$ **do**
        Calcular os pontos de cruzamento da camada cisalhante $\mathbf{r}_l(m, n)$ e tempos de propagação $\tau_{sm}(m, n)$
    end for
end if
for cada frequência $f \in [f_{min}, \ldots, f_{max}]$ **do**
    Calcular $k_0, \kappa_\chi, k_\psi^{crit}$
    Calcular valores de números de onda de rajada $k_\psi \in [-k_\psi^{max}, k_\psi^{max}]$
    Calcular intervalo de amostragem dos números de onda de rajada Δk_ψ
    Calcular espectro de energia dos números de onda de rajada $\Phi_{ww}(k_\psi)$
    Calcular matriz $\partial G(\mathbf{r}_m|\mathbf{r}_s)/\partial z_s$ para todos os pares (m, n) de pontos observador-aerofólio;
    Inicializar $S_{\Delta p \Delta p'} \leftarrow 0$
    Inicializar $S_{pp'} \leftarrow 0$
    for cada rajada $k_\psi \in [-k_\psi^{max}, k_\psi^{max}]$ **do**
        $w(k_\psi) \leftarrow \sqrt{\Phi_{ww}(k_\psi)}$
        Calcular $\Delta p(x_s, y_s, k_\psi)$ usando $w(k_\psi)$
        $S_{\Delta p \Delta p'}(\mathbf{r}_s, \mathbf{r}'_s) \leftarrow S_{\Delta p \Delta p'}(\mathbf{r}_s, \mathbf{r}'_s) + [\Delta p(x_s, y_s, k_\psi) \cdot \Delta p^*(x'_s, y'_s, k_\psi)] \cdot U_x \cdot \Delta k_\psi$
    end for
    $S_{pp'}(\mathbf{r}_m, \mathbf{r}_{m'}) \leftarrow 4\pi \left[S_{\Delta p \Delta p'} \cdot (\Delta x_s \cdot \Delta y_s) \cdot (\Delta x'_s \cdot \Delta y'_s)\right] \cdot [\partial G(\mathbf{r}_m|\mathbf{r}_s)/\partial z_s] \cdot [\partial G(\mathbf{r}_{m'}|\mathbf{r}'_s)/\partial z_s]$
end for

5 Descrição das classes

O pacote `amiet_tools` utiliza classes — i.e. estruturas de abstração usadas em programação orientada a objetos — para armazenar dados relacionados aos diferentes aspectos da simulação. As três classes utilizadas em`AmT` estão descritas abaixo, seguidas de uma lista de seus atributos:

- Classe `TestSetup`:
  - `TestSetup.c0`: velocidade do som em um meio em repouso c_0 (em m/s);
  - `TestSetup.rho0`: densidade do ar ρ_0 (em kg/m^3);
  - `TestSetup.p_ref`: pressão acústica de referência $p_{ref} = 20\mu$Pa RMS;
  - `TestSetup.Ux`: velocidade média do escoamento U_x (em m/s);
  - `TestSetup.turb_intensity`: intensidade da turbulência $\sqrt{\overline{w^2}}/U_x$;
  - `TestSetup.length_scale`: escala de comprimento de turbulência Λ (em m);
  - `TestSetup.z_sl`: altura da camada cisalhante z_{sl} (em m);
  - `TestSetup.flow_dir`: caractere único indicando a direção do escoamento (i.e. `'x'` para indicar escoamento na direção $+x$);
  - `TestSetup.dipole_axis`: caractere único indicando a direção do eixo dos dipolos (i.e. '`z`' para dipolos apontando "para cima", na direção $+z$);
  - `TestSetup.Mach`: número de Mach do escoamento $M_x = U_x/c_0$;
  - `TestSetup.beta`: fator de Prandtl-Glauert $\beta = \sqrt{1 - M_x^2}$.
- Classe `AirfoilGeom`:
  - `AirfoilGeom.b`: semi corda do aerofólio $b = c/2$ (em m);
  - `AirfoilGeom.d`: semi envergadura do aerofólio $d = L/2$ (em m);
  - `AirfoilGeom.Nx`: número de amostras sobre a corda N_x, amostragem não-uniforme;
  - `AirfoilGeom.Ny`: número de amostras sobre a envergadura N_y, amostragem uniforme;
  - `AirfoilGeom.dx`: intervalos de amostragem sobre a corda Δx_s (calculado internamente)
  - `AirfoilGeom.dy`: intervalo de amostragem sobre a envergadura Δy_s (calculado internamente).
- Classe `FrequencyVariables`:
  - `FrequencyVariables.freq`: frequência f (em Hz);
  - `FrequencyVariables.k0`: número de onda acústico $k_0 = 2\pi f/c_0$;
  - `FrequencyVariables.Kx`: número de onda de rajada na direção da corda $\kappa_\chi = \omega/U_x$;
  - `FrequencyVariables.Ky_crit`: número de onda crítico de rajada na direção da envergadura k_ψ^{crit}.

