forum acusticum 2026
8-12 September · Graz, Austria

LOUDPY : AN OPEN-SOURCE PYTHON-BASED FINITE ELEMENT TOOL FOR ELECTRO-VIBROACOUSTIC LOUDSPEAKER SIMULATION

Romain Degraeve[1], Olivier Dazel[2], Mathieu Gaborit[3]

[1] *International Master's Degree in Electroacoustics and Audio, Le Mans University, France*

This paper is a revised version of the authors' submitted manuscript that was peer-reviewed by at least two anonymous reviewers.

Abstract

An important aspect of loudspeaker development is understanding how structural geometry and material properties influence the small-signal frequency response and directivity. This question can be answered by the use of numerical methods such as FEA, which approximate the solution of the governing partial differential equations by solving their weak variational form over a discretised computational domain. However, commercial FEM software remains expensive and offers limited flexibility for customization. This paper presents LoudPy, an open-source Python-based simulation framework that models axymmetrical loudspeakers by coupling a FEM formulation for the acoustic and mechanical domains with a lumped-element model for the electrical circuit. Validation against COMSOL simulations demonstrates agreement within <1% of relative error across the studied frequency range. Key results include small-signal pressure frequency responses, pressure field maps, and a complex eigenvalue analysis, establishing LoudPy as an accessible and reliable alternative for loudspeaker acoustic modeling.

Keywords: *Loudspeaker Modeling, Finite Element Analysis, Electroacoustics, Open-source, Python*

1. Introduction

The Finite Element Method (FEM) [1] has become a standard and practically accessible tool for loudspeaker design, driven by the computational capabilities of modern desktop hardware. By directly solving the structural equations of motion and the acoustic Helmholtz equation over a discretized domain, FEM overcomes the limitations of classical lumped-parameter models, such as the Thiele–Small approach [2]. It enables high-resolution predictions of complex behaviors, including cone break-up modes, radiation patterns, and the influence of enclosure geometries [3].

While commercial packages like COMSOL Multiphysics® are widely used for these tasks, their prohibitive licensing and maintenance costs limit accessibility. Open-source solutions offer an attractive, transparent, and adaptable alternative. However, while general-purpose open-source FEM solvers like FEniCS, FreeFEM, and Elmer [4] exist, they are not specifically tailored to electroacoustics. Setting up a loudspeaker model requires deep expertise in variational formulations, physics coupling, and library configuration, steepening the learning curve for designers.

To address this gap, this project introduces LoudPy: a dedicated, open-source software tool for loudspeaker modeling. Implemented in Python, LoudPy relies only on NumPy and SciPy for numerical computations, and Gmsh for geometry meshing, with all higher-level electroacoustic functionalities built from scratch. LoudPy provides an accessible, community-driven alternative for loudspeaker analysis and is freely available under the GPL v3 license at `https://github.com/RomainDegraeve/LoudPy`.

2. Main Model Assumptions

In loudspeaker engineering, the main quantity of interest is the steady-state frequency response under small-signal excitation [2]. In this regime, both mechanical displacements and acoustic pressure amplitudes remain sufficiently small to justify linear formulations for the structural and acoustic domains. Furthermore the parts having direct influence on the pressure response of most loudspeakers are bodies of revolution, and exploiting this symmetry reduces the problem from three dimensions to two, yielding a substantial reduction in computational cost [5]. The following assumptions are adopted throughout LoudPy.

Corresponding author: *romain.degraeve.etu@univ-lemans.fr.*

Axisymmetric reduction

Both the geometry and the physical fields are assumed axisymmetric, reducing the problem to a two-dimensional meridional cross-section [5].

Isotropic material properties

All solid components are modelled as linearly elastic isotropic materials, characterised by a Young's modulus E, a Poisson's ratio ν, and a mass density ρ.

Isotropic damping

Structural damping is introduced through a material loss factor η, yielding a complex-valued stiffness.

Linear formulations

Both the structural and acoustic governing equations are linearised under the small-signal hypothesis [1].

