



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

Augustin Ernoult(#), Jérémie Cabaret(þ), Juliette Chabassier(#)

#) team Makutu, Inria, Bordeaux University

þ) Itemm, Le Mans

Itemm

Institut technologique européen
des métiers de la musique

Inria

Outline

① Context

② Physics of wind instruments

③ Openwind

④ Teaching activity

⑤ Conclusion

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 ? \rightarrow out-of-date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 ? \rightarrow out-of-date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 : ± 1 software per team and/or per researcher
- For instrument makers :
 - ▶ Physical prototyping : time and money consuming
 - ▶ CAD tools ? \rightarrow out-of-date
- A team with :
 - ▶ acousticians
 - ▶ numerical analysts
 - ▶ strong contact with instrument makers

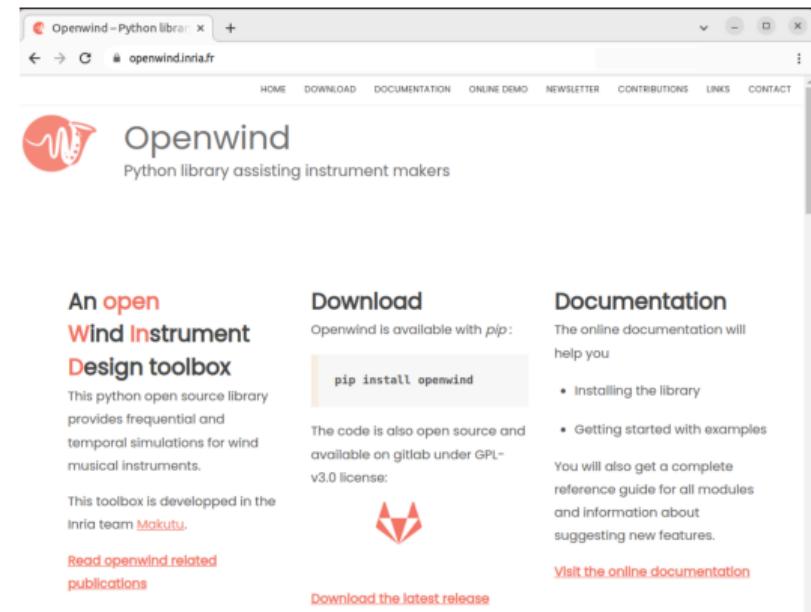
Ambition :

- 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)

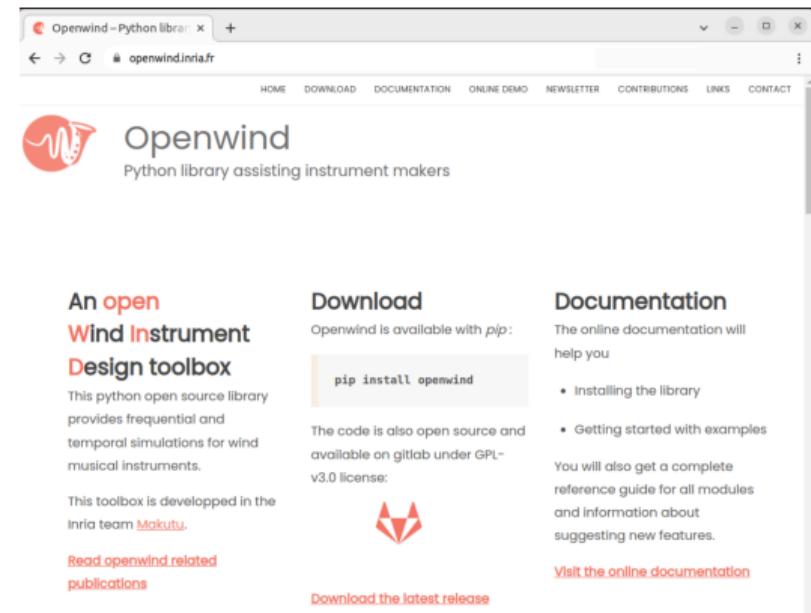


The screenshot shows the homepage of the Openwind website. At the top, there is a navigation bar with links to HOME, DOWNLOAD, DOCUMENTATION, ONLINE DEMO, NEWSLETTER, CONTRIBUTIONS, LINKS, and CONTACT. Below the navigation bar is the Openwind logo, which consists of a red stylized 'W' icon followed by the word 'Openwind' in a serif font. A subtitle below the logo reads 'Python library assisting instrument makers'. The main content area is divided into three columns. The left column is titled 'An open Wind Instrument Design toolbox' and contains a brief description of the library's purpose, mentioning it is developed by the Inria team [Makutu](#). It also links to [related publications](#). The middle column is titled 'Download' and provides instructions for installing the library using pip: `pip install openwind`. It also notes that the code is open source and available on gitlab under GPL-v3.0 license. The right column is titled 'Documentation' and states that the online documentation will help users. It includes links for 'Installing the library', 'Getting started with examples', and 'Visit the online documentation'. There is also a small Inria logo in the bottom right corner of the main content area.

-
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



The screenshot shows the homepage of the Openwind website. At the top, there is a navigation bar with links: HOME, DOWNLOAD, DOCUMENTATION, ONLINE DEMO, NEWSLETTER, CONTRIBUTIONS, LINKS, and CONTACT. Below the navigation bar is the Openwind logo, which consists of a red circular icon containing a white stylized 'W' or wind symbol, followed by the word 'Openwind' in a serif font and the subtitle 'Python library assisting instrument makers' in a smaller sans-serif font.

