

Comparative Analysis of Finite Difference and Finite Element Models for Audio Waveform Simulation

Juliette FLORIN, EURECOM

Semester project 2025

Contents

0.1	Introduction	1
0.2	Objective	2
0.3	Context and Motivation	2
0.4	The models and the theory	2
0.4.1	Finite Element Method (FEM)	2
0.4.2	Finite Difference Method (FDM)	3
0.4.3	Pressure to sound	3
0.4.4	Acoustic pressure model	3
0.5	The guitar string without damping and without modulus of elasticity	4
0.5.1	The model	4
0.5.2	Theory	4
0.5.3	Analytical resolution	5
0.5.4	Finite Difference Method	7
0.5.5	Finite Element Method (FEM)	9
0.6	Comparing FEM and FDM guitar string	11
0.6.1	Tension	12
0.6.2	Time interval	13
0.6.3	Number of nodes (points)	15
0.6.4	Time of simulation	17
0.7	To compare to reality: guitar with damping and without the modulus of elasticity	18
0.7.1	Comparison to Reality	19
0.7.2	Important note on the simulation	22
0.8	The cycling bell with damping and modulus of elasticity	23
0.8.1	The models	23
0.8.2	Theory	23
0.8.3	Analytical resolution	23
0.8.4	Comparing analytical to reality	29
0.8.5	Finite Difference Method	30
0.9	Conclusion	34
0.9.1	General conclusion	34
0.9.2	Additional work possible	34

0.1 Introduction

Accurate numerical modeling of sound-producing structures is fundamental in musical acoustics, audio engineering, virtual instrument synthesis, and many other cases. Among the available computational methods, the Finite Element Method (FEM) is commonly employed for simulating the vibrational behavior of complex structures, such as musical instruments. However, its effectiveness compared to alternative approaches, such as the Finite Difference Method (FDM), remains underexplored.

This study aims to assess the suitability of FEM for audio waveform simulation, investigating whether it provides superior accuracy in predicting the resonant frequencies, mode shapes, and acoustic response of physical objects. We evaluate the trade-offs in computational efficiency, convergence properties, and precision by conducting a comparative analysis with FDM. The results of this research will contribute to a deeper understanding of computational acoustics, used for both theoretical and practical applications in musical instrument modeling, acoustic design, and digital sound synthesis.

0.2 Objective

The primary goal of this project is to compare the performance and accuracy of the Finite Difference Method (FDM) and the Finite Element Model (FEM) in simulating audio waveforms for various physical phenomena. Specifically, we aim to generate and analyze the sound waveforms produced by two chosen objects: a guitar string and a simple bell. The project will investigate how the physical properties of these objects influence the behavior of both models and the resulting audio simulations.

0.3 Context and Motivation

A wave is an oscillatory phenomenon (a quantity that varies over time) propagating through space. In acoustics, it refers to vibrations of matter, meaning a mechanical motion oscillating in a continuous medium, whether fluid or solid. Examples of waves include the vibration of a string, waves on the surface of water, sound propagating through air, etc.

A hemispherical shell representing a cycling bell is a structure that experiences mechanical vibrations, with each vibration mode corresponding to a specific resonant frequency.

Studying these vibrations requires solving partial differential equations that describe different vibration types, such as longitudinal, transverse, flexural, and torsional. For simpler systems like strings or beams, these equations include the second-order D'Alembert equation and the fourth-order Euler-Bernoulli beam equation.

The vibrational behavior of hemispherical structures depends on natural frequencies, vibration modes, geometry, and material properties. Understanding these properties is crucial for engineering and scientific applications, with various methods available for analysis.

In the industry today, FEM is mainly used. [1] In many reports, it is said to be easier to compute and more efficient [2] [3] than the FDM. And many other statements comparing the two are made without any proof. However, here, we would like to understand why such statements are made in the context of acoustic waves and try to compute the waveform of an object.

0.4 The models and the theory

Two main models can be used to resolve a physical problem. Other models are derived from cases of these general ones. The first one is the FDM. This method is used when we have a physical equation to resolve. If that Equation is not resolvable, you can use the FDM. The second one is the FEM, which is used to model complex objects without too many simplifications.

In theory, those two methods used in the context of modeling physical phenomena should give you the same answer as the theory. However, there are numerical methods; thus, the results depend on the computational preciseness.

0.4.1 Finite Element Method (FEM)

The FEM is a numerical technique for solving differential equations, particularly useful for complex geometries and variable material properties. [4] The process involves transforming the strong form of the governing equations into a weak form, followed by discretization into finite elements. This results in the need to compute global system matrices, such as the mass and stiffness matrices.

In FEM, the computational domain is divided into smaller elements, and the solution is approximated using interpolation functions (shape functions) within each element. The unknowns, typically nodal displacements or field variables, are determined by assembling the contributions from individual elements into a global system of equations. We can see in the image below an example of how to cut down a string into small elements. Each element has a given length, weight, and properties, and we must represent a small portion of what we are trying to describe.

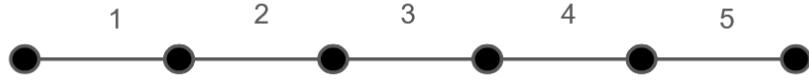


Figure 1: FEM representation example, we consider the elements: the lines

0.4.2 Finite Difference Method (FDM)

Sometimes referred to as the Euler Method, the Finite Difference Method (FDM) is not widely adopted in the industry but remains highly effective when applied to well-understood problems. FDM is a numerical technique for solving differential equations by directly approximating derivatives at discrete points within the computational domain. Unlike the FEM, FDM does not require a weak formulation but instead discretizes the strong form of the governing equations.

This method approximates derivatives using finite difference approximations, including forward, backward, and central differences. These approximations transform the differential equations into algebraic equations, linking function values at discrete grid points. We can see in the image below how small points would represent a string.



Figure 2: FDM representation example, we consider de nodes: the points

0.4.3 Pression to sound

Once we have the amplitude in time of a given model, we can compute the waveform at one distance, given the pressure felt at that distance in time.

In an acoustic simulation, the pressure generated by a source can be calculated using various physical parameters. If SI (International System of Units) units are used for all quantities, the resulting acoustic pressure is expressed in Pascals (Pa). However, to create a digital audio file, it is necessary to convert this acoustic pressure into an amplitude compatible with the audio format, typically normalized within the range $[-1, 1]$.

0.4.4 Acoustic pressure model

The acoustic pressure $p(r, t)$ generated by a source (for example, a vibrating string) is given by the equation:

$$p(r, t) = \frac{\rho_0 c_0}{4\pi r} \frac{\partial u(x, t_r)}{\partial t}$$

Where:

- $p(r, t)$ is the acoustic pressure in Pascals (Pa),
- ρ_0 is the air density in kg/m^3 ,
- c_0 is the speed of sound in air in m/s ,
- r is the distance to the source in m,
- $u(x, t_r)$ is the transverse displacement of the string in m,
- $\frac{\partial u}{\partial t}$ is the transverse velocity in m/s .

Let us note:

$$p(r, t) \sim \frac{\text{kg}/\text{m}^3 \cdot \text{m}/\text{s} \cdot \text{m}/\text{s}}{\text{m}} = \text{Pa}$$

Conversion of acoustic pressure to audio amplitude

Several steps must be followed to convert the calculated acoustic pressure into an amplitude compatible with a digital audio recording.

Normalization of pressure The first step is to normalize the acoustic pressure $p(t)$ by a reference pressure p_{\max} , to avoid saturation in the audio file. The normalization formula is:

$$A(t) = \frac{p(t)}{p_{\max}}$$

where p_{\max} is the maximum pressure in the simulation, and $A(t)$ is the normalized amplitude within the range $[-1, 1]$.

Mapping to digital audio format Once the pressure is normalized, the amplitudes can be adapted to match digital audio formats. For example, for a 16-bit PCM audio file, the normalized amplitude $A(t)$ can be converted to an amplitude within the range $[-32768, 32767]$ as follows:

$$\text{Amplitude 16-bit} = [A(t) \cdot 32767]$$

Creating the audio file Using Python, we can write directly in the audio file in WAV format.

Following these steps, the acoustic pressure calculated in a simulation can be converted into an appropriate audio amplitude, creating a digital audio file compatible with common standards (such as 16-bit PCM). This conversion ensures that the results from acoustic simulations can be easily analyzed or used in real-world audio applications.

0.5 The guitar string without damping and without modulus of elasticity

0.5.1 The model

Consider a string stretched along the horizontal axis between two fixed points, A and B . To visualize the tension T in the string, imagine that the string is attached at point A and pulled by a mass m via a pulley at point B . The tension T is given by: $T = mg$ where $g = 9.81 \text{ m/s}^2$ is the acceleration due to gravity. For example, for a mass of 1 kg, the tension $T = 9.81 \text{ N}$



Figure 3: Example string

When the string is at rest in equilibrium, the forces of tension in both directions cancel each other out (T and $-T$). The length of the string between its two fixed points is denoted as $L = AB$. The linear mass density of the string (mass per unit length) is represented by μ . We do not consider the Damping. Damping is not useful to know the frequencies we will hear. We also do not consider the material's elasticity because, as we will see later, they are negligible for the string in nylon; note that this is not the case for a metal string, as the elasticity is significant.

0.5.2 Theory

We are looking for the acoustic waves that are given when playing a guitar string without the elasticity part of the equation or the damping. [5] [6]

D'Alembert's wave equation The wave equation for transverse vibrations of the string, derived by D'Alembert, is given by:

$$\frac{\partial^2 y(x, t)}{\partial t^2} = \frac{T}{\mu} \frac{\partial^2 y(x, t)}{\partial x^2}$$

Where:

- $y(x, t)$ represents the transverse displacement of the string at position x and time t ,
- T is the tension in the string (in Newtons),
- μ is the linear mass density of the string (in kg/m),
- x is the spatial coordinate along the length of the string.

This equation describes the string's motion under the influence of tension and its mass distribution.

Wave speed The speed c of a transverse wave on the string is related to the tension T and the linear mass density μ by the following equation:

$$c = \sqrt{\frac{T}{\mu}}$$

Where:

- c is the speed of the wave propagation along the string (in meters per second),
- T is the tension in the string (in Newtons),
- μ is the linear mass density of the string (in kg/m).

This relationship shows that the wave speed increases with the tension and decreases with the mass per unit length of the string. The wave propagates faster for a string with higher tension, while a string with higher mass density will slow down the wave.

0.5.3 Analytical resolution

The vibrating string equation The equation describing the transverse vibrations of a stretched string is the wave equation:

$$\frac{\partial^2 y(x, t)}{\partial t^2} = c^2 \frac{\partial^2 y(x, t)}{\partial x^2}$$

Where:

- $y(x, t)$ is the transverse displacement of the string at position x and time t ,
- $c = \sqrt{\frac{T}{\mu}}$ is the speed of propagation of waves on the string, where T is the tension of the string and μ is the linear mass density of the string.

Solution in terms of standing waves For a vibrating string fixed at both ends (i.e., $y(0, t) = 0$ and $y(L, t) = 0$), we seek a solution in the form of standing waves.

Thus, we assume a solution of the form:

$$y(x, t) = f(x)g(t)$$

where $f(x)$ is a spatial function that depends on the position x along the string, and $g(t)$ is a temporal function that depends on time t .