Quadratic triangular elements (T6)

Mechanical and acoustic fields are interpolated using six-node triangular elements with second-order Lagrange polynomial shape functions [1].

Lumped electrical and magnetic domains.

The electrical and magnetic domains are modelled using lumped-element circuit theory [2], [6]. Unlike the mechanical and acoustic domains, where distributed field solutions provide insight into modal behaviour and directivity [5], electromagnetic FEM serves primarily to characterise $Bl(x)$ and $L_e(x)$ [7], both treated as constants under the small-signal assumption. Dedicated open-source tools such as FEMM [8] already allow accurate estimation of Bl and L_e at the rest position for a given loudspeaker geometry, while R_e requires no field simulation at all, making a redundant electromagnetic solver within LoudPy unnecessary.

3. Model Description

3.1. Acoustical Domain

The governing equation for linear acoustics in the frequency domain is the Helmholtz equation, which describes the propagation of acoustic pressure waves in a fluid medium. It is given by:

$$\nabla^2 p + k^2 p = 0 \quad \text{in } \Omega_a, \tag{1}$$

where p is the acoustic pressure, $k = \frac{\omega}{c}$ is the wavenumber (thermoviscous losses are neglected; $k = \omega/c$ is therefore real-valued), and c is the speed of sound in the medium.

From (1), the axisymmetric discrete variational form can be written as [9]:

$$\underbrace{\left(\int_{\Omega_a} (\nabla \mathbf{N}_p)^T (\nabla \mathbf{N}_p)\, 2\pi r\, d\Omega_a\right)}_{\mathbf{K}_a} \mathbf{p} - k^2 \underbrace{\left(\int_{\Omega_a} \mathbf{N}_p^T \mathbf{N}_p\, 2\pi r\, d\Omega_a\right)}_{\mathbf{M}_a} \mathbf{p} = 0 \tag{2}$$

The $\mathbf{K}_a$ and $\mathbf{M}_a$ corresponds respectively to the acoustical stiffness and mass matrices and are obtained by assembling element-wise contributions.

To satisfy the Sommerfeld radiation condition, a Perfectly Matched Layer (PML) is implemented using complex coordinate stretching. The physical coordinates (r, z) are mapped to complex coordinates $(\tilde{r}, \tilde{z})$ via stretching factors $s_x = \frac{\partial \tilde{x}}{\partial x}$, defined by the polynomial law:

$$s_x(x) = 1 - \mathrm{j}\alpha\lambda_0 \frac{n}{L} \left(\frac{x - x_0}{L}\right)^{n-1} \tag{3}$$

where L is the PML thickness, α is the scaling factor, and λ_0 is the wavelength.

The $\mathbf{K}_a$ and $\mathbf{M}_a$ matrix of the full system are rewritten as [9]:

$$\begin{aligned} \tilde{\mathbf{K}}_a &= \mathbf{K}_a + \mathbf{K}_{pml} \\ \tilde{\mathbf{M}}_a &= \mathbf{M}_a + \mathbf{M}_{pml} \end{aligned} \tag{4}$$

3.2. Mechanical Domain

The governing equation for the mechanical domain in the frequency domain describes the dynamic equilibrium of an elastic structure under harmonic excitation. It is given by the elastodynamic equation:

$$\nabla \cdot \boldsymbol{\sigma} + \omega^2 \rho_s \mathbf{u} = \mathbf{0}, \tag{5}$$

where: $\boldsymbol{\sigma}$ is the Cauchy stress tensor, ω is the angular frequency, ρ_s is the material density, $\mathbf{u}$ is the displacement vector. From (5), the axisymmetric discrete variational in antisymmetric form can be written as :

$$\underbrace{\left(\int_{\Omega_s} \mathbf{B}^T \mathbf{D} \mathbf{B}\, 2\pi r\, d\Omega_s\right)}_{\mathbf{K}_m} \mathbf{u} - \omega^2 \underbrace{\left(\rho_s \int_{\Omega_s} \mathbf{N}_u^T \mathbf{N}_u\, 2\pi r\, d\Omega_s\right)}_{\mathbf{M}_m} \mathbf{u} = 0 \tag{6}$$

The $\mathbf{K}_m$ and $\mathbf{M}_m$ corresponds respectively to the mechanical stiffness and mass matrices and are obtained by assembling all element contributions.

