Openwind: a software to simulate wind instrument, as a tool for acoustic teachers

Augustin Ernoult(♯), Jérémy Cabaret(♭), Juliette Chabassier(♯)

♯) team Makutu, Inria, Bordeaux University
♭) Itemm, Le Mans

Itemm
Institut technologique européen des métiers de la musique

Inria

Oultine

Motivations

Context :

- No shared tools in the academic community
  - current practice : ± 1 software per team and/or per researcher
- For instrument makers :
  - Physical prototyping : time and money consuming
  - CAD tools ? → out-of-date

  out-of-date
- A team with :
  - acousticians
  - numerical analysts
  - strong contact with instrument makers

Motivations

Context :

- No shared tools in the academic community
  - current practice : ± 1 software per team and/or per researcher
- For instrument makers :
  - Physical prototyping : time and money consuming
  - CAD tools ? → out-of-date

  out-of-date
- A team with :
  - acousticians
  - numerical analysts
  - strong contact with instrument makers

Ambition :

- A tool with
  - up-to-date physical models
  - up-to-date numerical methods
- A tool for :
  - researchers
  - wind instrument makers
  - teachers

Motivations

Context :

- No shared tools in the academic community
  - current practice : $\pm$ 1 software per team and/or per researcher
- For instrument makers :
  - Physical prototyping : time and money consuming
  - CAD tools ? $\rightarrow$ out-of-date

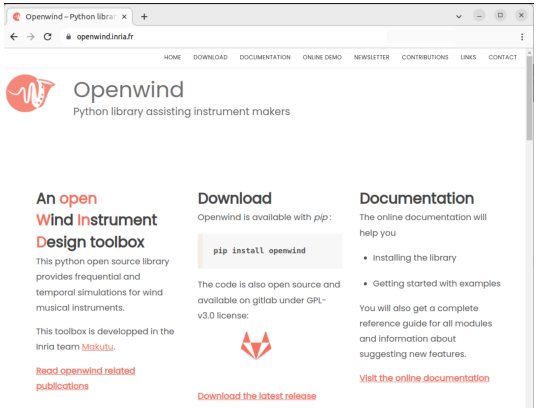
  out-of-date
- A team with :
  - acousticians
  - numerical analysts
  - strong contact with instrument makers

Ambition :

- A tool with
  - up-to-date physical models
  - up-to-date numerical methods
- A tool for :
  - researchers
  - wind instrument makers
  - teachers

Openwind : open Wind Instrument Design

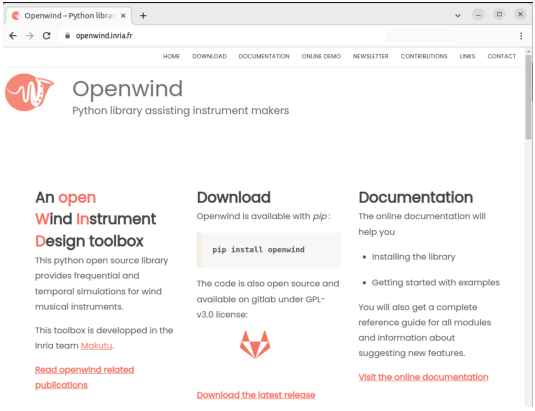
- openwind.inria.fr
- Python library
- Free and open source (GPL v.3.0)
  - a collaborative software ? (not yet)



1. A. Morrison and R. Worland. "*Calculating the acoustic input impedance of a simplified brass instrument as an educational laboratory activity*". JASA 151(6), 2022.

Openwind : open Wind Instrument Design

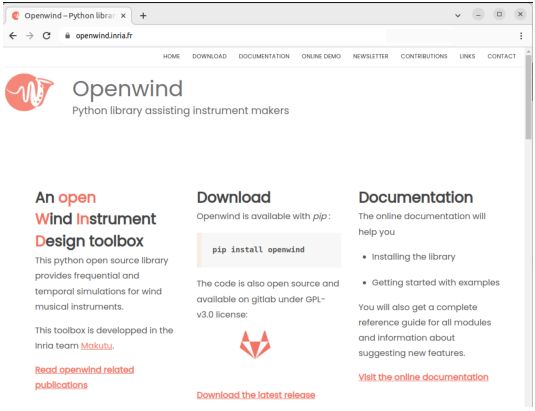
- openwind.inria.fr
- Python library
- Free and open source (GPL v.3.0)
  - a collaborative software ? (not yet)
- Available
  - Gitlab
  - documentation
  - web-interface



1. A. Morrison and R. Worland. "*Calculating the acoustic input impedance of a simplified brass instrument as an educational laboratory activity*". JASA 151(6), 2022.