Substituting $y(x, t) = f(x)g(t)$ into the wave equation, we get:

$$f(x)g''(t) = c^2 f''(x)g(t)$$

Divide both sides of this equation by $f(x)g(t)$, which gives:

$$\frac{g''(t)}{g(t)} = c^2 \frac{f''(x)}{f(x)}$$

Since the left-hand side depends only on t and the right-hand side depends only on x , both sides must be equal to a constant, which we will call $-\omega^2$, where ω is the angular frequency. This gives us two separate equations:

$$g''(t) + \omega^2 g(t) = 0$$

$$f''(x) + \frac{\omega^2}{c^2} f(x) = 0$$

Solving equations

$$\begin{aligned} g(t) &= A \cos(\omega t) + B \sin(\omega t) \\ f(x) &= C \cos\left(\frac{\omega}{c}x\right) + D \sin\left(\frac{\omega}{c}x\right) \end{aligned}$$

Boundary conditions The boundary conditions are that the string is fixed at both ends, i.e.,

$$f(0) = 0 \quad \text{and} \quad f(L) = 0$$

We apply these conditions to the solution for $f(x)$.

This gives us that the quantized values for ω are:

$$\omega_n = \frac{n\pi c}{L} \quad \text{where} \quad n = 1, 2, 3, \dots$$

We also have that at time 0, there is no speed.

$$g'(t) = 0 \quad \text{for all } t.$$

So

$$g(t) = A \cos(\omega t)$$

General solution for the vibration of the string The general solution for the displacement of the string, which is a combination of all the modes, is thus an infinite sum of these modes:

$$y(x, t) = \sum_{n=1}^{\infty} [A_n \cos(\omega_n t)] \sin\left(\frac{n\pi}{L}x\right)$$

where the coefficients A_n is determined by the initial displacement.

Modes of vibration The angular frequency is related to the frequency and gives us all the superposing modes in the final waveform. This tells us what the fundamental frequency and the harmonics will be. [\[7\]](#) [\[8\]](#)

$$w = 2\pi f = \frac{n\pi c}{L}$$

The fundamental wavelength of vibration corresponds to the string having two vibration nodes at the fixed points; it oscillates in its basic mode with the length of the string corresponding to half the wavelength. Hence, the fundamental frequency of vibration is given by:

$$f_1 = \frac{1}{2L} \sqrt{\frac{T}{\mu}}$$

Where:

- L is the length of the string,
- T is the tension in the string,
- μ is the linear mass density.

The harmonics of this fundamental will be $n * f_1$. The harmonics we will hear depend on where we pluck the string.

Conclusion D'Alembert's wave equation for transverse vibrations of a plucked string describes the string's motion under tension. The fundamental frequency of vibration is determined by the tension in the string, its linear mass density, and the length of the string. This theory is widely used to understand the behavior of musical strings and other systems involving elastic waves. It is an easy model to try to compute.

Parameters for a nylon string B3 (247 Hz)

[9] [10] [11] [12]

- Length of the string: $L = 0.65 \text{ m}$
- Linear mass density: $\mu = 0.000582 \text{ kg/m}$
- tension in the string: $T = 60 \text{ N}$

Using the wave speed formula for a string and the fundamental frequency calculation found before, we calculate the wave speed and fundamental:

$$c = 321 \text{ m/s}$$

$$f = 247 \text{ Hz}$$

0.5.4 Finite Difference Method

The theory

We use the finite difference method. In this method, derivatives and second derivatives are replaced by their estimates:

$$\frac{\partial u}{\partial x} \approx \frac{u(x + \delta_x, t) - u(x, t)}{\delta_x}, \quad (1)$$

$$\frac{\partial u}{\partial t} \approx \frac{u(x, t + \delta_t) - u(x, t)}{\delta_t}, \quad (2)$$

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u(x + \delta_x, t) - 2u(x, t) + u(x - \delta_x, t)}{\delta_x^2}, \quad (3)$$

$$\frac{\partial^2 u}{\partial t^2} \approx \frac{u(x, t + \delta_t) - 2u(x, t) + u(x, t - \delta_t)}{\delta_t^2}. \quad (4)$$

Here, δ_x and δ_t are tiny intervals in space and time, respectively.

We discretize the space in length x and time t , and rewrite D'Alembert's equation:

$$\frac{u(t + \delta_t, x) - 2u(t, x) + u(t - \delta_t, x)}{\delta_t^2} - c^2 \frac{u(t, x + \delta_x) - 2u(t, x) + u(t, x - \delta_x)}{\delta_x^2} = 0. \quad (5)$$

By setting:

$$\gamma = \frac{c^2 \delta_t^2}{\delta_x^2},$$

We get:

$$u(t + \delta_t, x) = \gamma(u(t, x - \delta_x) + u(t, x + \delta_x)) + 2(1 - \gamma)u(t, x) - u(t - \delta_t, x). \quad (4)$$

We divide the string of length L into $N - 1$ segments of length:

$$\delta_x = \frac{L}{N - 1},$$

and the observation time T into K intervals of duration:

$$\delta_t = \frac{T}{K - 1}.$$

We study vibrations at the corresponding points and instants:

$$t_k = (k - 1)\delta_t, \quad x_n = (n - 1)\delta_x, \quad u_{n,k} = u(x_n, t_k).$$

We rewrite the discretized D'Alembert equation:

$$\begin{aligned} \forall k = 2, \dots, K-1, \quad \forall n = 2, \dots, N-1, \\ u_{n,k+1} = \gamma(u_{n-1,k} + u_{n+1,k}) + 2(1-\gamma)u_{n,k} - u_{n,k-1}. \end{aligned} \quad (6)$$

Initial conditions For a guitar string, the initial deformation is due to the plucking of the string at a position x_p . This can be expressed as:

$$u(x, 0) = \begin{cases} h \cdot \frac{x}{x_p}, & \text{if } 0 \leq x \leq x_p, \\ h \cdot \frac{L-x}{L-x_p}, & \text{if } x_p \leq x \leq L, \end{cases}$$

Where h is the maximum Amplitude of the pluck, L is the length of the string, and x_p is the position of the pluck.

The initial velocity is zero everywhere:

$$\frac{\partial u}{\partial t}(x, 0) = 0, \quad \text{for all } x \in [0, L].$$

In discrete notation, the initial conditions become:

$$\begin{aligned} u_{i,0} &= \begin{cases} h \cdot \frac{x_i}{x_p}, & \text{if } 0 \leq x_i \leq x_p, \\ h \cdot \frac{L-x_i}{L-x_p}, & \text{if } x_p \leq x_i \leq L, \end{cases} \\ u_{i,1} &= u_{i,0}, \quad \text{for all } i. \end{aligned}$$

Boundary conditions For a guitar string fixed at both ends, the boundary conditions are:

$$u(0, t) = u(L, t) = 0, \quad \text{for all } t.$$

In discrete notation, this becomes:

$$u_{0,j} = u_{N,j} = 0, \quad \text{for all } j,$$

Where N is the number of discrete segments of the string.

The computation and the result

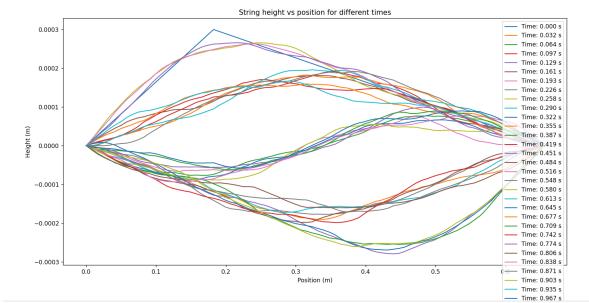


Figure 4: String at 0.18 m excited in time for a nylon string B3 (247 Hz) for one second with 80 space points

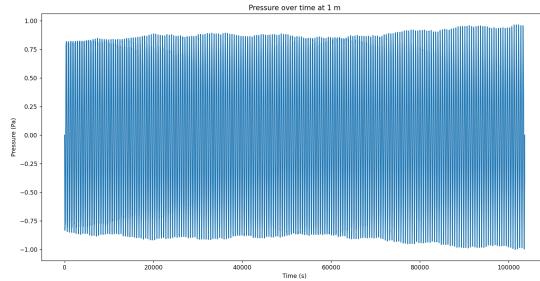


Figure 5: Pressure in time for a nylon string B3 (247 Hz) for one second with 80 space points

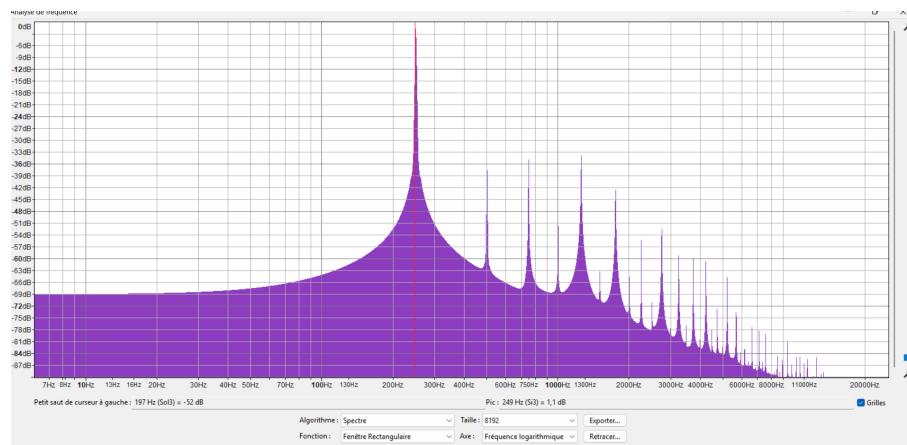


Figure 6: FFT of the signal of the guitar

When we analyze the sound of a plucked guitar string using an FFT (Fast Fourier Transform), we observe that the spectrum contains all the harmonics.

What is particularly fascinating is that the relative amplitudes of these harmonics depend on where the string is plucked. This phenomenon can be explained by the physics of standing waves and the resulting interaction between the fundamental frequency and its harmonics.

For example, if you pluck the string at half its length (i.e., at its midpoint), the even harmonics are notably absent.

This relationship can be understood by considering the Fourier series representation of the wave. The position where the string is plucked determines which harmonics are most effectively excited, as different positions correspond to different initial conditions for the sinusoidal components of the wave. In half the distance, the sinusoid is null for the even harmonics.

0.5.5 Finite Element Method (FEM)

Weak form of the wave equation The weak form of the wave equation is given as [4] [13] [14]:

$$M\ddot{U}(t) + KU(t) = 0$$

Where:

- M : Global mass matrix (from inertia term),
- K : Global stiffness matrix
- $U(t)$: Vector of nodal displacements,
- $\ddot{U}(t)$: Vector of nodal accelerations.

The goal is to compute U^{n+1} , the displacement at the next time step, given U^n (current displacements) and U^{n-1} (displacements at the previous time step).