As the damping is assumed isotropic, the $\mathbf{K}_m$ is rewritten as :

$$\tilde{\mathbf{K}}_{\mathbf{m}} = \mathbf{K}_{\mathbf{m}}(1 + \mathrm{j}\eta) \tag{7}$$

where η is the isotropic loss factor.

To extract the structural resonances, the generalized eigenvalue problem is formulated as :

$$\tilde{\mathbf{K}}_m \boldsymbol{\phi}_n = \lambda_n \mathbf{M}_m \boldsymbol{\phi}_n, \tag{8}$$

where $\lambda_n = \omega_n^2$ represents the eigenvalue and $\boldsymbol{\phi}_n$ is the corresponding mode shape. In `LoudPy`, this system is solved for a subset of modes using the ARPACK algorithm (`scipy.sparse.linalg.eigs`) combined with a shift-invert strategy to target modes near a specific frequency.

Because the stiffness matrix $\tilde{\mathbf{K}}_m$ incorporates the complex isotropic loss factor, the resulting eigenvalues λ_n are inherently complex, taking the form $\lambda_n = \omega_{n,0}^2(1 + j\eta)$, where $\omega_{n,0}$ is the undamped natural frequency. This complex nature also extends to the mode shapes $\tilde{\boldsymbol{\phi}}_{\boldsymbol{n}}$ as soon as at least two different values of η are applied to different parts of the mechanical structure. In such cases, the damping becomes non-proportional regarding the whole mechanical structure, resulting in complex mode shapes that capture the spatial phase differences and traveling wave effects between regions with distinct damping properties.

3.3. Electrical Domain

The electrical behavior of the loudspeaker is modeled using a Lumped Element Model (LEM). The LEM represents the electrical circuit of the moving coil, including the voice coil resistance R_e, inductance L_e. The governing equation for the electrical circuit can be expressed as:

$$U(\omega) = R_e I(\omega) + (\mathrm{j}\omega)^n L_e I(\omega) \tag{9}$$

where $U(\omega)$ is the applied voltage and $I(\omega)$ is the current through the voice coil. In this model, L_e is the inductance base and n is the fractional order parameter used to model the lossy nature of the voice coil impedance [10].

3.4. Coupling Terms

To model the multiphysics behavior of the loudspeaker, coupling terms are introduced to link the acoustic, mechanical, and electrical equations (2, 6, 9).

3.4.1. Acoustical-Mechanical Coupling

The interaction between the fluid domain(s) and the structural components is governed by the coupling matrix $\mathbf{C}_{am}$, defined over the shared boundary Γ_{am}:

$$\mathbf{C}_{am} = \int_{\Gamma_{am}} \mathbf{N}_p^T \mathbf{n} \mathbf{N}_u \,(2\pi r)\, d\Gamma, \tag{10}$$

where $\mathbf{n}$ is the outward unit normal vector to the mechanical domain. This results in two interaction terms: the mechanical force exerted by the acoustic pressure on the structure $\mathbf{F}_p(\omega)$, and the acoustic source term generated by the structural acceleration $\mathbf{Q}_u(\omega)$:

$$\mathbf{F}_p = \mathbf{C}_{am}\mathbf{p} \tag{11}$$

$$\mathbf{Q}_u = \rho_0 \omega^2 \mathbf{C}_{am}^T \mathbf{u} \tag{12}$$

where ρ_0 is the fluid density.

3.4.2. Mechanical-Electrical Coupling

The mechanical and electrical domains are coupled via the electromagnetic transducer effect within the voice coil. Because the macroscopic force factor Bl is a lumped parameter representing the total electromechanical conversion, the coupling must be normalized by the total area of the coupling interface to correctly distribute the force.