Openwind : open Wind Instrument Design

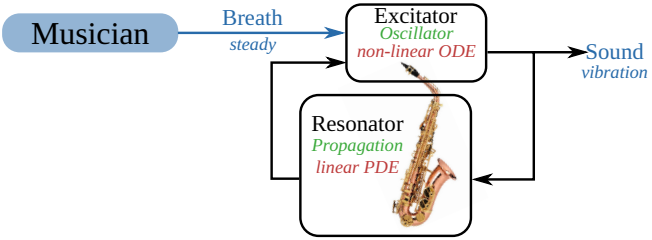
- openwind.inria.fr
- Python library
- Free and open source (GPL v.3.0)
  - a collaborative software ? (not yet)
- Available
  - Gitlab
  - documentation
  - web-interface
- Used mainly for teaching :
  - General education science [1]
  - Instrument makers
  - Students projects (high school, bachelor, master, etc.)



1. A. Morrison and R. Worland. "*Calculating the acoustic input impedance of a simplified brass instrument as an educational laboratory activity*". JASA 151(6), 2022.

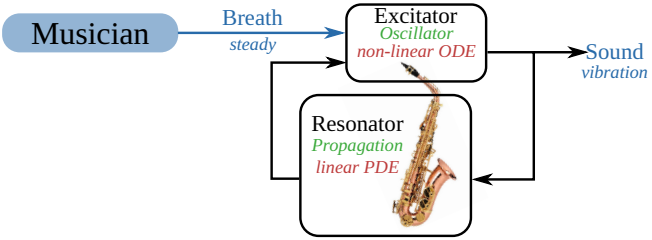
Oultine

The couple musician-instrument



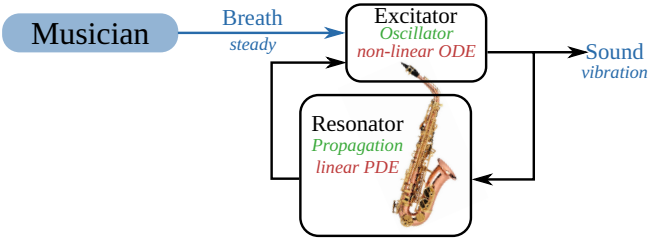
- Musician : steady flow/pressure
- Instrument : Sound = Air Vibration

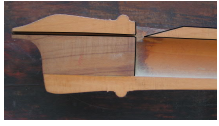
The couple musician-instrument



- Musician : steady flow/pressure
- Instrument : Sound = Air Vibration
- Resonator : propagation of acoustic waves

The couple musician-instrument

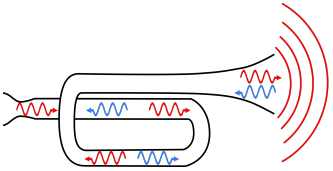




- Musician : steady flow/pressure
- Instrument : Sound = Air Vibration
- Resonator : propagation of acoustic waves
- Excitateur : oscillator
  - Types :
    - Reed(s) (clarinet)
    - Lips (trumpet)
    - Air Jet (flute)

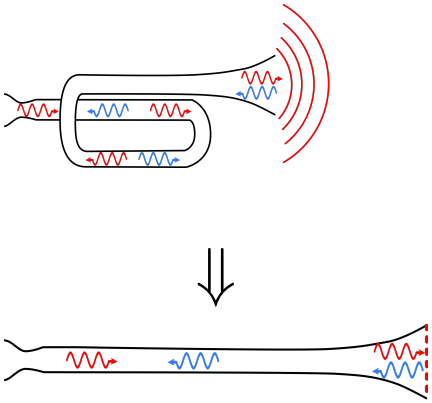
Propagation along the air column

- Acoustic propagation
  - 3D
  - inside the instrument and in the room
  - too heavy to model