An open Wind Instrument Design toolbox

This python open source library provides frequential and temporal simulations for wind musical instruments.

This toolbox is developped in the Inria team [Makutu](#).

[Read openwind related publications](#)

Download

Openwind is available with pip:

```
pip install openwind
```

The code is also open source and available on gitlab under GPL-v3.0 license:

[!\[\]\(380e7c6d831ff54f822215b541f79d41_img.jpg\) Download the latest release](#)

Documentation

The online documentation will help you

- Installing the library
- Getting started with examples

You will also get a complete reference guide for all modules and information about suggesting new features.

[Visit the online documentation](#)

-
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.)

The screenshot shows the Openwind website's homepage. At the top, there is a navigation bar with links to HOME, DOWNLOAD, DOCUMENTATION, ONLINE DEMO, NEWSLETTER, CONTRIBUTIONS, LINKS, and CONTACT. Below the navigation bar, there is a logo consisting of a red circle with a white wavy line inside, followed by the word "Openwind" in a serif font, and the tagline "Python library assisting instrument makers" below it.

An open Wind Instrument Design toolbox

This python open source library provides frequential and temporal simulations for wind musical instruments.

This toolbox is developped in the Inria team [Makutu](#).

[Read openwind related publications](#)

Download

Openwind is available with pip:

```
pip install openwind
```

The code is also open source and available on gitlab under GPL-v3.0 license:

[Download the latest release](#)

Documentation

The online documentation will help you

- Installing the library
- Getting started with examples

You will also get a complete reference guide for all modules and information about suggesting new features.

[Visit the online documentation](#)

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.

Outline

1 Context

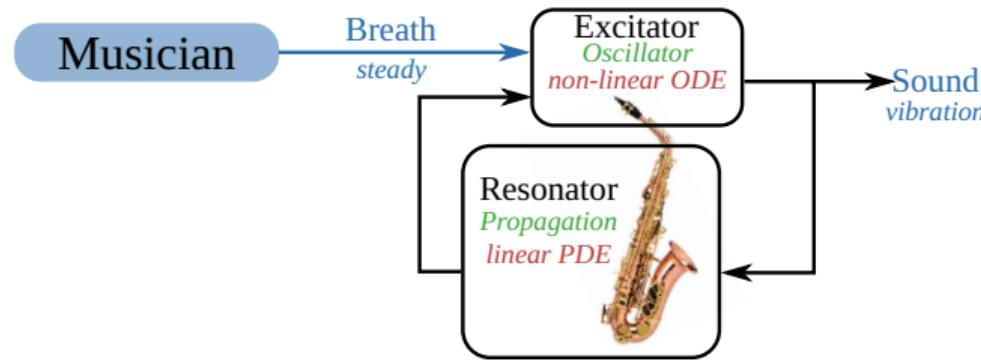
2 Physics of wind instruments

3 Openwind

4 Teaching activity

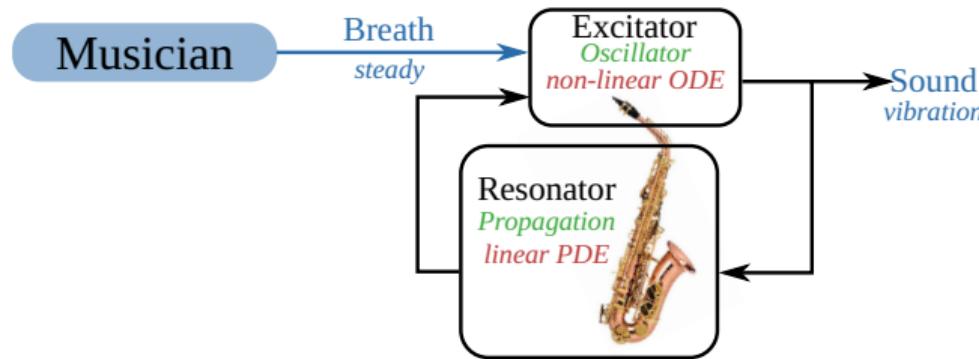
5 Conclusion

The couple musician-instrument



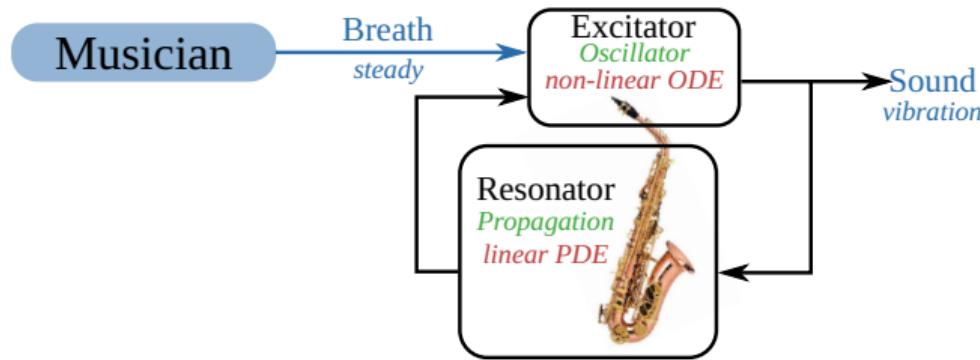
- Musician : steady flow/pressure
- Instrument : Sound = Air Vibration