Let $S_{me} = \int_{\Gamma_{me}} (2\pi r)\, d\Gamma$ be the surface area of the coil boundary. The normalized coupling vector $\mathbf{C}_{me}$ is defined over Γ_{me} as:

$$\mathbf{C}_{me} = \frac{1}{S_{me}} \int_{\Gamma_{me}} \mathbf{N}_u^T \mathbf{e}_z \,(2\pi r)\, d\Gamma, \tag{13}$$

where $\mathbf{e}_z$ is the unit vector in the direction of motion. Note that the voice coil is assumed rigid due its high stiffness compared to the diaphragm thus the volumetric Lorentz forced is simplified here as a surface integration over Γ_{me}. This assumption yields equivalent results with a lower computational cost.

This coupling accounts for the back-electromotive force $e(\omega)$ induced in the electrical circuit, and the Lorentz force $\mathbf{F}_{\text{coil}}$ acting on the mechanical structure:

$$e(\omega) = j\omega\, Bl\, \mathbf{C}_{me}^T \mathbf{u} \tag{14}$$

$$\mathbf{F}_{\text{coil}} = Bl\, I(\omega)\, \mathbf{C}_{me} \tag{15}$$

3.5. Fully coupled system

Thanks to the equations (2, 6, 9) modelling the three physical domains and the coupling relations (11, 12, 14, 15), it is possible to assemble the fully coupled block matrix system:

$$\begin{bmatrix} \underset{\text{Acoustical domain}}{\boxed{\tilde{\mathbf{K}}_a - k^2 \tilde{\mathbf{M}}_a}} & \underset{\text{Meca}\to\text{Acou}}{\boxed{-\rho_0\omega^2 \mathbf{C}_{am}^T}} & \mathbf{0} \\ \underset{\text{Acou}\to\text{Meca}}{\boxed{-\mathbf{C}_{am}}} & \underset{\text{Mechanical domain}}{\boxed{\tilde{\mathbf{K}}_m - \omega^2 \mathbf{M}_m}} & \underset{\text{Elec}\to\text{Meca}}{\boxed{-Bl\,\mathbf{C}_{me}}} \\ \mathbf{0} & \underset{\text{Meca}\to\text{Elec}}{\boxed{j\omega\, Bl\, \mathbf{C}_{me}^T}} & \underset{\text{Electrical domain}}{\boxed{R_e + (j\omega)^n L_e}} \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{u} \\ I \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ V \end{bmatrix} \tag{16}$$

Solving this system gives access to the three field variables $\mathbf{p}(\omega)$, $\mathbf{u}(\omega)$, and $I(\omega)$ as a function of V_{in}, the input voltage of the coil. This is done in LoudPy by assembling the frequency-dependent dynamic operators into a single monolithic sparse block matrix using `scipy.sparse.bmat`, and solving the global linear system simultaneously at each frequency step using the direct sparse solver `scipy.sparse.linalg.spsolve`.

4. Comparison with COMSOL Multiphysics®: Case of a Circular Diaphragm

To validate the finite element formulations implemented in LoudPy, a comparative study is performed using COMSOL Multiphysics® as a reference.

To keep this comparison straightforward, the considered geometry is a thin, axisymmetric, deformable membrane clamped at its outer edge. An external vertical force is applied at the center of the membrane. Two acoustic domains are coupled to the front and rear surfaces of the membrane. Furthermore, the membrane is mounted in a rigid baffle, which is modeled using the natural boundary condition of the Helmholtz equation (i.e., the normal derivative of the pressure is zero at the boundary).

Identical simulation setups are defined in both COMSOL and LoudPy, using the exact same geometry, material parameters, mesh sizing, and Perfectly Matched Layer (PML) dimensions.

A first comparison evaluating the eigenfrequencies and the loss factors $\left(\eta = \frac{\Im(\lambda)}{\Re(\lambda)}\right)$ of the uncoupled mechanical structure is presented in Figure 1.