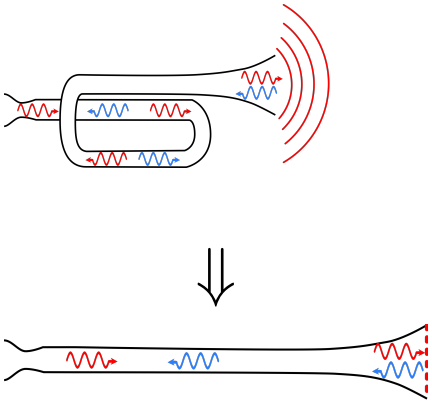
Propagation along the air column

- Acoustic propagation
  - 3D
  - inside the instrument and in the room
  - too heavy to model
- Hypothesies
  - radiation = boundary condition
  - bend : no influence
  - axial symmetry



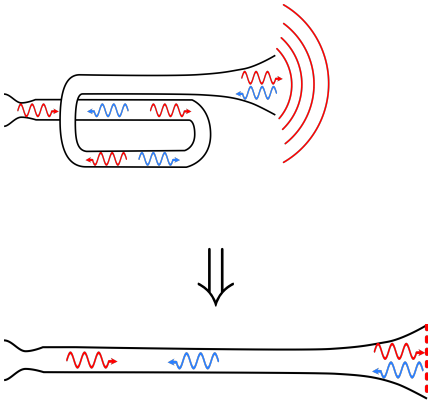
Propagation along the air column

- Acoustic propagation
  - 3D
  - inside the instrument and in the room
  - too heavy to model
- Hypothesies
  - radiation = boundary condition
  - bend : no influence
  - axial symmetry
- Propagation 1D
  - only the air column
  - the "bore profile" : $R(x)$

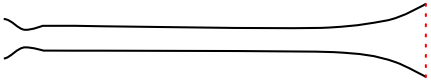


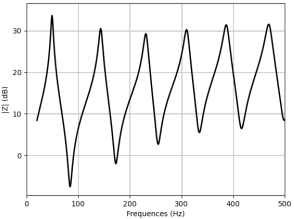
Propagation along the air column

- Acoustic propagation
  - 3D
  - inside the instrument and in the room
  - too heavy to model
- Hypothesies
  - radiation = boundary condition
  - bend : no influence
  - axial symmetry
- Propagation 1D
  - only the air column
  - the "bore profile" : $R(x)$
- Holes
  - junction of pipes (T-joint)
  - with acoustic masses



Equations





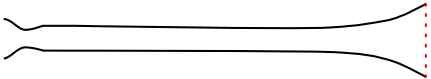
- Transmission line equation :

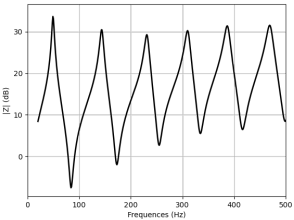
$$\begin{cases} \dfrac{j\omega\rho}{S}\left[1-\mathcal{J}(k_v(\omega)R)\right]^{-1}\hat{u}-\dfrac{d\hat{p}}{dx}=0, & k_v(\omega)=\sqrt{j\omega\dfrac{\rho}{\mu}} \\ \dfrac{j\omega S}{\rho c^2}\left[1+(\gamma-1)\mathcal{J}(k_t(\omega)R)\right]\hat{p}-\dfrac{d\hat{u}}{dx}=0, & k_t(\omega)=\sqrt{j\omega\rho\dfrac{C_p}{\kappa}} \end{cases}$$

  - Wall losses : thermo (κ) + viscous (μ)

- Input impedance : $Z(\omega) = p(\omega)/u(\omega)$
  - Ratio : (response)/(source)
  - Quantify the ease of vibration of the air column

Equations





- Transmission line equation :

$$\begin{cases} \dfrac{j\omega\rho}{S}\left[1-\mathcal{J}(k_v(\omega)R)\right]^{-1}\hat{u}-\dfrac{d\hat{p}}{dx}=0, & k_v(\omega)=\sqrt{j\omega\dfrac{\rho}{\mu}} \\ \dfrac{j\omega S}{\rho c^2}\left[1+(\gamma-1)\mathcal{J}(k_t(\omega)R)\right]\hat{p}-\dfrac{d\hat{u}}{dx}=0, & k_t(\omega)=\sqrt{j\omega\rho\dfrac{C_p}{\kappa}} \end{cases}$$