Uma instância de cada classe é criada para armazenar os valores das variáveis relacionados às condições de um experimento e para passá-las às diferentes funções no pacote. Valores numéricos podem ser atribuídos diretamente através de um script em Python, ou podem ser lidos em um arquivo externo através das funções `loadTestSetup` e `loadAirfoilGeom`.

Uma quarta classe chamada `MicArrayCsmHDF5` também está inclusa no pacote, e permite a leitura e escrita de dados de CSM de arranjos de microfones no formato HDF5 adotado pela comunidade "Array Methods" [9, 10] para compartilhar dados de "beamforming".

Referências

[1] R. K. Amiet. Acoustic radiation from an airfoil in a turbulent stream. *Journal of Sound and Vibration*, 41, No. 4:407–420, 1975.

[2] Fabio Casagrande Hirono, Phillip Joseph, and Filippo Fazi. An open-source implementation of analytical turbulence-airfoil interaction noise model. In *26th AIAA/CEAS Aeroacoustics Conference*, 2020. URL https://doi.org/10.2514/6.2020-2544. AIAA Paper 2020-2544.

[3] Gabriel Reboul. *Modélisation du bruit à large bande de soufflante de turboréacteur*. PhD Thesis, Laboratoire de Mécanique des Fluides et d'Acoustique - École Centrale de Lyon, Lyon - France, 2010.

[4] Michel Roger. Broadband noise from lifting surfaces: Analytical modeling and experimental validation. In Roberto Camussi, editor, *Noise Sources in Turbulent Shear Flows: Fundamentals and Applications*. Springer-Verlag Wien, 2013.

[5] Leandro de Santana. *Semi-analytical methodologies for airfoil noise prediction*. PhD Thesis, Faculty of Engineering Science - Katholieke Universiteit Leuven, Leuven - Belgium, 2015.

[6] Fabio Casagrande Hirono. *Far-Field Microphone Array Techniques for Acoustic Characterisation of Aerofoils*. PhD Thesis, Institute of Sound and Vibration Research - University of Southampton, Southampton - UK, 2018.

[7] C. J. Chapman. Similarity variables for sound radiation in a uniform flow. *Journal of Sound and Vibration*, 233, No. 1:157–164, 2000.

[8] Chaitanya Paruchuri. *Aerofoil geometry effects on turbulence interaction noise*. PhD Thesis, Institute of Sound and Vibration Research - University of Southampton, Southampton - UK, 2017.

[9] Christopher Bahr, William Humphreys Jr., Daniel Ernst, Thomas Ahlefeldt, Carsten Spehr, Antonio Pereira, Quentin Leclere, Christophe Picard, Ric Porteous, Danielle Moreaux, Jeoffrey Fischer, and Con Doolan. A comparison of microphone phased array methods applied to the study of airframe noise in wind tunnel testing. In *23rd AIAA/CEAS Aeroacoustics Conference*, 2017. AIAA Paper AIAA 2017-3718.

[10] Ennes Sarradj, Gert Herold, Pieter Sijtsma, Roberto Merino-Martinez, Anwar Malgoezar, Mirjam Snellen, Thomas Geyer, Christopher Bahr, Ric Porteous, Danielle Moreau, and Con Doolan. A microphone array method benchmarking exercise using synthesized input data. In *23rd AIAA/CEAS Aeroacoustics Conference*, 2017. AIAA Paper AIAA 2017-3719.