Time discretization: To discretize the wave equation over time, we approximate the second derivative $\ddot{U}(t)$ using the central difference scheme:

$$\ddot{U} \approx \frac{U^{n+1} - 2U^n + U^{n-1}}{\Delta t^2}$$

Substitute this into the weak form:

$$M \frac{U^{n+1} - 2U^n + U^{n-1}}{\Delta t^2} + KU^n = 0$$

Rewriting:

$$\frac{M}{\Delta t^2} (U^{n+1} - 2U^n + U^{n-1}) + KU^n = 0$$

Simplify for U^{n+1} :

$$\frac{M}{\Delta t^2} U^{n+1} = \frac{M}{\Delta t^2} \cdot 2U^n - \frac{M}{\Delta t^2} \cdot U^{n-1} - KU^n$$

$$U^{n+1} = 2U^n - U^{n-1} - \Delta t^2 M^{-1} KU^n$$

Interpretation of the terms

- $2U^n$: The current displacement contributes the most to the next time step.
- $-U^{n-1}$: The inertia (past displacement) opposes changes.
- $-\Delta t^2 M^{-1} KU^n$: The stiffness and mass interaction modifies the displacement over time.

Step-by-step computation

1. Assemble the global mass and stiffness matrices:

- M : Comes from the weak form of the inertia term.
- K : Comes from the weak form of the elastic term.

These are computed for all elements and assembled into the global system.

2. Boundary conditions:

Apply boundary conditions to M and K , such as fixing the ends of the string (displacement = 0).

3. Iterative update:

Use the update formula:

$$U^{n+1} = 2U^n - U^{n-1} - \Delta t^2 M^{-1} KU^n$$

Compute U^{n+1} at every time step, starting with initial conditions:

- U^0 : Initial displacement (e.g., triangular shape from the pluck),
- U^1 : First step, often approximated as $U^1 = U^0$.

Local mass matrix and stiffness matrix derivation To compute the local mass matrix (M_{local}) and local stiffness matrix (K_{local}) for a single finite element in a 1D string using the FEM, follow these steps:

Shape functions For a 1D element with linear shape functions, we define:

$$N_1(x) = \frac{x_{e+1} - x}{\Delta x}, \quad N_2(x) = \frac{x - x_e}{\Delta x}$$

Where:

- x_e and x_{e+1} are the positions of the two nodes of the element,
- $\Delta x = x_{e+1} - x_e$ is the length of the element.

Mass matrix (M_{local}) The mass matrix is derived from the term:

$$\int_{x_e}^{x_{e+1}} \rho N_i(x) N_j(x) dx$$

For a two-node linear element ($i, j = 1, 2$), the result is:

$$M_{\text{local}} = \frac{\rho \Delta x}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

- The diagonal terms (2) represent the inertia contribution of each node.
- The off-diagonal terms (1) represent the coupling between the two nodes.

Stiffness Matrix (K_{local}) The stiffness matrix is derived from the term [4] [13] [14]:

$$\int_{x_e}^{x_{e+1}} T \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} dx$$

For linear elements, the derivatives of the shape functions are constants:

$$\frac{\partial N_1}{\partial x} = -\frac{1}{\Delta x}, \quad \frac{\partial N_2}{\partial x} = \frac{1}{\Delta x}$$

Substitute these into the integral:

$$K_{\text{local}} = \frac{T}{\Delta x} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

- The diagonal terms (1) represent the stiffness contribution at each node.
- The off-diagonal terms (-1) represent the coupling between the two nodes.

Global matrices assembly The global mass matrix (M) and global stiffness matrix (K) are assembled by summing the contributions from all elements. Each local matrix is added to the global matrix by mapping its nodes (degrees of freedom) to the global indices.

Applying boundary conditions For a string that is fixed at both ends ($u(0, t) = u(L, t) = 0$), the boundary conditions are applied as follows:

- Modify the global mass matrix (M) and stiffness matrix (K) such that the rows and columns corresponding to the boundary nodes are set to zero.
- Set the diagonal entries for the boundary nodes to 1, ensuring their displacements remain zero.

In practice, in python notation:

$$M[0, :] = 0, \quad M[-1, :] = 0, \quad M[0, 0] = 1, \quad M[-1, -1] = 1$$

$$K[0, :] = 0, \quad K[-1, :] = 0, \quad K[0, 0] = 1, \quad K[-1, -1] = 1$$

This enforces the fixed displacement at the boundaries while keeping the system consistent. We can now put this in a Python code and simulate it.

0.6 Comparing FEM and FDM guitar string

With the implementation of both the FEM and the FDM in Python, the objective is to conduct a comparative analysis of the two approaches using various performance metrics.

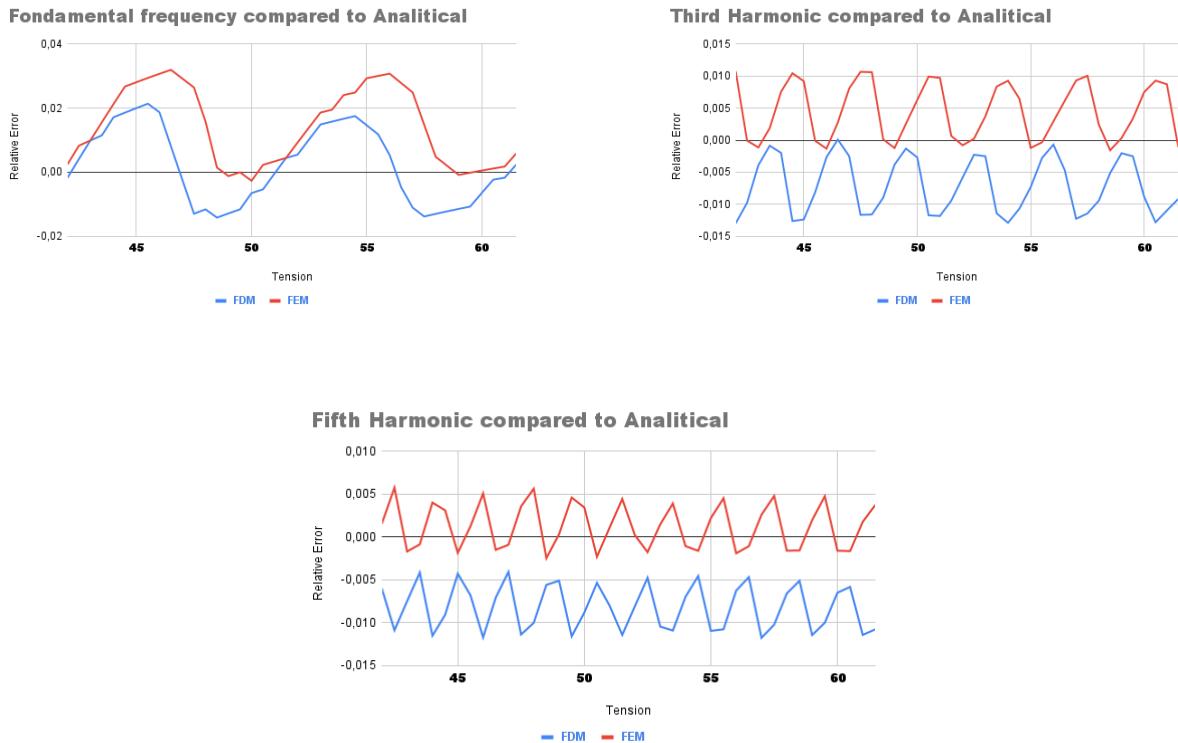
By systematically varying different parameters for each model (as detailed in the following sections), the waveform was computed for both FEM and FDM. Additionally, the expected theoretical frequency was calculated to serve as a reference. Using Audacity, I retrieved the given fundamental and harmonics for each waveform. I always used a rectangular FFT of 4096 in size. The relative error of the frequencies for each method was then computed and plotted across different scenarios, allowing for an evaluation of the accuracy and reliability of FEM and FDM in approximating the expected results.

0.6.1 Tension

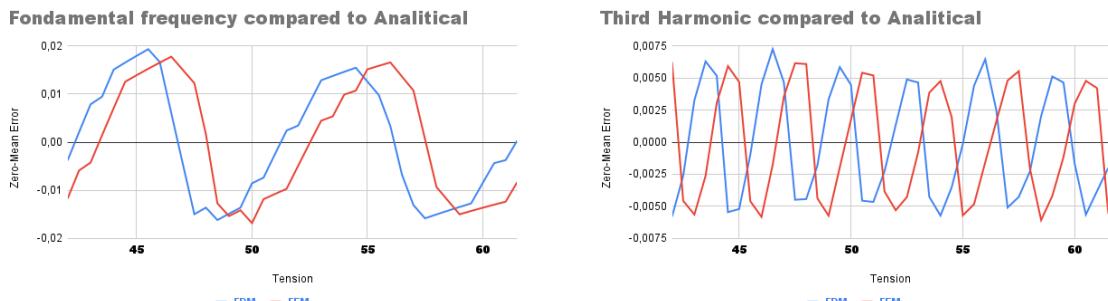
The tension is directly related to the frequency through the speed. I fixed all other parameters.

- **Length of the string:** $L = 0.655$ m
- **Number of points on the string:** $N = 80$
- **time interval:** $\Delta t = 1 \times 10^{-5}$ s
- **Number of time points:** $N_t = 1 \times 10^5$
- **Total simulation time:** $T = 1$ s
- **Amplitude:** $A = 3 \times 10^{-4}$ m
- **Linear mass density:** $\mu = 4.300172754 \times 10^{-4}$ kg/m

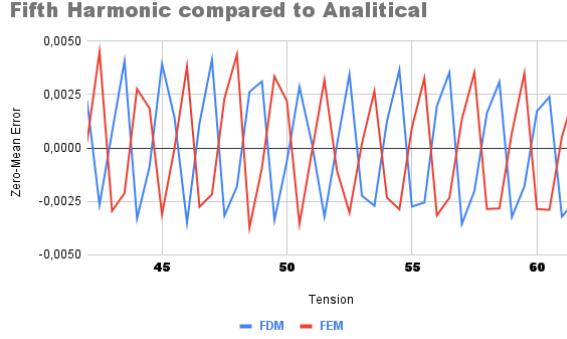
I computed 40 waveforms for 40 different tension values between 42 and 61.5 every 0.5 step. The resolution is 2.5 Hz, and we played the guitar at half its length at 0.3275 m. Playing at the location: we only have odd harmonics.



Zero-mean error We can see a sort of periodicity and that the two methods follow the same curves. To see this, let us center the curves around 0. I plotted the Zero-mean error.



[!tbp]



We can see now that both models follow the same error variation with phase shifts.

The periodicity of this error is very particular. A relationship between the harmonic and the fundamental is to be seen.

Analysis of harmonics and periodicity Here, the fundamental frequency f_1 has a tension periodicity of 9.5 N corresponding to 25 Hz; we examine the higher harmonics:

- $3f_1$: Periodicity 3.10 N or 8 Hz.
- $5f_1$: Periodicity 1.9 N or 5 Hz.

By computing the periodicity ratios:

$$\frac{9.5}{3} = 3.17, \quad \frac{9.5}{5} = 1.9 \quad (7)$$

$$\frac{25}{3} = 8.3, \quad \frac{25}{5} = 5 \quad (8)$$

We observe a clear relationship between the harmonic order and the periodicity of the error.

Interpretation of results These periodicity errors arise due to the discretization in time. Since we sample the frequency domain at discrete intervals, numerical artifacts manifest at harmonic positions. This behavior suggests that the discretization process influences the periodic detection of frequency variations in a structured manner. I believe that maybe this is also because we are doing an FFT. However, this is only an interpretation; further analysis might be needed.