The couple musician-instrument



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

The couple musician-instrument

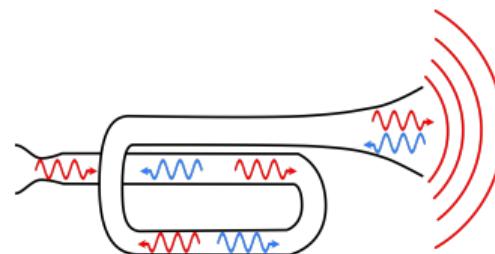


- Musician : steady flow/pressure
- Instrument : Sound = Air Vibration
- Resonator : propagation of acoustic waves
- Excitateur : oscillator
 - ▶ Types :
 - Reed(s) (clarinet)
 - Lips (trumpet)
 - Air Jet (flute)



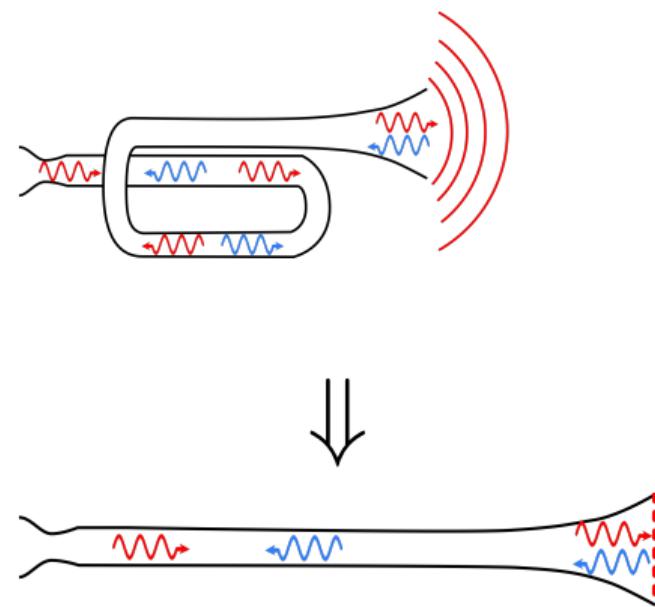
Propagation along the air column

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



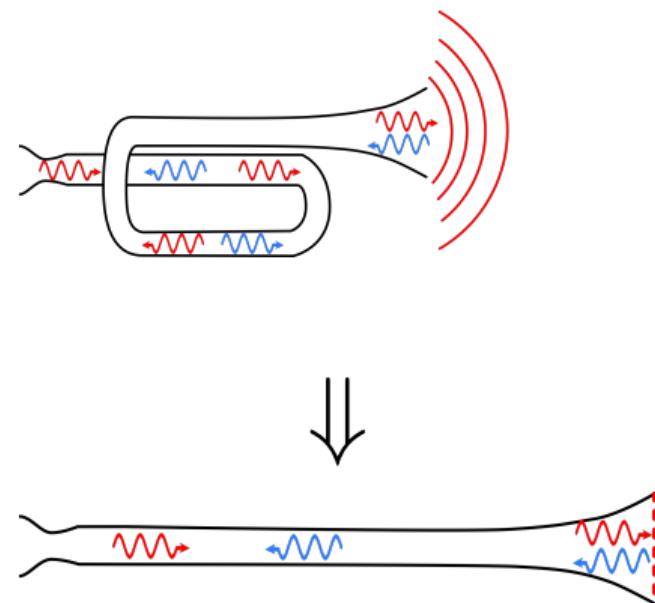
Propagation along the air column

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



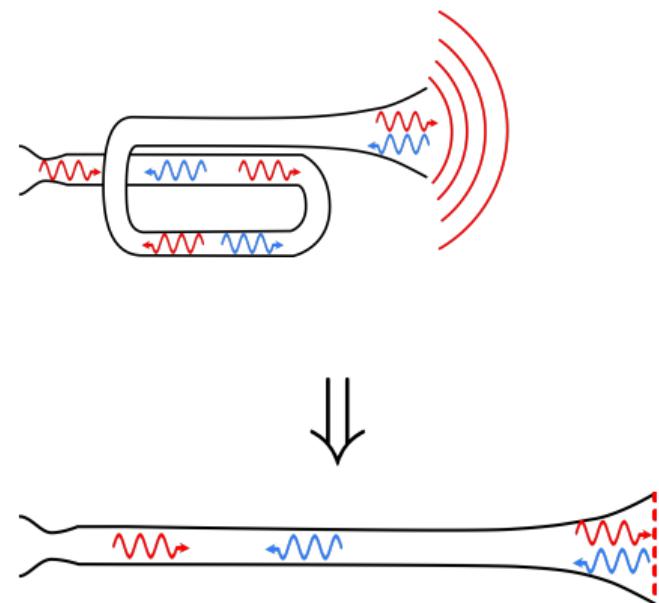
Propagation along the air column

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

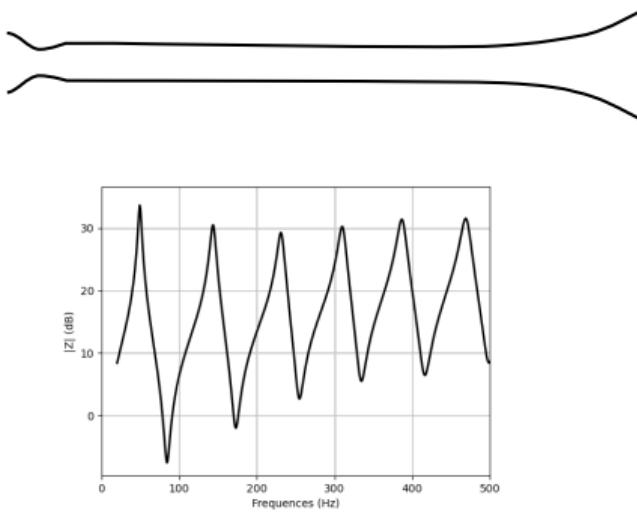