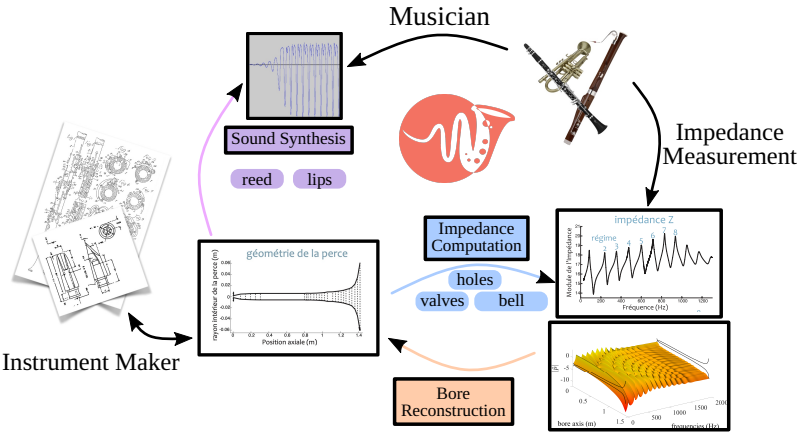
  - Wall losses : thermo (κ) + viscous (μ)

- Input impedance : $Z(\omega) = p(\omega)/u(\omega)$
  - Ratio : (response)/(source)
  - Quantify the ease of vibration of the air column

- Reed/Lips
  - 1 d.o.f. mass-spring oscillator + Bernoulli
  - 5 "control" parameters

Oultine

Features


Musician
Sound Synthesis
reed
lips
Impedance
Measurement
Impedance
Computation
holes
valves
bell
Bore
Reconstruction
Instrument Maker
géométrie de la perce
Position axiale (m)
impédance Z
régime
Fréquence (Hz)
bore axis (m)
frequencies (Hz)

Impedance computation

- Example : cone impedance at 20°C

```
import numpy as np
from openwind import ImpedanceComputation

freq = np.arange(20, 2000, 1)
geometry = [[0, 0.5, 4e-3, 10e-3, 'cone']]
#or geometry = filename.txt

result = ImpedanceComputation(freq, geometry,
                              temperature=20)
result.plot_impedance()
Z = result.impedance
```

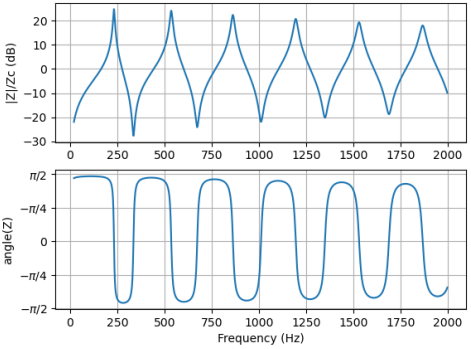
Impedance computation

- Example : cone impedance at 20°C

```
import numpy as np
from openwind import ImpedanceComputation

freq = np.arange(20, 2000, 1)
geometry = [[0, 0.5, 4e-3, 10e-3, 'cone']]
#or geometry = filename.txt

result = ImpedanceComputation(freq, geometry,
                              temperature=20)
result.plot_impedance()
Z = result.impedance
```



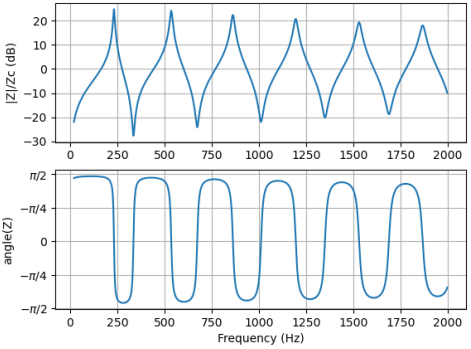
Impedance computation

- Example : cone impedance at 20°C

```
import numpy as np
from openwind import ImpedanceComputation

freq = np.arange(20, 2000, 1)
geometry = [[0, 0.5, 4e-3, 10e-3, 'cone']]
#or geometry = filename.txt

result = ImpedanceComputation(freq, geometry,
                              temperature=20)
result.plot_impedance()
Z = result.impedance
```



- Options :
  - Computation method (FEM, TMM, modal, mesh)
  - Physic quantities (T°, RH, CO2, ...)
  - Models (losses, radiation, wavefront, ...)