0.6.2 Time interval

In numerical simulations, the choice of time step precision significantly impacts the accuracy of results. During one second, I decided to decrease the time interval and see how that affected the error.

The parameters are fixed to:

- **tension:** 42.86 N
- **Length:** 0.655 m
- **Number of points on the string:** 80
- **Linear mass density:** 4.30×10^{-4} kg/m

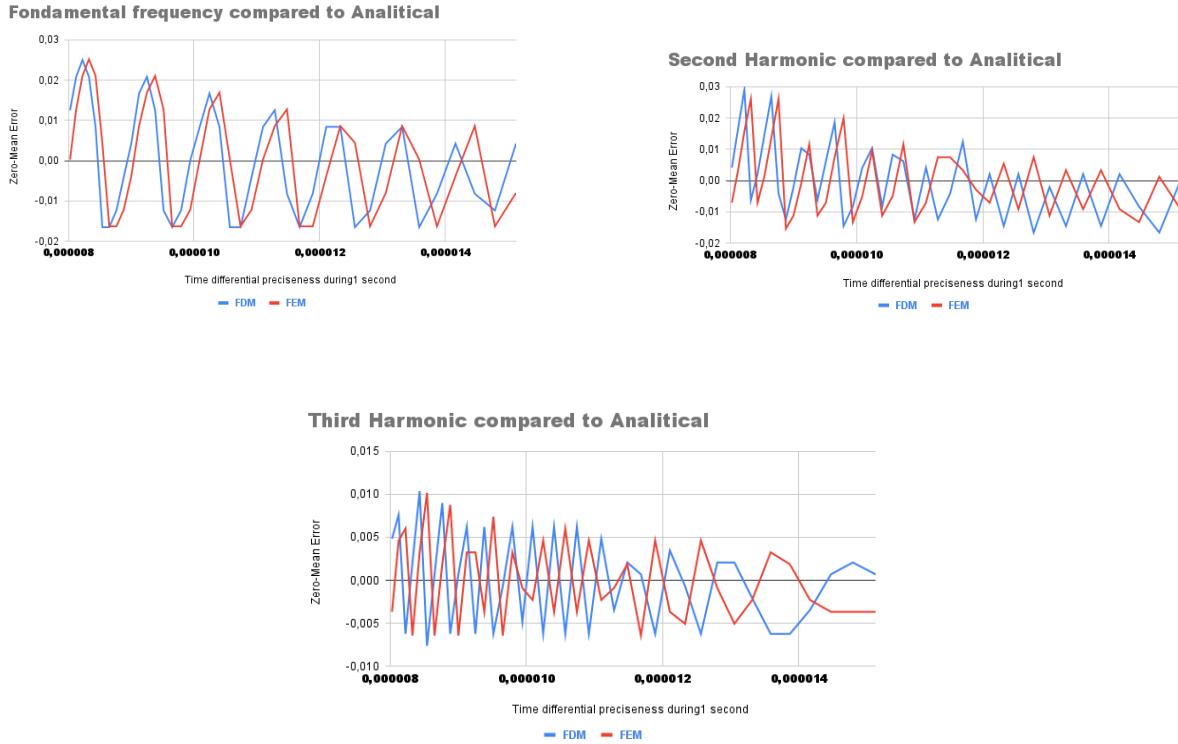
A comparison of time step values reveals that when using the time interval in the range of 2.27×10^{-5} to 1.52×10^{-5} seconds:

- FDM provides results.
- FEM diverges, yielding NaN values.

This discrepancy suggests that FEM requires a higher level of time precision than FDM to achieve comparable accuracy. The divergence of FEM at lower precision indicates its sensitivity to time discretization, emphasizing the need for finer resolution in FEM-based simulations.

But let us now compare FEM and FDM for non-NaN values.

The data step of 6.7×10^{-4} was repeated 40 times, and we played at a location of 0.18 m. We thus have all the harmonics, but I only measured the frequency of the first three frequencies. I then plotted the zero mean error of the frequencies.



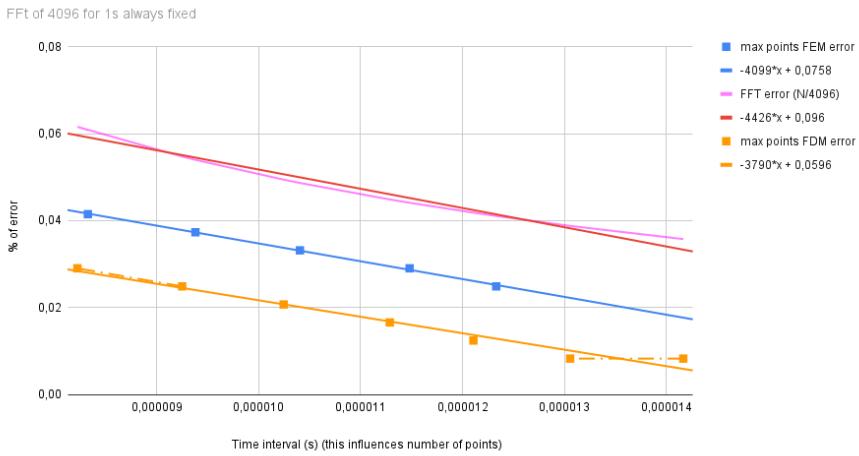
We can see once again that both FEM and FDM exhibit the same evolution in error precision. However, the smaller the time interval, the more points we have in time, and the larger the error becomes. This is not what one would intuitively expect.

This phenomenon is actually due to the way I am measuring the FFT. My window size remains the same even as the sampling frequency, directly linked to the time interval, increases. As a result, the error in the FFT measurement also increases.

This can be observed below. I compared the evolution of the fundamental error precision due to the FFT as the sampling frequency increases with the evolution of the error increase (maximum of the peaks) for FEM and FDM. They follow the same trend. This explains the counterintuitive result.

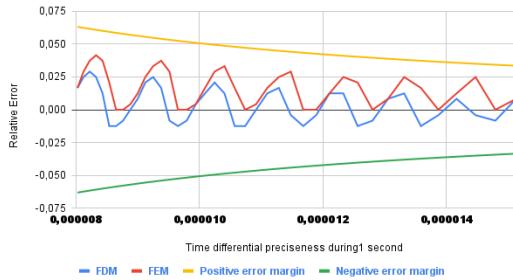
It is essential to realize that if I wanted to reduce the error in the FFT, I would need to use a larger window size when increasing the sampling frequency, corresponding to reducing the time interval.

Fundamental error FFT frequency interval compared to Amplitude of errors according to a time interval (increase in time points)

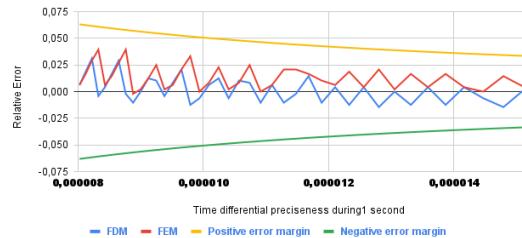


Now, comparing in a broader perspective with only the error, not the zero mean error:

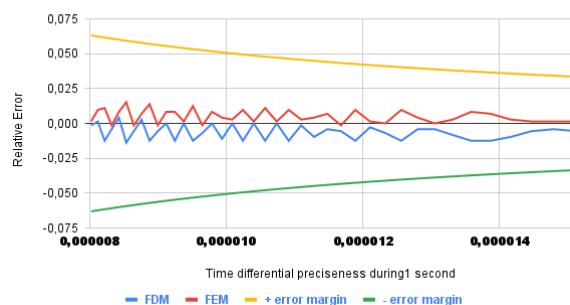
Fondamental frequency compared to Analytical



Second Harmonic compared to Analytical



Third Harmonic compared to Analytical



This showed that both FEM and FDM follow the same error variation, but we can see that FDM converges for a smaller time interval and that the error max is smaller for FDM for the fundamental and the second harmonic.

0.6.3 Number of nodes (points)

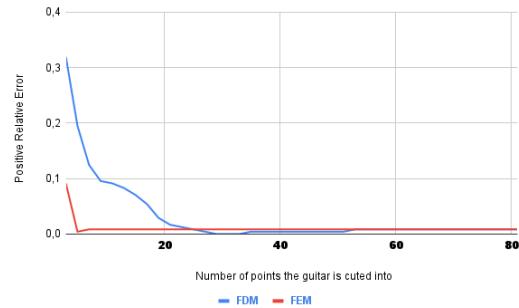
When doing FEM or FDM, you have several nodes you can create; for FEM, you can have a minimum of 1 element (2 nodes), and for FDM, you can also have 2 points. However, one element or two points would not create any sound in our case since both extreme points are fixed. Thus, we can cut our guitar strings into at least three points until we have as many as we wish. Here, we see how the number of points affects the error. The fixed parameters are:

- **tension:** 42.86 N

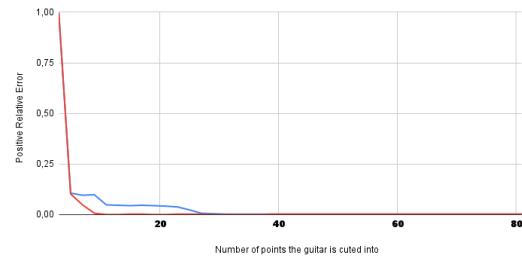
- **Length:** 0.655 m
- **time Interval:** $1.0 \times 10^{-5} \text{ s}$
- **Number of points in time:** 1.0×10^5
- **time of wave simulated:** 1.0 s
- **Amplitude:** $3.0 \times 10^{-4} \text{ m}$
- **Linear mass density:** $4.30 \times 10^{-4} \text{ kg/m}$

We continue to play at the position of the time interval simulations.