Propagation along the air column

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

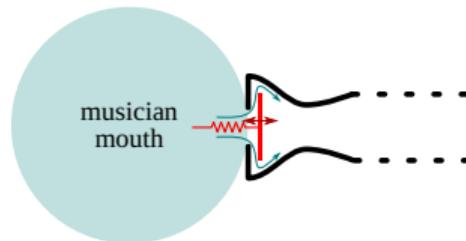
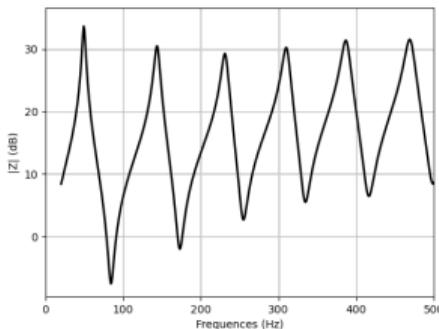
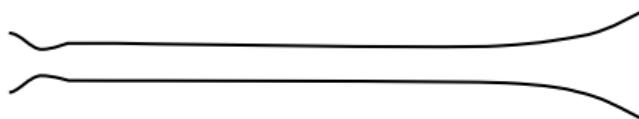


- Transmission line equation :

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

- ▶ 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 :

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

- ▶ 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

Outline

1 Context

2 Physics of wind instruments

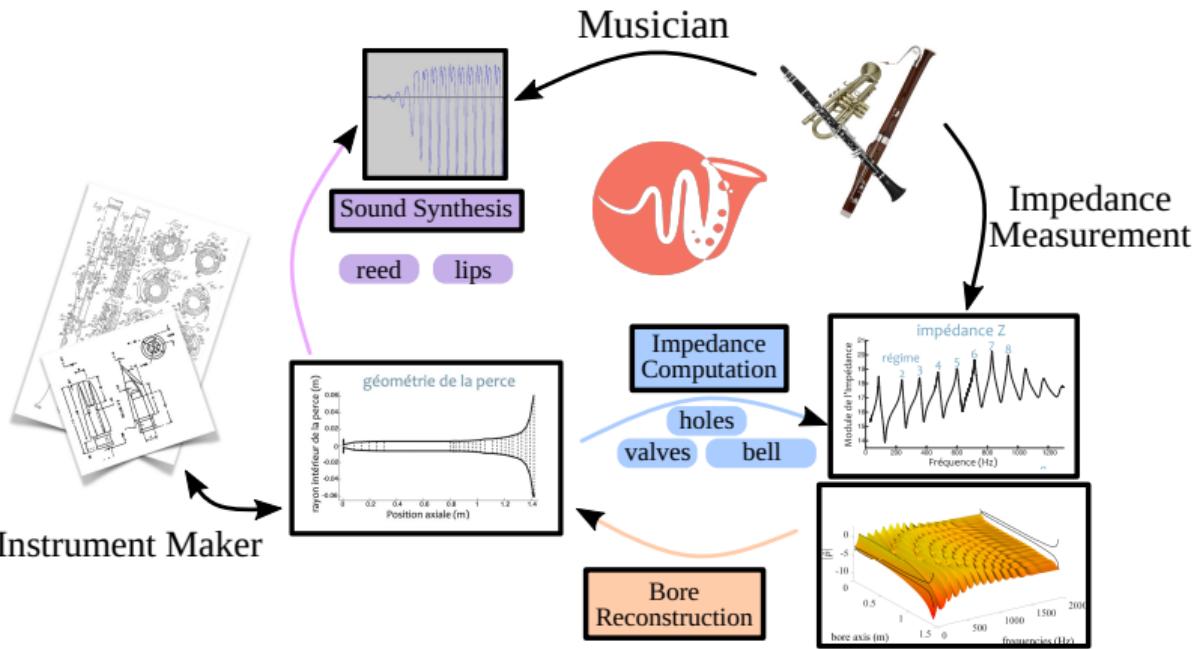
3 Openwind

- Python Library
- Browser interface

4 Teaching activity

5 Conclusion

Features



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
```

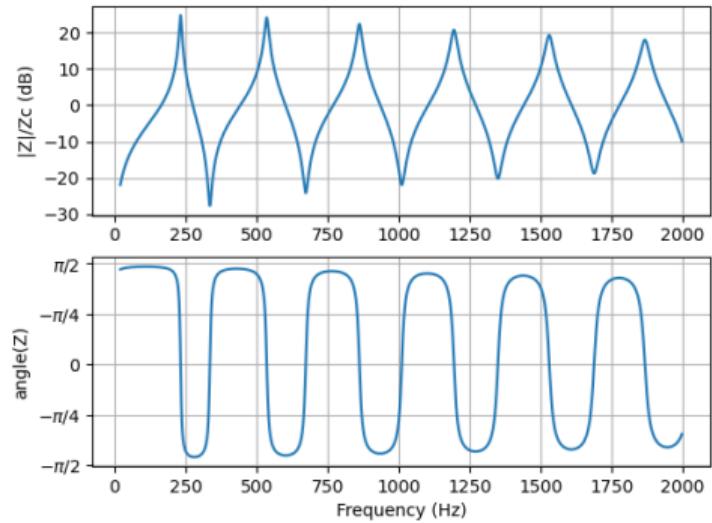
Impedance computation

- 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
```