Sound synthesis

- Let's make sound

```
from openwind import simulate, Player
from openwind.temporal.utils import export_mono

duration = 1.0 # seconds
geometry = [[0, 0.75, 4e-3, 10e-3, 'cone']]
embouchure = Player('WOODWIND_REED_SCALED')

result = simulate(duration, geometry,
                  player=embouchure)

sound = result.values['bell_radiation_flow']
export_mono('my_sound.wav', sound, result.ts)
```

Sound synthesis

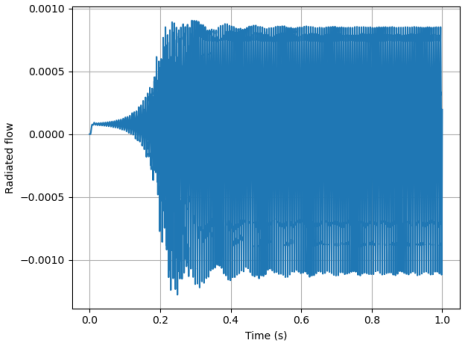
- Let's make sound

```
from openwind import simulate, Player
from openwind.temporal.utils import export_mono

duration = 1.0 # seconds
geometry = [[0, 0.75, 4e-3, 10e-3, 'cone']]
embouchure = Player('WOODWIND_REED_SCALED')

result = simulate(duration, geometry,
                  player=embouchure)

sound = result.values['bell_radiation_flow']
export_mono('my_sound.wav', sound, result.ts)
```