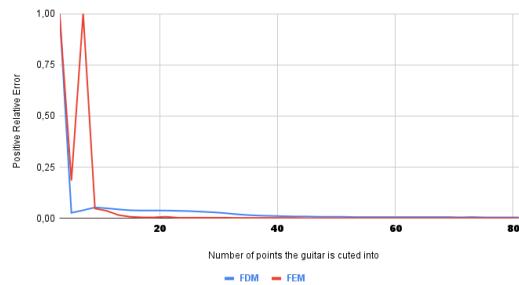
Fondamental compared to Analytical



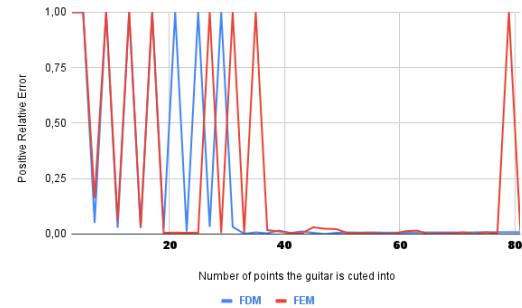
Second Harmonic compared to Analytical



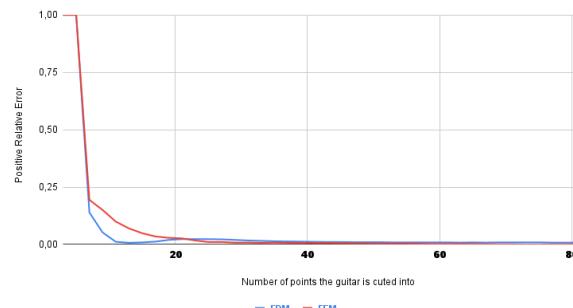
Third Harmonic compared to Analytical



Fourth Harmonic compared to Analytical



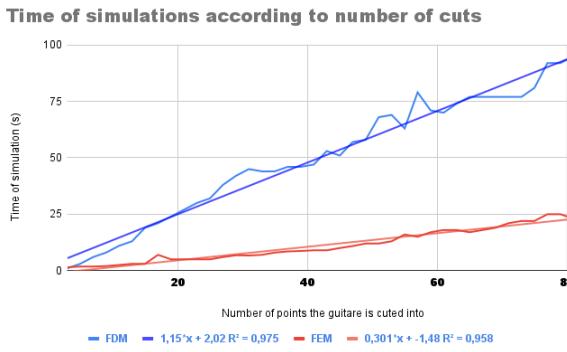
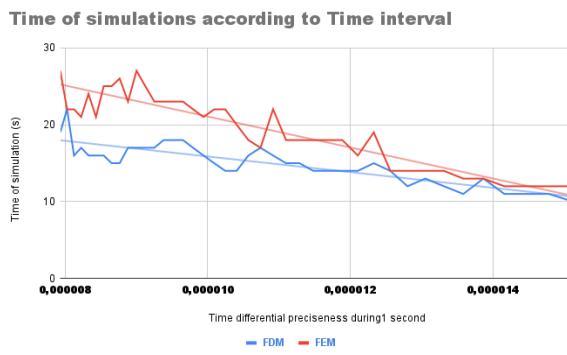
Fifth Harmonic compared to Analytical



When we see an error that is equal to one, it means that we did not detect any frequency. We can see that the fundamental FEM converges faster than FDM for the fundamental and the second harmonic. However, after 40 points, the error is equivalent, and they both converge to the same mistake for most harmonics. The fourth was hard to detect because it had a small amplitude, but we can still see some convergence after 40 points were tested.

0.6.4 Time of simulation

I have made two Python codes, FEM and FDM. On the same computer, I ran them and measured the time of simulation each time I did the previous experiences. It is important to note that this data depends on the computer and how we code it in Python. When considering my analysis, please remember that this is not representative of every scenario.



Conclusion on the time The tension, thus the frequency, does not influence the time intuitively. However, we can see that FDM is slowly getting slower and slower according to tension. However, FDM is still faster than the FEM in those conditions.

For the time interval, since the smaller the time interval, the more data we have to calculate, the result follows our intuition: the more significant the step, the quicker it is. But we still see that FDM is quicker than FEM.

As the number of points increases, we observe that the simulation time for FEM is significantly shorter compared to FDM. This can be explained by the way the code has been implemented. In the case of FDM, a Python loop iterates over all points, whereas in FEM, computations are performed using a complex matrix product and summation directly on the points. Consequently, FEM is expected to be computationally more efficient.

0.7 To compare to reality: guitar with damping and without the modulus of elasticity

D'Alembert equation with fluid damping [15] [11]

The D'Alembert equation for transverse vibrations of a string with fluid damping is given by:

$$\mu \frac{\partial^2 u}{\partial t^2} - T \frac{\partial^2 u}{\partial x^2} + \sigma \frac{\partial u}{\partial t} = 0, \quad (9)$$

Where:

- $u(x, t)$: Transverse displacement of the string at position x and time t ,
- $c = \sqrt{\frac{T}{\mu}}$: Wave velocity in m/s, dependent on the tension T in N and the linear mass density μ in kg/m,
- σ : Fluid damping coefficient, accounting for air friction. kg/s

This equation includes a damping term $\sigma \frac{\partial u}{\partial t}$, which represents the resistance due to air friction. This term is dominant at low frequencies.

FDM with damping We discretize the string in space and time:

- The spatial domain $[0, L]$ is divided into N points with spacing $\Delta x = \frac{L}{N}$.
- The temporal domain $[0, \tau]$ is divided into K points with spacing $\Delta t = \frac{\tau}{K}$.

Using finite differences, we approximate:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{n+1,k} - 2u_{n,k} + u_{n-1,k}}{\Delta x^2}, \quad (10)$$

$$\frac{\partial^2 u}{\partial t^2} \approx \frac{u_{n,k+1} - 2u_{n,k} + u_{n,k-1}}{\Delta t^2}, \quad (11)$$

$$\frac{\partial u}{\partial t} \approx \frac{u_{n,k+1} - u_{n,k-1}}{2\Delta t}. \quad (12)$$

Substituting these into the D'Alembert equation yields:

$$\mu \frac{u_{n,k+1} - 2u_{n,k} + u_{n,k-1}}{\Delta t^2} - T \frac{u_{n+1,k} - 2u_{n,k} + u_{n-1,k}}{\Delta x^2} + \sigma \frac{u_{n,k+1} - u_{n,k-1}}{2\Delta t} = 0$$

Rearranging terms:

$$u_{n,k+1} \left(\mu \frac{1}{\Delta t^2} + \frac{\sigma}{2\Delta t} \right) = \frac{T}{\Delta x^2} (u_{n+1,k} + u_{n-1,k}) + \left(2\mu \frac{1}{\Delta t^2} - 2 \frac{T}{\Delta x^2} \right) u_{n,k} + u_{n,k-1} \left(-\mu \frac{1}{\Delta t^2} + \frac{\sigma}{2\Delta t} \right)$$

We start with the equation:

$$u_{n,k+1} \left(\mu \frac{1}{\Delta t^2} + \frac{\sigma}{2\Delta t} \right) = \frac{T}{\Delta x^2} (u_{n+1,k} + u_{n-1,k}) + \left(2\mu \frac{1}{\Delta t^2} - 2 \frac{T}{\Delta x^2} \right) u_{n,k} + u_{n,k-1} \left(-\mu \frac{1}{\Delta t^2} + \frac{\sigma}{2\Delta t} \right).$$

To isolate $u_{n,k+1}$, divide through by $\mu \frac{1}{\Delta t^2} + \frac{\sigma}{2\Delta t}$:

$$u_{n,k+1} = \frac{\frac{T}{\Delta x^2} (u_{n+1,k} + u_{n-1,k}) + \left(2\mu \frac{1}{\Delta t^2} - 2 \frac{T}{\Delta x^2} \right) u_{n,k} + u_{n,k-1} \left(-\mu \frac{1}{\Delta t^2} + \frac{\sigma}{2\Delta t} \right)}{\mu \frac{1}{\Delta t^2} + \frac{\sigma}{2\Delta t}}.$$

Define the coefficients:

$$\gamma = \frac{T^2 \Delta t^2}{\Delta x^2}, \quad (13)$$

$$\alpha = \mu + \frac{\sigma \Delta t}{2}, \quad (14)$$

$$\theta = -\mu + \frac{\sigma \Delta t}{2}. \quad (15)$$

The equation becomes:

$$u_{n,k+1} = \frac{\gamma \frac{1}{\Delta t^2} (u_{n+1,k} + u_{n-1,k}) + \left(2 \frac{\mu}{\Delta t^2} - 2\gamma \frac{1}{\Delta t^2}\right) u_{n,k} + u_{n,k-1} \left(\theta \frac{1}{\Delta t^2}\right)}{\alpha \frac{1}{\Delta t^2}}.$$

Thus, the final equation is:

$$u_{n,k+1} = \frac{\gamma}{\alpha} (u_{n-1,k} + u_{n+1,k}) + \frac{2(\mu - \gamma)}{\alpha} u_{n,k} + \frac{\theta}{\alpha} u_{n,k-1}. \quad (16)$$

Boundary and initial conditions

- **Boundary conditions:** The ends of the string are fixed:

$$u(0, t) = u(L, t) = 0, \quad \forall t. \quad (17)$$

- **Initial conditions:**

- The string is initially triangular:

$$u(n, 0) = \begin{cases} U_0 \frac{n}{n_p}, & 0 \leq n \leq n_p, \\ U_0 \frac{N-n}{N-n_p}, & n_p \leq n \leq N, \end{cases} \quad (18)$$

where n_p is the plucking position.

- The initial velocity is zero everywhere:

$$\frac{\partial u}{\partial t}(n, 0) = 0. \quad (19)$$

0.7.1 Comparison to Reality

We are looking at a real guitar. This study compares our Finite Difference Method (FDM) simulation results to real-world conditions. The following physical parameters characterize the experimental setup, which we will use as the basis for our simulations:

- **Fundamental frequency:**

$$f = 246.9 \text{ Hz}$$

(corresponding to the musical note B3), determined using a Fast Fourier Transform (FFT) with a window size 4096.

- **String dimensions:**

- **Length:**

$$L = 65.5 \text{ cm} = 0.655 \text{ m}$$

- **Diameter:**

$$d = 0.69 \text{ mm} = 6.9 \times 10^{-4} \text{ m}$$

- **Linear mass density:**

$$\mu = 1150 \times \left(\frac{d}{2}\right)^2 \pi = 43.0017 \times 10^{-5} \text{ kg/m} = 4.300 \times 10^{-4} \text{ kg/m}$$

- **Playing position:** The string is plucked at

$$0.180 \text{ m}$$

- **Theoretical tension:**

$$T = 45.02 \text{ N}$$

These parameters will serve as the reference for validating our numerical simulations against experimental data. [9] [10]



Figure 7: Guitar Parameters

The simulation was performed under the following conditions:

- **Number of points:**
80 points
- **Playing position:**
22nd point
- **Initial Amplitude:**
 $0.3 \text{ mm} = 3.0 \times 10^{-4} \text{ m}$
- **Simulation duration:**
3 s

- **Time step:**
 $\Delta t = 9.65 \times 10^{-6} \text{ s}$
- **Damping**
 $\sigma = 0.0013 \text{ kg/s}$

(Chosen to match the time of damping of the real guitar)

To determine the appropriate values for our simulation for string tension and time step (Δt) in the simulation, we analyzed the error introduced by different choices:

- At $T = 45 \text{ N}$, the frequency error is +2%.
- At $\Delta t = 96.53 \times 10^{-7} \text{ s}$, the frequency error is -1.2%.

These parameters were chosen to balance accuracy in the numerical simulation while minimizing errors compared to the experimental conditions.