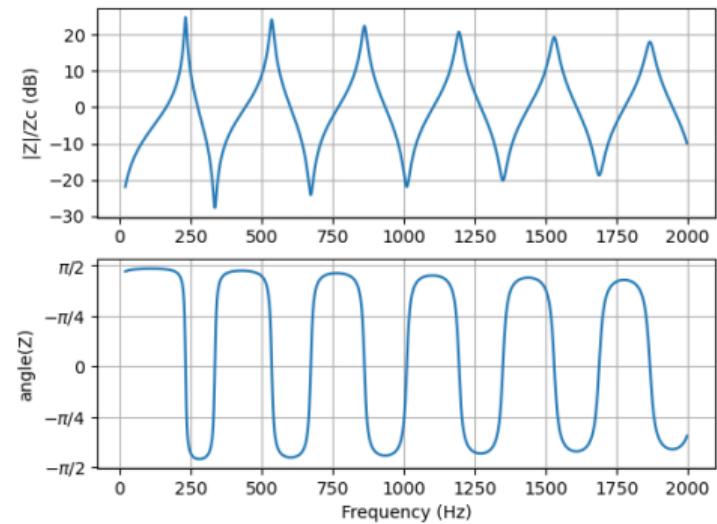
Impedance computation

- 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

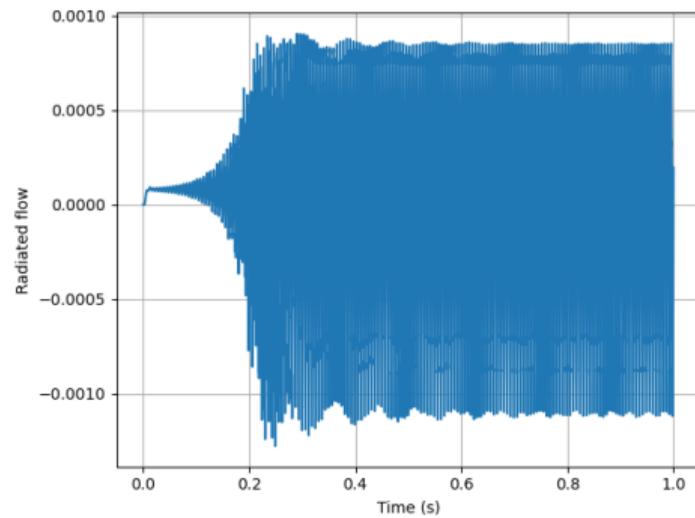
- Let's make sound

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

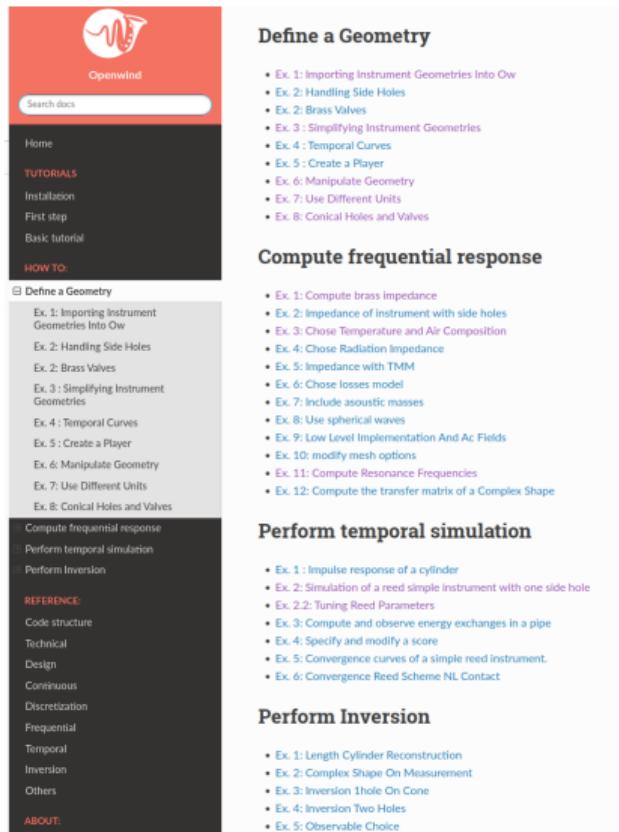
duration = 1.0 # seconds
geometry = [[0, 0.75, 4e-3, 10e-3, 'cone']]
embouchure = Player('WOODWIND_REED_SCALED')