A second comparison evaluates the acoustic pressure

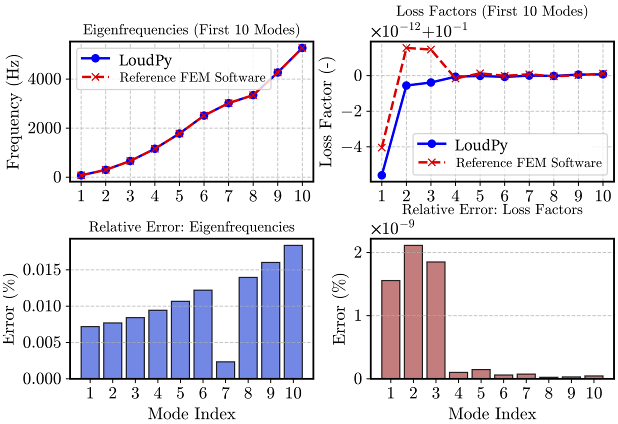

Figure 1: Comparison of eigenfrequencies and loss factors between LoudPy and COMSOL Multiphysics® for the mechanical structure.

response (both Sound Pressure Level and phase) generated by an applied force of 1 N. The observation points are located at a distance of 0.5 m from the center of the membrane at four different angles as shown in Figure 2.

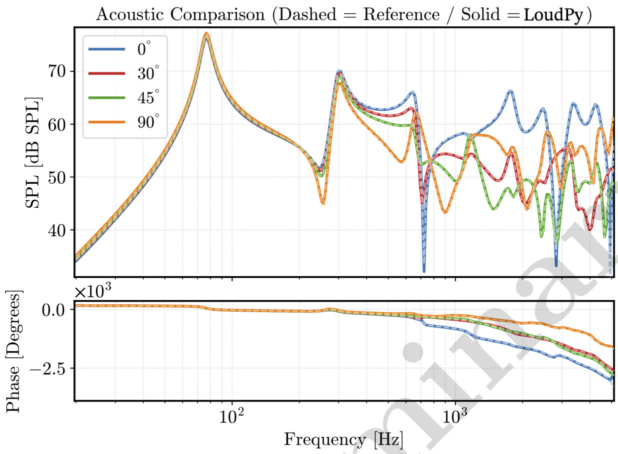

Figure 2: Comparison of the acoustic pressure response (SPL and phase) between LoudPy and COMSOL Multiphysics®.

Both of these studies show excellent agreement between LoudPy and COMSOL. The relative error between the two software packages is strictly under 0.02% for both the eigenfrequencies and the loss factors for first 10 modes.

Concerning the acoustic pressure response, despite the complex behavior induced by the mechanical deformations of the membrane, the curves (dotted lines for COMSOL and solid lines for LoudPy) remain superimposed with less than 1 dB of difference. The phase curves are also nearly perfectly superimposed, showing a difference of less than 1°.

The minor discrepancies between the COMSOL and LoudPy results can be explained by the fact that the meshing is generated internally by each software. Consequently, the meshes are not perfectly identical (although they share the same maximum size parameters), which naturally accounts for these very small numerical differences.

5. Application to the Geometry of a Moving Coil Loudspeaker

This section presents the simulation results for the fully coupled system governed by Equation 16. The geometry is generated directly via the Gmsh API or imported from a STEP file. A critical aspect of this process is the definition of physical groups in Gmsh. This allows the Python script to automatically categorize elements into distinct domains (e.g., acoustic fluid, mechanical structure) and boundaries (e.g., fluid-structure interfaces, forced excitation surfaces, and symmetry axes where radial displacement is constrained).

Figure 3 illustrates the scaled mechanical deformation and the near-field acoustic pressure (within a 30 cm × 30 cm square region) for an imposed excitation at 10 kHz.