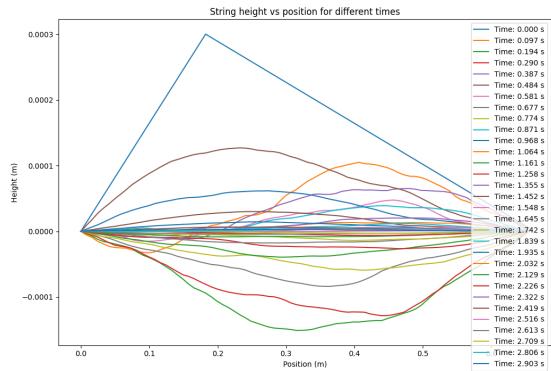


Figure 8: Simulation result of this particular string

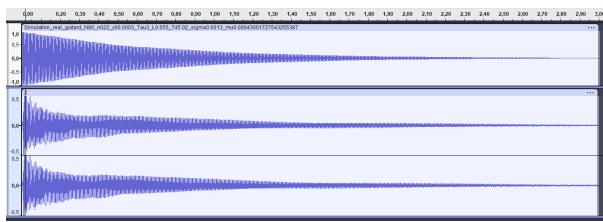


Figure 9: Damping looked at on Audacity, the upper waveform is the simulation and under is the real signal

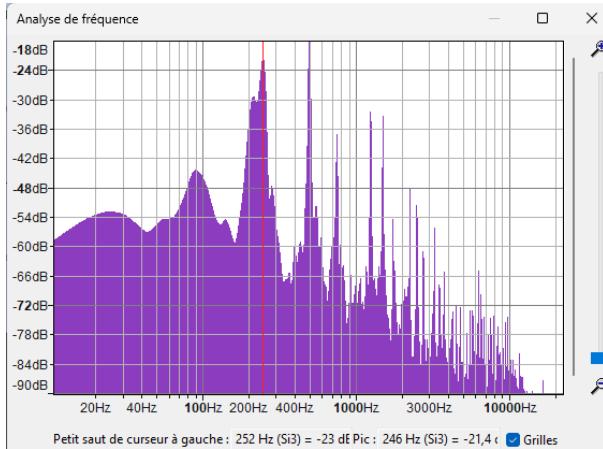


Figure 10: FTT of the real signal

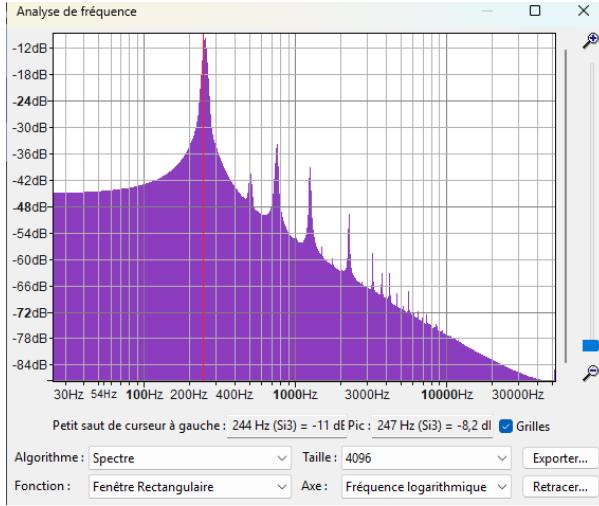


Figure 11: FFT on the simulated waveform

Conclusion In the frequency domain, the simulated signal closely matches the real signal, demonstrating the accuracy of the numerical Model. However, it is essential to note that perceptually, the simulated sound still exhibits noticeable differences from the real instrument. Specifically, the simulation accurately represents the vibrating string. Still, it does not account for the reflections from surrounding surfaces or the acoustic filtering effects of the guitar body, which significantly contribute to the instrument's characteristic timbre. The timbre is not represented in the simulation. However, if the goal is solely to identify the type of string, the simulation can still achieve this.

0.7.2 Important note on the simulation

When performing a FFT analysis, it is crucial to clearly define the sampling rate and the sampling window, as errors can arise from these parameters. The accuracy of the FFT is inherently limited by the resolution of the discrete data and the chosen windowing function.

Moreover, achieving higher numerical precision in the simulation does not necessarily translate to increased accuracy in practical measurements. In reality, experimental data acquisition is subject to various sources of uncertainty, including instrument limitations and environmental noise. Additionally, the FFT introduces a loss of precision due to spectral leakage and finite resolution. Therefore, while an exact numerical approach may be beneficial for theoretical analysis, it is essential to acknowledge the inherent limitations of real-world measurements.

0.8 The cycling bell with damping and modulus of elasticity

0.8.1 The models

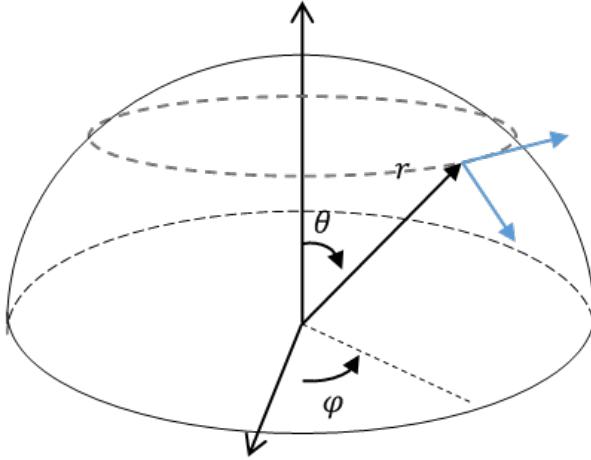


Figure 12: The model of cycling bell

We modeled the bell as the shell of a hemisphere with radius R , positioned relative to its center in spherical coordinates (r, θ, φ) as shown in the figure. We have a fixed point at the top.

We also assume that the shell of the bell is very thin (thickness $h \ll R$), so for every point on the shell, we can write: $r \approx R$.

0.8.2 Theory

As I said previously, we did not take into account the flexion in the d'Alembert equation. I said it was a negligible term. However, that term is helpful for the cycling bell as it is mainly made of metal. [16]

The fundamental equation of dynamics with damping with elasticity is expressed as follows:

$$D\nabla^4 u + \sigma \frac{\partial u}{\partial t} + \rho h \frac{\partial^2 u}{\partial t^2} = 0 \quad (20)$$

Where:

- $D = \frac{Eh^3}{12(1-\nu^2)}$: the bending rigidity, $N * m$
- h : thickness, m
- ρ : density, kg/m^3
- E : Young's modulus, $Pa = N/m^2$
- ν : Poisson's ratio,
- ∇^4 : the biharmonic operator (double Laplacian).
- σ : damping $N/(s * m^2)$

0.8.3 Analytical resolution

We seek solutions of the form:

$$u(\theta, \varphi, t) = u(\theta, \varphi)e^{i\omega t}. \quad (21)$$

Substituting into the equation gives:

$$\nabla^4 u - \left(\omega^2 \frac{\rho h}{D} - i\omega \frac{\sigma}{D} \right) u(\theta, \varphi) = 0 \quad (22)$$

Or equivalently:

$$(\nabla^4 - k_\omega^4) u(\theta, \varphi) = 0 \quad (23)$$

Where:

$$k_\omega^4 = \omega^2 \frac{\rho h}{D} - i\omega \frac{\sigma}{D}. \quad (24)$$

But we can write:

$$(\nabla^4 - k_\omega^4) = (\nabla^2 + k_\omega^2)(\nabla^2 - k_\omega^2), \quad (25)$$

The solution can be split into two components:

$$u(\theta, \varphi) = u_+(\theta, \varphi) + u_-(\theta, \varphi),$$

With:

$$(\nabla^2 + k_\omega^2)u_+(\theta, \varphi) = 0, \quad (26)$$

$$(\nabla^2 - k_\omega^2)u_-(\theta, \varphi) = 0. \quad (27)$$

In the spherical coordinate system, the Laplacian is expressed as:

$$\nabla^2 u = \Delta u = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial u}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial u}{\partial \theta} \right) + \frac{1}{r^2 (\sin \theta)^2} \frac{\partial^2 u}{\partial \varphi^2}. \quad (28)$$

Since the shell is very thin, we can assume that r is constant and equal to R . Thus, the waves described by equation (22) propagate mainly along the bell's surface, with variations in θ and φ : $u = u(\theta, \varphi, t)$.

$$\frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial u}{\partial \theta} \right) = \frac{1}{r^2 \sin \theta} \left(\cos \theta \frac{\partial u}{\partial \theta} + \sin \theta \frac{\partial^2 u}{\partial \theta^2} \right) = \frac{1}{r^2} \left(\cot \theta \frac{\partial u}{\partial \theta} + \frac{\partial^2 u}{\partial \theta^2} \right) \quad (29)$$

Having $r = R = \text{constant}$, the Laplacian becomes:

$$\nabla^2 u = \frac{1}{R^2} \left(\cot \theta \frac{\partial u}{\partial \theta} + \frac{\partial^2 u}{\partial \theta^2} + \frac{1}{(\sin \theta)^2} \frac{\partial^2 u}{\partial \varphi^2} \right) \quad (30)$$

We seek the solutions by separating variables:

$$u(\theta, \varphi) = w(\theta) \cdot h(\varphi) \quad (31)$$

We write:

$$(\nabla^2 + k_\omega^2) w(\theta) \cdot h(\varphi) = 0 \quad (32)$$

$$\left(\sin \theta \cos \theta \frac{\partial w}{\partial \theta} + (\sin \theta)^2 \frac{\partial^2 w}{\partial \theta^2} \right) \frac{1}{w(\theta)} + (\sin \theta)^2 R^2 k_\omega^2 = -\frac{\partial^2 h}{\partial \varphi^2} \frac{1}{h(\varphi)} \quad (33)$$

In this equation, the term on the right-hand side depends only on φ , and the term on the left-hand side depends only on θ . This implies that each term must equal a constant λ . We thus obtain:

$$\begin{cases} \left(\sin \theta \cos \theta \frac{\partial w_+}{\partial \theta} + (\sin \theta)^2 \frac{\partial^2 w_-}{\partial \theta^2} \right) \frac{1}{w_+} + (\sin \theta)^2 R^2 k_\omega^2 = \lambda \\ -\partial^2 \frac{h_+}{\partial \varphi^2} \frac{1}{h_+} = \lambda \end{cases} \quad (34)$$

We can rewrite it:

$$\begin{cases} (\sin \theta)^2 \partial^2 \frac{w_+}{\partial \theta^2} + \sin \theta \cos \theta \partial \frac{w_+}{\partial \theta} + \left((\sin \theta)^2 R^2 k_\omega^2 - \lambda \right) \cdot w_+ = 0 \\ \lambda h_+ + \partial^2 \frac{h_+}{\partial \varphi^2} = 0 \end{cases} \quad (35)$$

Or:

$$\begin{cases} \partial^2 \frac{w_+}{\partial \theta^2} + \cot \theta \partial \frac{w_+}{\partial \theta} + \left(R^2 k_\omega^2 - \frac{\lambda}{(\sin \theta)^2} \right) \cdot w_+(\theta) = 0 \\ \lambda h_+ + \partial^2 \frac{h_+}{\partial \varphi^2} = 0 \end{cases} \quad (36)$$

We would also get:

$$\begin{cases} \partial^2 \frac{w_-}{\partial \theta^2} + \cot \theta \partial \frac{w_-}{\partial \theta} + \left(-R^2 k_\omega^2 - \frac{\lambda}{(\sin \theta)^2} \right) \cdot w_-(\theta) = 0 \\ \lambda h_- + \frac{\partial^2 h_-}{\partial \varphi^2} = 0 \end{cases} \quad (37)$$

We need to remember these equations:

$$\varphi \in [0; \pi/2] \cup (2k\pi) \quad (38)$$

$$\theta \in \left[0; \frac{\pi}{2} \right] \quad (39)$$

The solutions to the second equality in (36) and (37) are exponential functions with:

$$h_+(\varphi) = h_{0+} e^{\pm i \sqrt{\lambda} \varphi} \quad (40)$$

$$h_-(\varphi) = h_{0-} e^{\pm i \sqrt{\lambda} \varphi} \quad (41)$$

Physically, the functions h_+ and h_- are 2π periodic. This means that $\sqrt{\lambda}$ must be an integer, i.e., $\sqrt{\lambda} = m \in \mathbf{Z}$.