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

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



Other features



The screenshot shows the Openwind software interface. At the top left is the Openwind logo. Below it is a search bar labeled "Search docs". The main menu includes "Home", "TUTORIALS", "Installation", "First step", "Basic tutorial", and "HOW TO:". The "HOW TO:" section is expanded, showing examples categorized by feature: "Define a Geometry" (Ex. 1: Importing Instrument Geometries Into Ow, Ex. 2: Handling Side Holes, Ex. 3: Brass Valves, Ex. 4: Simplifying Instrument Geometries, Ex. 5: Temporal Curves, Ex. 6: Create a Player, Ex. 7: Manipulate Geometry, Ex. 8: Use Different Units, Ex. 9: Conical Holes and Valves), "Compute frequential response" (Ex. 1: Compute brass impedance, Ex. 2: Impedance of instrument with side holes, Ex. 3: Choose Temperature and Air Composition, Ex. 4: Choose Radiation Impedance, Ex. 5: Impedance with TMM, Ex. 6: Choose losses model, Ex. 7: Include acoustic masses, Ex. 8: Use spherical waves, Ex. 9: Low Level Implementation And Ac Fields, Ex. 10: modify mesh options, Ex. 11: Compute Resonance Frequencies, Ex. 12: Compute the transfer matrix of a Complex Shape), "Perform temporal simulation" (Ex. 1: Impulse response of a cylinder, Ex. 2: Simulation of a reed simple instrument with one side hole, Ex. 2.2: Tuning Reed Parameters, Ex. 3: Compute and observe energy exchanges in a pipe, Ex. 4: Specify and modify a score, Ex. 5: Convergence curves of a simple reed instrument, Ex. 6: Convergence Reed Scheme NL Contact), "Perform Inversion" (Ex. 1: Length Cylinder Reconstruction, Ex. 2: Complex Shape On Measurement, Ex. 3: Inversion Hole On Cone, Ex. 4: Inversion Two Holes, Ex. 5: Observable Choice), and "REFERENCE" (Code structure, Technical, Design, Continuous, Discretization, Frequential, Temporal, Inversion, Others). The bottom right corner of the screenshot contains the text "12/23".

Outline

1 Context

2 Physics of wind instruments

3 Openwind

- Python Library
- Browser interface

4 Teaching activity

5 Conclusion

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

Openwind Demo Define Geometry Acoustic Computations Openwind Website

Legal Notice

 **Openwind**
Python library assisting instrument makers

Instrument Geometry

Examples Import Bore Import Holes/Valves Import Fingering Chart Clear All

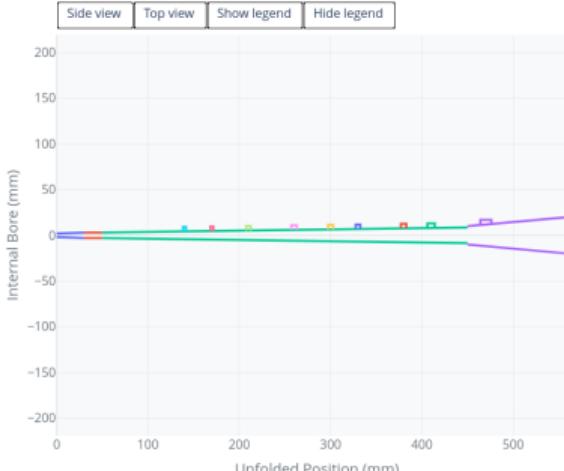
Bore Holes/Valves Fingering Chart

Start Position (mm)	End Position (mm)	Start Diameter (mm)	End Diameter (mm)	Section type	Parameters
0	30	4	6	cone	
30	50	6	6	cone	
50	450	6	17	cone	
450	560	20	40	cone	
560	570	40	36	cone	

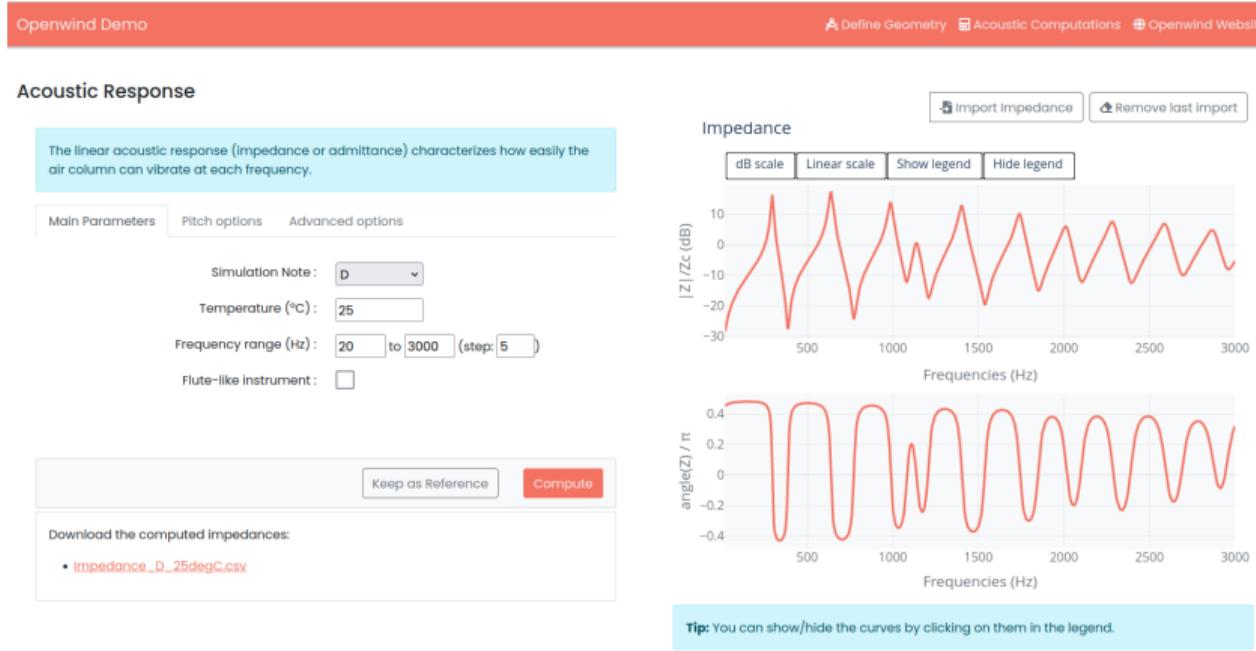
Select All Select none Add line Delete line(s)