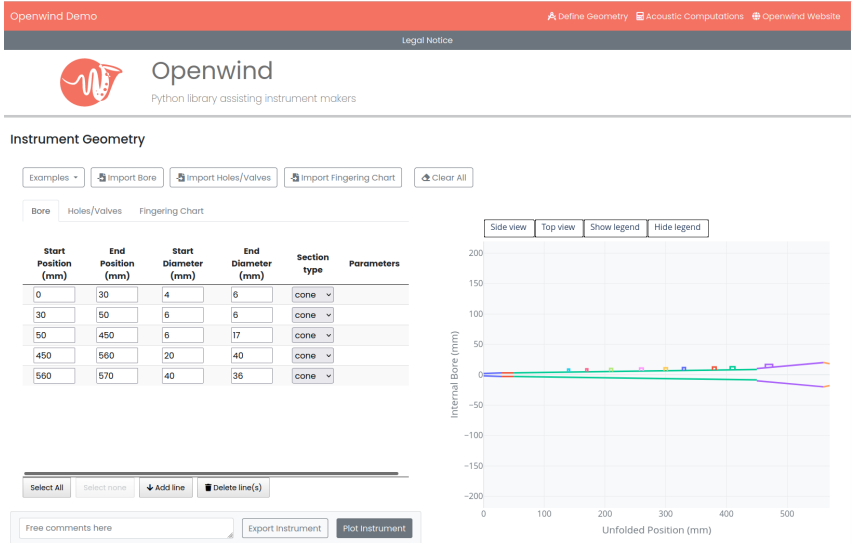
Other features

Oultine

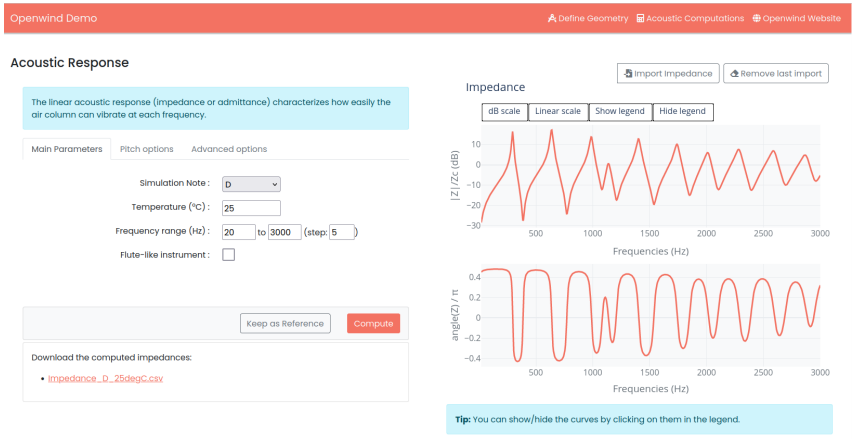
Browser interface

- https://demo-openwind.inria.fr
- Free, without account
  - Save your geometries and results locally
- Manual/tutorial not yet available

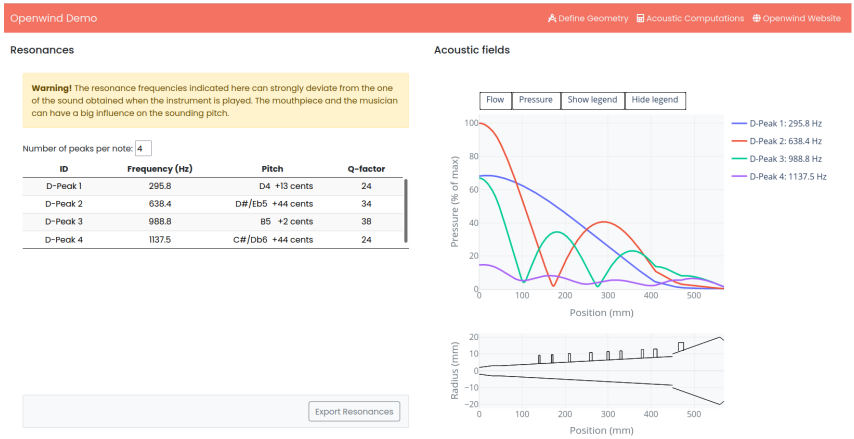
Demonstration : geometry and fingering chart



Demonstration : frequency domain computation



Demonstration : resonances and ac. fields



Oultine

ITEMM (Le Mans)

- School for instrument makers, sound technicians and sellers
  - Piano
  - Guitar
  - **Wind instruments**
- Student profile
  - A-level (heterogeneous)
  - scientific background : (impedance, mode shape, resonances, ...)

ITEMM (Le Mans)

- School for instrument makers, sound technicians and sellers
  - Piano
  - Guitar
  - **Wind instruments**
- Student profile
  - A-level (heterogeneous)
  - scientific background : (impedance, mode shape, resonances, ...)

Activity Context

Virtual prototyping in acoustics courses

- Objectives
  - put into practice the theoretical knowledge
  - illustrate how simulation tools can be included in workshops
  - give a new tool for future practice

Activity 1 : Optimizing the harmonicity of an instrument (brass like)

Objectives

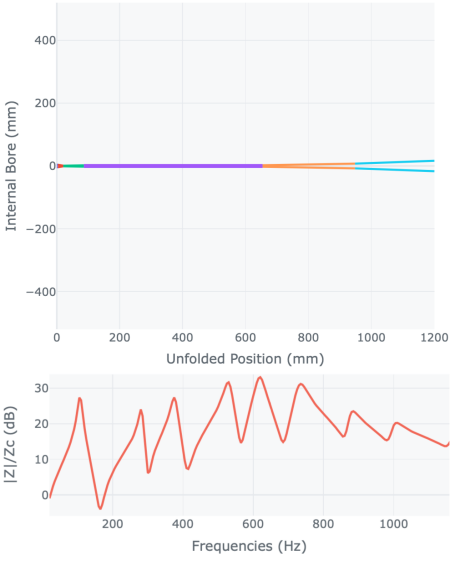
- getting to know the tool
- experimenting the link : geometry-harmonicity
- development of a workflow including simulation tools

Activity 1 : Optimizing the harmonicity of an instrument (brass like)

Objectives

- getting to know the tool
- experimenting the link : geometry-harmonicity
- development of a workflow including simulation tools

- Instructions :
  - Initial geometry : cylindrical and conical parts
  - Modify the geometry to have harmonic resonances ($f_n = nf_1$)