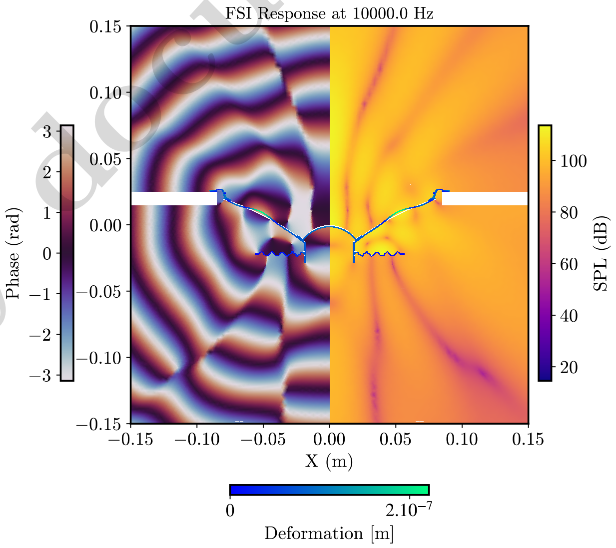

Figure 3: Scaled mechanical deformation and near-field acoustic pressure map (30 cm × 30 cm) for an axisymmetric loudspeaker (r = 8 cm) in a baffle, simulated at 10 kHz using LoudPy.

These results were calculated using LoudPy for an axisymmetric loudspeaker with an 8 cm radius mounted in a rigid baffle. This specific frequency was chosen because the acoustic wavelength is significantly smaller than the radius of the transducer (which is a typical size for a woofer), thereby highlighting complex acoustic interference patterns in the radiation field.

Another capability enabled by LoudPy is the use of the complex eigenvalue solver to extract mode shapes, their corresponding eigenfrequencies, and the associated loss factors. Figure 4 shows the structural mode shapes zoomed in on the surround region, at modes where the displacement is maximal in this region, for the same loudspeaker presented in Figure 3. Both the positive and negative extremes of the displacement are plotted, allowing for clear visualization of the nodes and antinodes of the mode shapes.

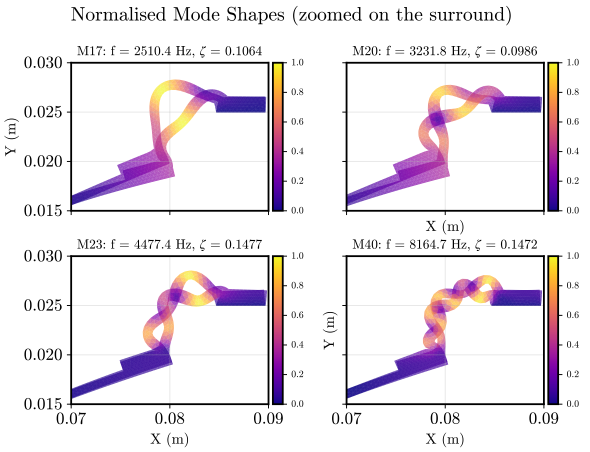

Figure 4: Structural mode shapes zoomed in on the surround of the loudspeaker. Both positive and negative displacement extremes are superimposed to visualize the nodal points and antinodes. Computed in LoudPy.

Such an analysis can be performed very rapidly. The calculation time to extract all modes under 20 kHz is typically less than a minute, even with a mesh density of at least three elements across the thinnest sections. This computational efficiency allows the user to quickly iterate over material properties or geometrical parameters to shift or damp undesirable resonances, ultimately helping to flatten the acoustic pressure response of the modeled transducer.
Figure 5 presents the main results that can be obtained with LoudPy for the loudspeaker geometry depicted in Figure 3. These include the on-axis and off-axis acoustic pressure responses (SPL) evaluated at 1 W and 1 m, as well as the electrical impedance.