Free comments here Export Instrument Plot Instrument

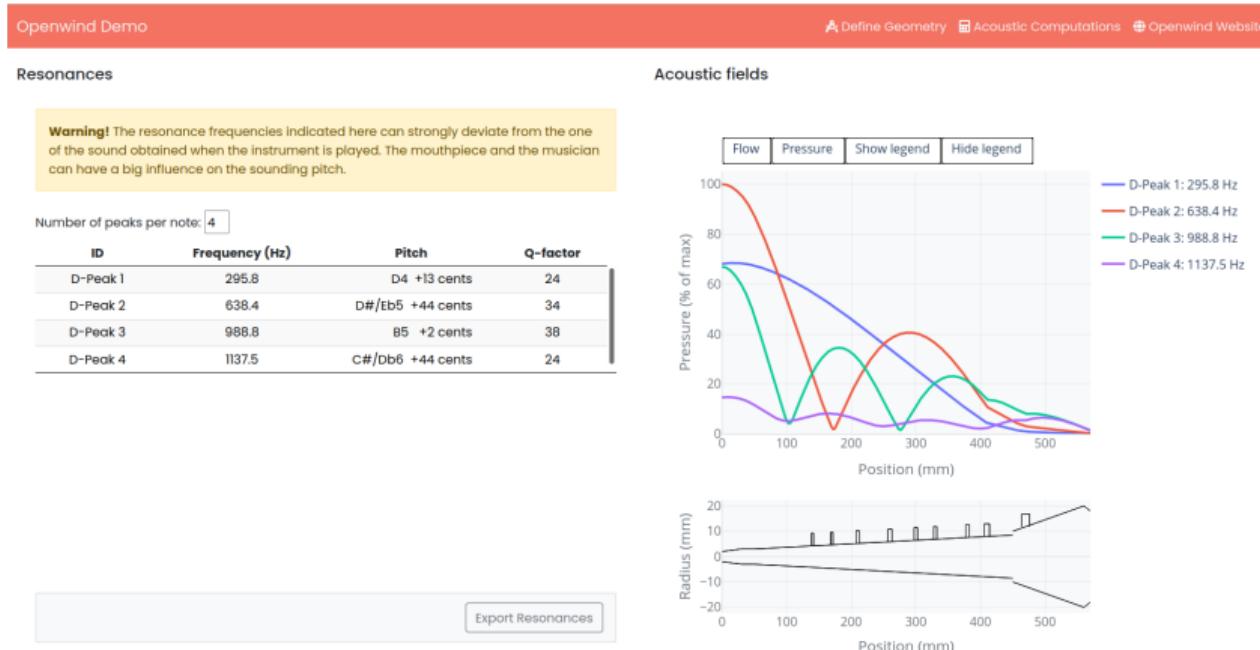
Side view Top view Show legend Hide legend



Demonstration : frequency domain computation



Demonstration : resonances and ac. fields



Outline

1 Context

2 Physics of wind instruments

3 Openwind

4 Teaching activity

- The school
- Activity 1 : harmonicity
- Activity 2 : Tuning a specific resonance

5 Conclusion

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

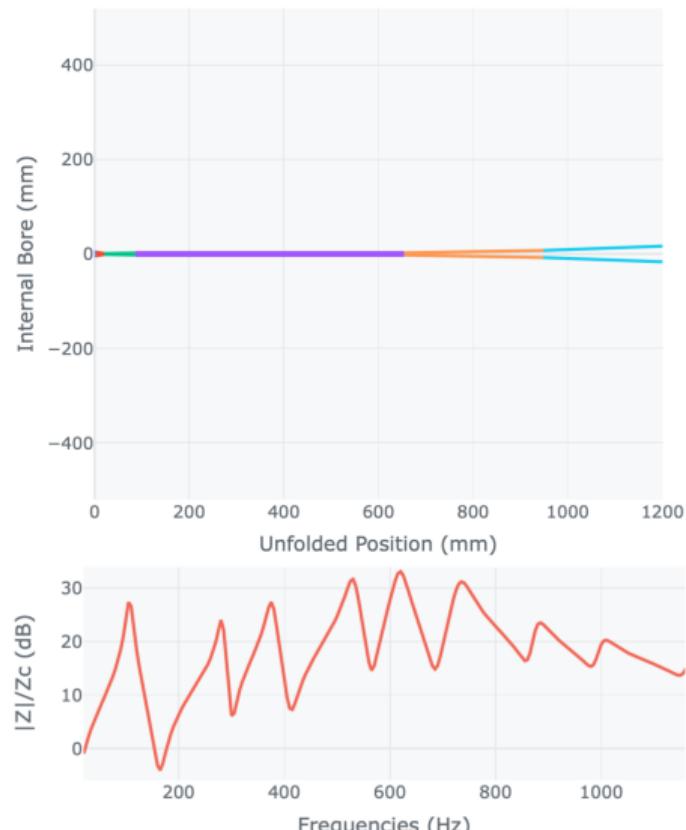
Activity 1 : Optimizing the harmonicity of an instrument (brass like)

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 ($f_n = n f_1$)