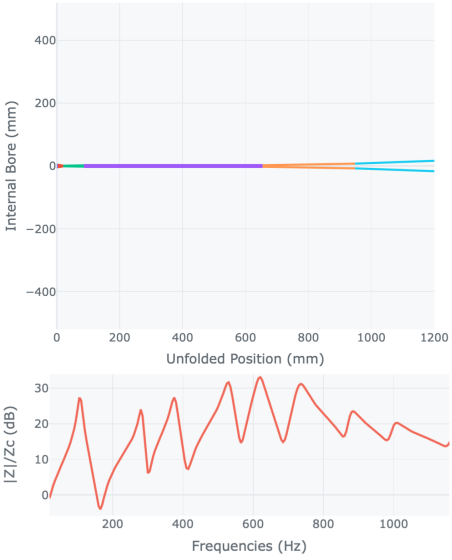


Activity 1 : Optimizing the harmonicity of an instrument (brass like)

Objectives

- getting to know the tool
- experimenting the link : geometry-harmonicity
- development of a workflow including simulation tools

- Instructions :
  - Initial geometry : cylindrical and conical parts
  - Modify the geometry to have harmonic resonances ($f_n = nf_1$)
- Steps :
  - Simulate the impedance
  - Observe the periodic aspect of the resonances
  - Adjust the length of parts and simulate impedance
  - Repeat the steps until harmonization of the resonances



Activity 2 : Tuning a specific resonance (for side holes instruments)

Objectives

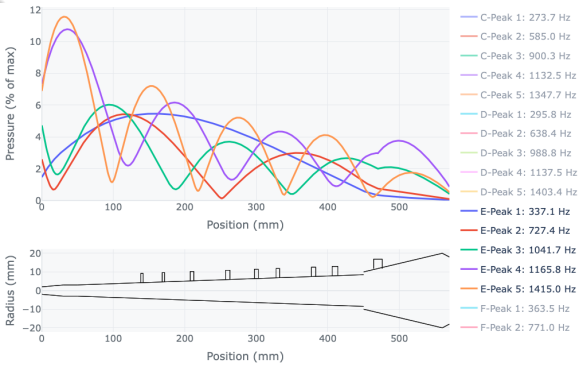
- Using pressure profile to instrument design
- Relationship : geometry, pressure profile, resonances

Activity 2 : Tuning a specific resonance (for side holes instruments)

Objectives

- Using pressure profile to instrument design
- Relationship : geometry, pressure profile, resonances

- Instructions :
  - Tune a resonance without altering the others, by modifying the bore at a specific position

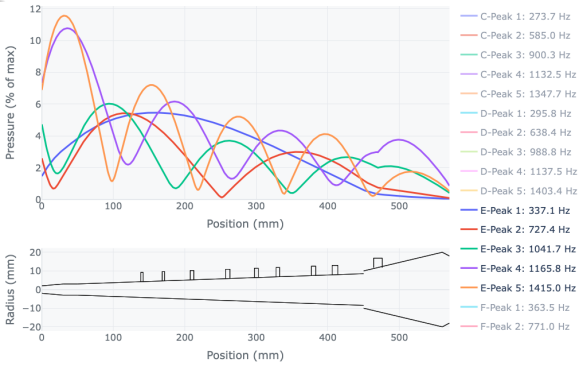


Activity 2 : Tuning a specific resonance (for side holes instruments)

Objectives

- Using pressure profile to instrument design
- Relationship : geometry, pressure profile, resonances

- Instructions :
  - Tune a resonance without altering the others, by modifying the bore at a specific position
- Steps :
  - Simulate the impedance of the instrument
  - Display the pressure profile
  - Analyse the appropriate position for the bore modification
  - modify bore diameter, or hole dimension
  - Try and repeat



Oultine

Conclusion

- GUI in a teaching context
  - Instrument Makers
  - First approach of acoustic (temperature effect, standing waves, resonances, ...)
- Python library
  - Advanced student projects
  - Thesis
- Don't hesitate to contact us :
  - troubles
  - suggestion
  - need help
- Open-source : you can contribute
  - new features
  - share examples
  - ...

Feel free to use it !

https ://openwind.inria.fr