We can write by setting: $\Omega^2 = R^2 k_\omega^2$

The periodicity in $h(\varphi) = h(\varphi + 2\pi)$ implies that m must be an integer.

Now, we need to solve the equation (for both w_+ and w_-):

$$\frac{\partial^2 w(\theta)}{\partial \theta^2} + \cot \theta \frac{\partial w(\theta)}{\partial \theta} + \left(\pm \Omega^2 - \frac{\lambda}{(\sin \theta)^2} \right) w(\theta) = 0 \quad (42)$$

We can change the variable to find solutions to this equation (42).

By setting $x = \cos \theta$, we can show that the equation can be rewritten with $x \in [0; 1]$ as:

$$(1 - x^2) \frac{\partial^2 w(x)}{\partial x^2} - 2x \frac{\partial w(x)}{\partial x} + \left(\pm \Omega^2 - \frac{m^2}{1 - x^2} \right) w(x) = 0 \quad (43)$$

This Equation is called the "associated Legendre equation".

We can rewrite the equation (43) by seeking a solution in the form of:

$$w(x) = (1 - x^2)^{\frac{m}{2}} p(x) \quad (44)$$

In this case, the equation is rewritten with $x \in [0; 1]$ as:

$$(1 - x^2) \frac{\partial^2 p(x)}{\partial x^2} - 2x(m+1) \frac{\partial p(x)}{\partial x} + (\pm \Omega^2 - m(m+1)) p(x) = 0 \quad (45)$$

We seek the solution $p(x)$ in the form of a convergent power series on $x \in [0; 1]$:

$$p(x) = \sum_{n \geq 0} a_n x^n \quad (46)$$

$$\frac{\partial p(x)}{\partial x} = \sum_{n \geq 0} a_{n+1} (n+1) x^n \quad (47)$$

$$\frac{\partial^2 p(x)}{\partial x^2} = \sum_{n \geq 0} a_{n+2} (n+2)(n+1) x^n \quad (48)$$

Substituting into the previous equation, we get:

$$\sum_{n \geq 0} a_{n+2} (n+2)(n+1)x^n - \sum_{n \geq 0} a_{n+2} (n+2)(n+1)x^{n+2} \quad (49)$$

$$- \sum_{n \geq 0} a_{n+1} (n+1)(m+1)x^{n+1} + \sum_{n \geq 0} a_n (\pm \Omega^2 - m(m+1))x^n = 0 \quad (50)$$

This simplifies to:

$$\sum_{n \geq 0} a_{n+2} (n+2)(n+1)x^n - \sum_{n \geq 2} a_n (n)(n-1)x^n - \sum_{n \geq 1} a_n (m+1)x^n + \sum_{n \geq 0} a_n (\pm \Omega^2 - m(m+1))x^n = 0 \quad (51)$$

For $n \geq 2$, the coefficients of x^n are:

$$a_{n+2} (n+2)(n+1) - a_n \cdot (n(n+m) + m(m+1) \mp \Omega^2) \quad (52)$$

And the first two coefficients are given by for x^0 and x^1 (we try in $x = 0$ and by recurrence with the derivative at $x = 0$):

$$2a_2 + a_0 (\pm \Omega^2 - m(m+1)) \quad (53)$$

$$6a_3 + a_1 (\pm \Omega^2 - (m+1)^2) \quad (54)$$

Thus, to ensure that all those coefficients are zero, we must have the following:

$$a_2 = \frac{1}{2} (m(m+1) \mp \Omega^2) \cdot a_0 \quad (55)$$

$$a_3 = \frac{1}{6} ((m+1)^2 \mp \Omega^2) \cdot a_1 \quad (56)$$

By substituting the derivatives of $p(x)$ into the equation (45) and by thinking through recurrence, we can write:

$$a_{n+2} = a_n \frac{n^2 + n(2m+1) + m(m+1) \mp \Omega^2}{(n+2)(n+1)} \quad (57)$$

First case

To avoid divergence of the series (even when $x = 1$), it is required that all the coefficients are zero from a certain rank forward.

This means that the values of Ω that are suitable must satisfy for some n_0 :

$$\Omega = \pm \sqrt{n_0^2 + n_0(2m+1) + m(m+1)} \quad (58)$$

We can also see that there is a complex solution:

$$\Omega = \pm i \sqrt{n_0^2 + n_0(2m+1) + m(m+1)} \quad (59)$$

It should be noticed that the recurrence relation between the coefficients of the power series is of order 2, and for all the coefficients to be zero from a certain rank onwards, all coefficients must have the same parity as n_0 .

By setting $k = n_0 + m$, this is rewritten as: Ω can be either

$$\Omega = \pm \sqrt{k(k+1)} \quad (60)$$

or

$$\Omega = \pm i \sqrt{k(k+1)} \quad (61)$$

Second case

We seek convergence for all $x \leq 1$. D'Alembert's rule says convergence for all $x < 1$. But at $x = 1$ we need to use:

Rule Raabe-Duhamel for $x = 1$

$$\left| \frac{a_2(k+1)}{a_2 k} \right| = 1 - \frac{2-m}{k} + o\left(\frac{1}{k}\right)$$

Thus, to avoid divergence, we will have convergence if $m = 0$; however, when that is the case, $w(x) = p(x)$. With the condition at the limit, the top is fixed $p(1) = 0$.

Conjecture: The only case where that condition is verified is when all the terms are null after a certain rank.

We can verify this numerically: We can see that the function only passes by 0 for the values of Ω where we have a finite sum. I checked that all the values of Ω where the plot passed by zero is the case where Ω follows the first case seen before where all the terms are null after a rank n_0 .

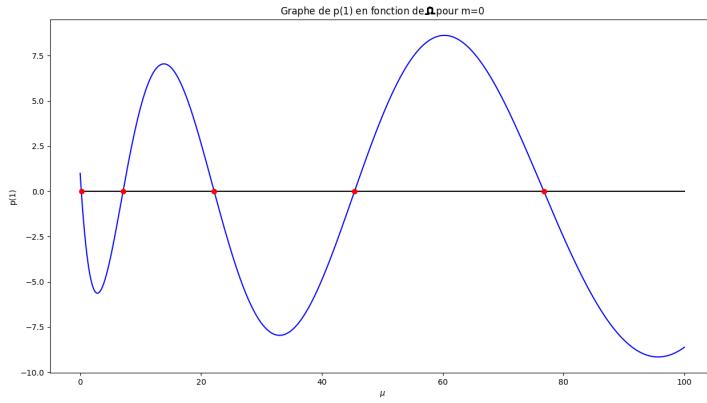


Figure 13: $p(1)$ according to Ω in $m = 0$

The frequency solutions are either way, because $i^4 = 1$, $\Omega^4 = (Rk_\omega)^4$:

$$\omega^2 \frac{\rho h R^4}{D} - i\omega \frac{\sigma R^4}{D} - k^2 (k+1)^2 = 0 \quad (62)$$

This means:

$$\omega = \frac{1}{2\frac{\rho h R^4}{D}} \left(i \frac{\sigma R^4}{D} \pm \sqrt{-\left(\frac{\sigma R^4}{D}\right)^2 + 4\frac{\rho h R^4}{D} k^2 (k+1)^2} \right) \quad (63)$$

That is to say:

$$\omega = \left(i \frac{\sigma}{2\rho h} \pm \frac{1}{2\frac{\rho h R^4}{D}} \sqrt{-\left(\frac{\sigma R^4}{D}\right)^2 + 4\frac{\rho h R^4}{D} k^2 (k+1)^2} \right) \quad (64)$$

Thus:

$$\omega = \left(i \frac{\sigma}{2\rho h} \pm \sqrt{-\left(\frac{\sigma}{2\rho h}\right)^2 + \frac{D}{\rho h R^4} k^2 (k+1)^2} \right) \quad (65)$$

We thus have an attenuation term $e^{\frac{-\sigma}{2\rho h} t}$ and a harmonic term $e^{2i\pi f_k t}$ with:

$$f_k = \frac{1}{2\pi} \sqrt{-\left(\frac{\sigma}{2\rho h}\right)^2 + \frac{D}{\rho h R^4} k^2 (k+1)^2} \quad (66)$$

We have the characteristic quantities of the material and the geometry of the shell:

- The bending rigidity $D = \frac{Eh^3}{12(1-\nu^2)}$ [Pa · m³]
- R is the radius of the shell [m]
- h is the thickness [m]
- ρ is the density [kg/m³]
- E is Young's modulus [N/m²]
- ν is Poisson's ratio [dimensionless]
- σ is the damping

We calculate two characteristic coefficients of the material and the geometry of the shell:

- The attenuation coefficient $\gamma = \frac{\sigma}{2\rho h}$
- The stiffness coefficient $\alpha = \frac{D}{\rho h R^4}$

Next, we choose two integers m_0 and n_0 that determine the vibration mode.

For each pair (m_0, n_0) chosen, we determine the coefficient $k_0 = n_0 + m_0$ which allows the calculation of the vibration frequency:

$$f_{k_0} = \frac{1}{2\pi} \sqrt{-\gamma^2 + \alpha k_0^2 (k_0 + 1)^2} \quad (67)$$

The vibration mode is written as follows:

$$u(\theta, \varphi, t) = u_{(m_0, n_0)}(\theta, \varphi) e^{\frac{-\sigma}{2\rho h} t} e^{2i\pi f_{k_0} t} \quad (68)$$

The elongations are written for the chosen m and n_0 as:

$$u_{(m_0, n_0)} = w_{(m_0, n_0)}(\theta) \cdot h_{m_0}(\varphi) \quad (69)$$

With:

$$h_{m_0}(\varphi) = h_0 e^{im_0 \varphi} \quad (70)$$

$$w_{(m_0, n_0)}(\theta) = (\sin \theta)^{\frac{m_0}{2}} \cdot p_{n_0}(\cos \theta) \quad (71)$$

$$p_{n_0}(x) = a_0 + a_1 x^1 + a_2 x^2 + \dots + a_{n_0} x^{n_0} \quad (72)$$

To calculate the coefficients of $p_{n_0}(x)$, we first calculate:

$$\Omega_0 = \sqrt{k_0 (k_0 + 1)} \quad (73)$$

If n_0 is even, all coefficients a_n with odd n are zero.

If n_0 is odd, all coefficients a_n with even n are zero.

Next, for n varying from 0 to n_0 , we calculate:

$$a_{n+2} = a_n \frac{n^2 + n(2m_0 + 1) + m_0(m_0 + 1) - \Omega_0^2}{(n+2)(n+1)} \quad (74)$$

We start with a_0 for the even coefficients.

We start with a_1 for the odd coefficients.

Due to the recurrence relation between the coefficients, we notice that all even coefficients are proportional to a_0 and all odd coefficients are proportional to a_1 .

Thus, the coefficients can be adjusted based on the boundary conditions.

Parameters for numerical application:

$$\sigma = 10, \text{kg} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$$

$$R = 0.05, \text{m}$$

$$h = 0.001, \text{m}$$

Material	Density [kg/m ³]	Young's Modulus E [GPa]	Poisson's Ratio ν
Steel	7850	210	0.24 to 0.30
Aluminum	2700	62	0.24 to 0.33
Copper	8920	128	0.33
Brass (70% Cu + 30% Zn)	8470	80 to 100	0.37

Numerical test

Using Matlab, we could compute the sound the bell would do in theory by superposing all the harmonics with the proper amplitudes.