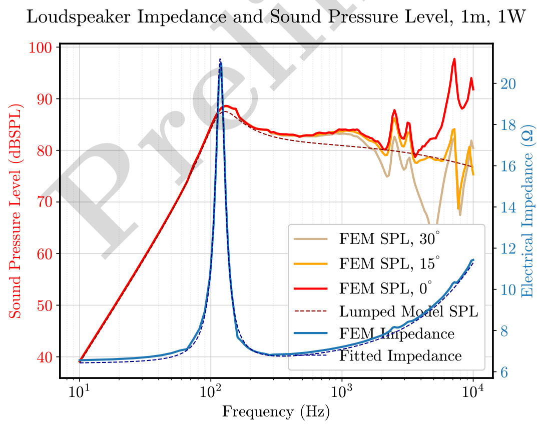

Figure 5: Simulated acoustic and electrical performance of the loudspeaker. The figure shows the on-axis and off-axis SPL (1 W, 1 m), the electrical impedance curve with its lumped-element fit.

Finally, a curve-fitting procedure is performed on the electrical impedance to extract the lumped Thiele-Small parameters (M_{ms}, C_{ms}, and R_{ms}) from the FEM model. These fitted values are used to generate lumped-element equivalent curves for both the SPL response and the electrical impedance using analytical formulas.
The mechanical and electrical impedances are defined as:

$$Z_m(\omega) = j\omega M_{ms} + R_{ms} + \frac{1}{j\omega C_{ms}} \tag{17}$$

$$Z_{\text{etot}}(\omega) = R_e + (j\omega)^n L_e + \frac{(Bl)^2}{Z_m(\omega)}. \tag{18}$$

Using the baffled monopole assumption, the on-axis acoustic pressure at a distance r, approximated by:

$$p(\omega, r) = \frac{j\omega\rho_0 S_d}{2\pi r} \cdot \frac{Bl{\cdot}U_{\text{in}}}{(R_e+(j\omega)^n L_e){\cdot}Z_m(\omega)+(Bl)^2} \tag{19}$$

Figure 5 shows an agreement with less than 1 dB of difference between the on-axis pressure response calculated by LoudPy and the lumped-element monopole approximation at lower frequencies (below the first structural resonance). Above this frequency, more complex behavior starts to appear in the FEM-computed response due to the mechanical deformations of the membrane and the resulting acoustic interferences. Based on the three standard angles studied (0°, 15°, and 30°), the off-axis pressure response is satisfactory between 100 Hz and 1 kHz, although the parameters and shape of the loudspeaker could be further optimized to achieve an even flatter response.
Figure 6 illustrates a classical directivity map for the loudspeaker depicted in Figure 3, evaluating the acoustic pressure at a distance of 0.5 m over an angular range from -90° to 90°.

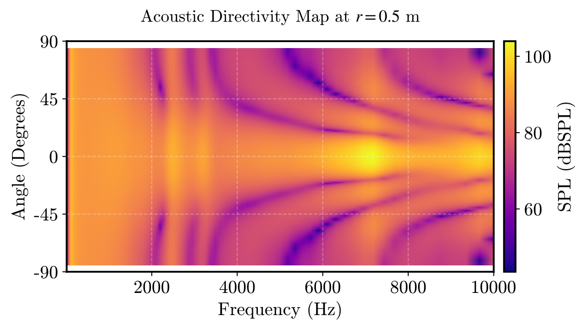

Figure 6: Acoustic directivity map of the loudspeaker at a distance of 0.5 m, showing the sound pressure level (SPL) across an angular range from -90° to 90°.

The directivity is mostly flat below 2 kHz, with a variation of less than ±5 dB between any off-axis angle and the on-axis pressure level. Above this frequency, more complex radiation behavior begins to emerge, characterized by the appearance of directivity nulls (or "zero lines"). Additionally, structural resonances of the speaker cone, dustcap and surround become visible as fluctuations in the on-axis pressure level.

6. Conclusion

This paper introduced LoudPy, an open-source FEM solver tailored for electroacoustic simulations, and the main theoretical and simplifying assumptions it relies on. A comparative study with COMSOL Multiphysics® was conducted, demonstrating excellent agreement and validating the accuracy of the proposed open-source framework. Furthermore, this work showcased LoudPy's ability to generate classical electroacoustic analyses. These include structural mode shapes, fluid-structure interaction (FSI) deformations, acoustic pressure responses, and electrical impedance curves. The framework's capability to perform automated curve-fitting for lumped Thiele-Small (T&S) parameters and compute detailed acoustic directivity maps was also demonstrated.