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

Objectives

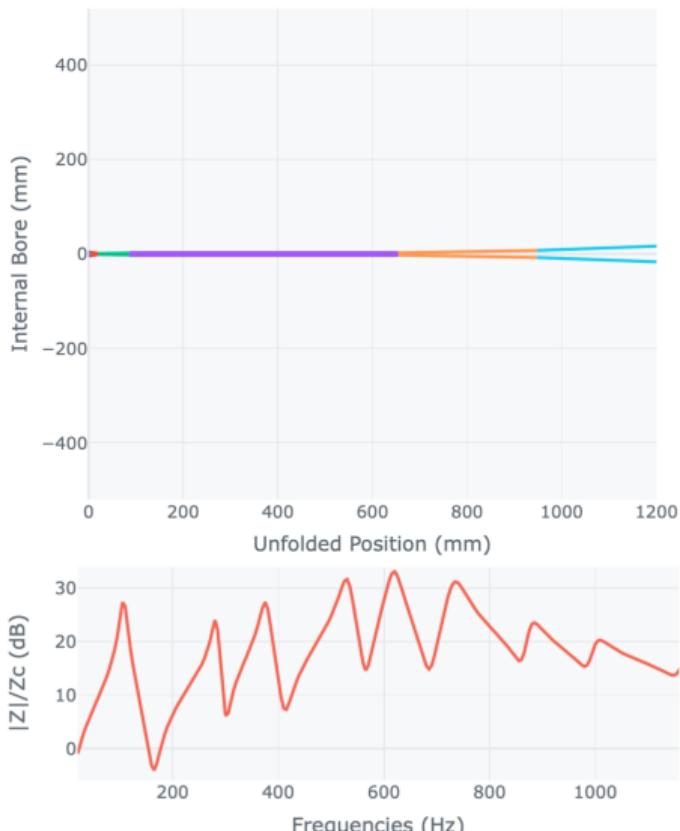
- ▶ getting to know the tool
- ▶ experimenting the link : geometry-harmonicity
- ▶ development of a workflow including simulation tools

● Instructions :

- ▶ Initial geometry : cylindrical and conical parts
- ▶ Modify the geometry to have harmonic resonances ($f_n = n f_1$)

● 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

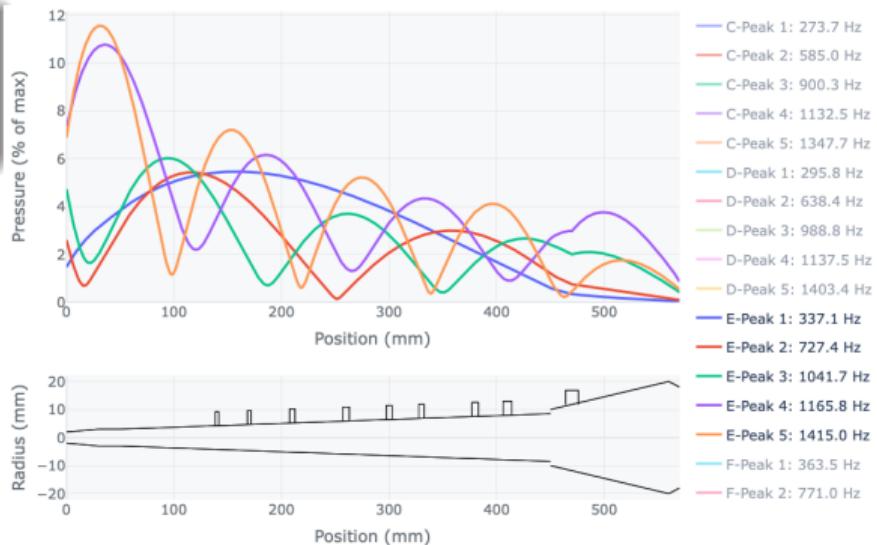
Activity 2 : Tuning a specific resonance (for side holes instruments)

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

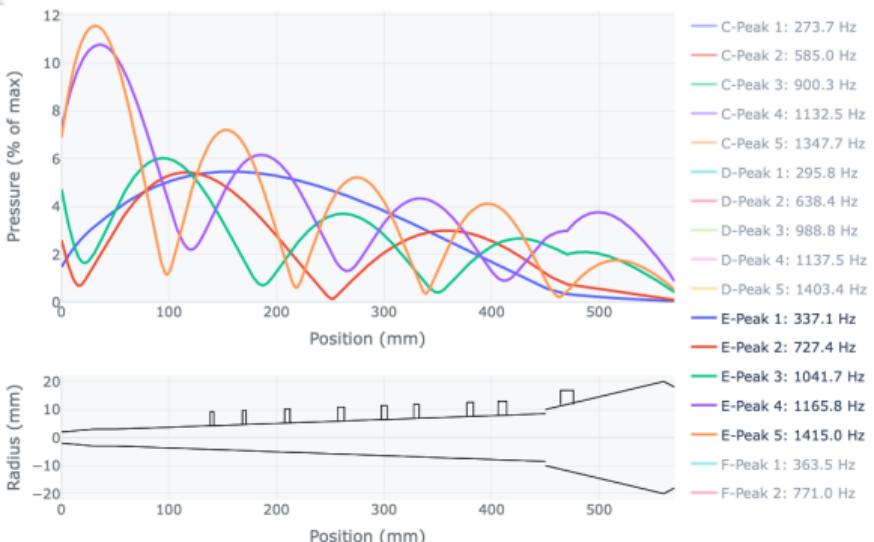
- ▶ Using pressure profile to instrument design
- ▶ Relationship : geometry, pressure profile, resonances

● 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



Outline

1 Context

2 Physics of wind instruments

3 Openwind

4 Teaching activity

5 Conclusion

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>