Future work will focus on expanding LoudPy's capabilities. Planned developments include the implementation of time-domain solvers, the extension to full 3D geometries using shell elements to model non-axisymmetric transducer and more specifically non-axisymmetric acoustic rear loads, integrated magnetic field simulations for the motor structure, and the development of a Graphical User Interface (GUI) to further enhance its accessibility for loudspeaker designers.

7. Acknowledgments

I sincerely thank Prof. Olivier Dazel (Director of LAUM) and Prof. Mathieu Gaborit for their weekly guidance, availability, and for introducing me to the finite element method. I am also grateful to Prof. Manuel Melon and Prof. Bruno Gazengel for making this project possible within the IMDEA Master's programme. Finally, I thank everyone who supported this work; your interest in an open-source Python FEM tool for loudspeaker modeling was a great source of motivation.

CRediT Author Statement

Romain Degraeve: Conceptualization; Methodology; Software; Validation; Formal analysis; Investigation; Data Curation; Writing – Original Draft; Visualization.

Dr Olivier Dazel: Conceptualization; Methodology; Resources; Writing – Review & Editing; Supervision; Project administration.

Dr Mathieu Gaborit: Conceptualization; Methodology; Writing – Review & Editing; Supervision; Project administration.

Dr Manuel Melon : Project administration.

References

[1] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The Finite Element Method: Its Basis and Fundamentals*, 6th. Oxford: Butterworth-Heinemann, 2005.

[2] A. N. Thiele, "Loudspeakers in vented boxes," *Journal of the Audio Engineering Society*, vol. 19, no. 5, pp. 382–392, 1971.

[3] COMSOL AB, *Loudspeaker driver—frequency-domain analysis*, `https://www.comsol.com/model/loudspeaker-driver-frequency-domain-analysis-276`, 2026. Accessed: Jan. 15, 2026.

[4] A. Logg, K.-A. Mardal, and G. N. Wells, Eds., *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*. Berlin: Springer, 2012. DOI: `10.1007/978-3-642-23099-8`.

[5] G. Bank and J. R. Wright, "Loudspeaker driving-point impedance curves," *Journal of the Audio Engineering Society*, vol. 31, no. 4, pp. 230–236, 1983. Accessed: Jun. 15, 2026. [Online]. Available: `https://www.aes.org/e-lib/browse.cfm?elib=XXXX`.

[6] W. M. Leach, *Introduction to Electroacoustics and Audio Amplifier Design*, 3rd. Kendall/Hunt, 2002.

[7] W. Klippel, "Tutorial: Loudspeaker nonlinearities—causes, parameters, symptoms," *Journal of the Audio Engineering Society*, vol. 54, no. 10, pp. 907–939, 2006.

[8] D. Meeker, *FEMM: Finite element method magnetics, version 4.2*, `http://www.femm.info`, 2010. Accessed: Jan. 15, 2026.

[9] O. Dazel, "Lecture notes in numerical methods for acoustics," Université du Mans, Fall 2015. `https://perso.univ-lemans.fr/~odazel/wa_files/L5_Numerical_methods.pdf`, 2015.

[10] J. R. Wright, "An empirical model for loudspeaker motor impedance," *Journal of the Audio Engineering Society*, vol. 38, no. 10, pp. 749–754, Oct. 1990.

Use of Artificial Intelligence Tools

I used state-of-the-art generative artificial intelligence models. Specifically, these included
GPT–4o, Anthropic Claude 3.7 Google Gemini 2.5 Pro a Mistral AI (Mistral Large), and DeepSeek (DeepSeek–R1 and DeepSeek–V3).

These tools were used to assist with handling and visualizing numerical results, organizing, the analysis of matrix numerical stability, the selection of appropriate numerical methods for solving linear systems, as well as structuring, refactoring, and functionally organizing specific parts of the code.

In addition, these models were also used to help improve grammar, spelling, and overall clarity.