Figure 14: Example of the result of the waveform simulated

0.8.4 Comparing analytical to reality

Important information Before comparing our mathematical model to reality, we can see the numerical frequency is related to the damping term:

$$f_k = \frac{1}{2\pi} \sqrt{-\left(\frac{\sigma}{2\rho h}\right)^2 + \frac{D}{\rho h R^4} k^2 (k+1)^2} \quad (75)$$

However, for a cycling bell in particular, that is not perfect; it is important to realize that there are no given damping terms. Analyzing the effect of the term damping on the frequency, I realized that it had minimal impact on the frequency. We can see this in the figure below.

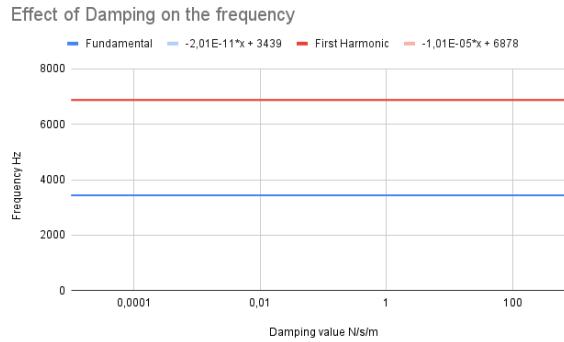


Figure 15: Damping effect on the fundamental frequency

We can now compare it to reality without carrying about the Damping. I fixed it at $10 \text{ kg} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$ to match the damping speed of a real life cycling bell that I tested.

Comparing to reality I took a big aluminum bell.



Figure 16: Aluminium bell

The Parameters were:

- **Damping coefficient (fixed arbitrarily):**

$$\eta = 1.00 \times 10^1 \text{ kg} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$$

- **Density:**

$$\rho = 2.70 \times 10^3 \text{ kg} \cdot \text{m}^{-3}$$

- **Radius:**

$$R = 4.00 \times 10^{-2} \text{ m}$$

- **Thickness:**

$$h = 8.00 \times 10^{-4} \text{ m}$$

- **Young's modulus:**

$$E = 6.20 \times 10^{10} \text{ N} \cdot \text{m}^{-2}$$

- **Poisson's ratio:**

$$\nu = 3.00 \times 10^{-1}$$

Cycling Bell	f_1 (Hz)	f_2 (Hz)	f_3 (Hz)	f_4 (Hz)
Model (Matlab)	6.890×10^2	1.385×10^3	2.305×10^3	3.462×10^3
Reality	6.880×10^2	1.742×10^3	3.080×10^3	3.502×10^3
Math	6.924×10^2	1.385×10^3	2.308×10^3	3.462×10^3

Table 1: Comparison of fundamental frequencies (f_n) between Matlab simulations, real measurements, and mathematical calculations for aluminium.

We can see that the FFT greatly influences the model used in Matlab. The code of Matlab directly corresponds to the mathematical frequency; thus, it should give the same result. The window is still at 4096; we are losing information and precision. But we are also doing the same FFT in reality. Thus, we have the same error margin. We also find almost the same frequency result. When we listen to the audio file, we realize that we have the same sound in our ears; this shows that the mathematical Model is close to reality. I also tried with another bell, which I thought was made of brass, but realized that the heard and mathematical frequencies were very different (more than 1000 Hz difference). Knowing that the model was correct showed that the cycling bell was not made of brass. This model could be used to check the material of a given object.

0.8.5 Finite Difference Method

D'Alembert's equation in a solid material is generally expressed as:

$$\Delta u - \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2} = 0 \quad (76)$$

We assume that the mechanical vibrations in the material obey D'Alembert's wave equation. This is not an unreasonable assumption since it is known to hold for waves in fluids, i.e., longitudinal waves.

However, a bell obeys the plate equation, which is expressed as:

$$\frac{\partial^2 u}{\partial t^2} = -\frac{D}{\rho h} \nabla^4 u$$

Where:

- The flexural rigidity $D = \frac{Eh^3}{12(1-\nu^2)}$,
- h : thickness,
- ρ : density,
- E : Young's modulus,
- ν : Poisson's ratio,
- ∇^4 : biharmonic operator (double Laplacian).

In spherical coordinates, the Laplacian is expressed as:

$$\Delta u = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial u}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial u}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 u}{\partial \varphi^2} \quad (77)$$

Since the shell is very thin, we can consider r as constant and equal to R . Thus, the waves described by equation (76) primarily propagate along the surface of the bell, with variations in θ and φ : $u = u(\theta, \varphi, t)$. Further, simplifications lead to:

$$\begin{aligned} \nabla^4 u &= \frac{\partial^4 u}{\partial \theta^4} + 2 \cot \theta \frac{\partial^3 u}{\partial \theta^3} + \left(\frac{\cos^2 \theta - 2}{\sin^2 \theta} \right) \frac{\partial^2 u}{\partial \theta^2} + \frac{\cot \theta}{\sin^2 \theta} \frac{\partial u}{\partial \theta} \\ &\quad + \frac{\partial^4 u}{\partial \varphi^4} + 2 \frac{1 + \cos^2 \theta}{\sin^2 \theta} \frac{\partial^2 u}{\partial \varphi^2}. \end{aligned}$$

Boundary conditions are defined as follows:

- At the point $\theta = 0$, vibration is always zero:

$$\forall t \forall \varphi u(0, \varphi, t) = 0$$

- Along the great circle $\theta = \frac{\pi}{2}$, vibration is free along θ :

$$\forall t \forall \varphi \frac{\partial u}{\partial \theta}(0, \varphi, t) = 0$$

By setting $r = R = \text{constant}$, the Laplacian becomes:

$$\frac{1}{R^2} \left(\cot \theta \frac{\partial u}{\partial \theta} + \frac{\partial^2 u}{\partial \theta^2} + \frac{1}{(\sin \theta)^2} \frac{\partial^2 u}{\partial \varphi^2} \right)$$

For the double Laplacian, we can factor out $\frac{1}{R^4}$. This results in 9 terms:

$$\cot \theta \frac{\partial}{\partial \theta} \left(\cot \theta \frac{\partial u}{\partial \theta} \right) = \frac{-\cot \theta}{(\sin \theta)^2} \frac{\partial u}{\partial \theta} + (\cot \theta)^2 \frac{\partial^2 u}{\partial \theta^2}$$

$$\cot \theta \frac{\partial}{\partial \theta} \left(\frac{\partial^2 u}{\partial \theta^2} \right) = \cot \theta \frac{\partial^3 u}{\partial \theta^3}$$

$$\cot \theta \frac{\partial}{\partial \theta} \left(\frac{1}{(\sin \theta)^2} \frac{\partial^2 u}{\partial \varphi^2} \right) = -2 \frac{(\cot \theta)^2}{(\sin \theta)^2} \frac{\partial^2 u}{\partial \varphi^2} + \frac{\cot \theta}{(\sin \theta)^2} \frac{\partial^3 u}{\partial \theta \partial \varphi^2}$$

$$\frac{\partial^2}{\partial \theta^2} \left(\cot \theta \frac{\partial u}{\partial \theta} \right) = \frac{2 \cot \theta}{(\sin \theta)^2} \frac{\partial u}{\partial \theta} + 2 \frac{-1}{(\sin \theta)^2} \frac{\partial^2 u}{\partial \theta^2} + \cot \theta \frac{\partial^3 u}{\partial \theta^3}$$

$$\frac{\partial^2}{\partial \theta^2} \left(\frac{\partial^2 u}{\partial \varphi^2} \right) = \frac{\partial^4 u}{\partial \theta^4}$$

$$\begin{aligned} \frac{\partial^2}{\partial \theta^2} \left(\frac{1}{(\sin \theta)^2} \frac{\partial^2 u}{\partial \varphi^2} \right) &= \left(\frac{2}{(\sin \theta)^4} + 4 \frac{(\cot \theta)^2}{(\sin \theta)^2} \right) \frac{\partial^2 u}{\partial \varphi^2} + \frac{-4 \cot \theta}{(\sin \theta)^2} \frac{\partial^3 u}{\partial \theta \partial \varphi^2} + \frac{1}{(\sin \theta)^2} \frac{\partial^4 u}{\partial \varphi^2 \partial \theta^2} \\ \frac{1}{(\sin \theta)^2} \frac{\partial^2}{\partial \varphi^2} \left(\cot \theta \frac{\partial u}{\partial \theta} \right) &= \frac{\cot \theta}{(\sin \theta)^2} \frac{\partial^3 u}{\partial \theta \partial \varphi^2} \\ \frac{1}{(\sin \theta)^2} \frac{\partial^2}{\partial \varphi^2} \left(\frac{\partial^2 u}{\partial \theta^2} \right) &= \frac{1}{(\sin \theta)^2} \frac{\partial^4 u}{\partial \varphi^2 \partial \theta^2} \\ \frac{1}{(\sin \theta)^2} \frac{\partial^2}{\partial \varphi^2} \left(\frac{1}{(\sin \theta)^2} \frac{\partial^2 u}{\partial \varphi^2} \right) &= \frac{1}{(\sin \theta)^4} \frac{\partial^4 u}{\partial \varphi^4} \end{aligned}$$

We thus obtain the following terms:

$$\begin{aligned} \nabla^4 u &= \frac{\partial^4 u}{\partial \theta^4} + \frac{1}{(\sin \theta)^4} \frac{\partial^4 u}{\partial \varphi^4} + \left(\frac{2}{(\sin \theta)^2} \right) \frac{\partial^4 u}{\partial \varphi^2 \partial \theta^2} - 2 \frac{\cot \theta}{(\sin \theta)^2} \frac{\partial^3 u}{\partial \theta \partial \varphi^2} + 2 \cot \theta \frac{\partial^3 u}{\partial \theta^3} \\ &\quad + \left((\cot \theta)^2 + \frac{-2}{(\sin \theta)^2} \right) \frac{\partial^2 u}{\partial \theta^2} + \left(\frac{2}{(\sin \theta)^4} + 2 \frac{(\cot \theta)^2}{(\sin \theta)^2} \right) \frac{\partial^2 u}{\partial \varphi^2} + \frac{\cot \theta}{(\sin \theta)^2} \frac{\partial u}{\partial \theta} \end{aligned}$$

This means:

$$\begin{aligned} \nabla^4 u &= \frac{\partial^4 u}{\partial \theta^4} + \frac{\partial^4 u}{(\sin \theta \partial \varphi)^4} + 2 \frac{\partial^4 u}{(\sin \theta \partial \varphi)^2 \partial \theta^2} - 2 \cot \theta \frac{\partial^3 u}{\partial \theta (\sin \theta \partial \varphi)^2} + 2 \cot \theta \frac{\partial^3 u}{\partial \theta^3} + \left(\frac{(\cos \theta)^2 - 2}{(\sin \theta)^2} \right) \frac{\partial^2 u}{\partial \theta^2} \\ &\quad + 2 \left(\frac{1 + (\cos \theta)^2}{(\sin \theta)^2} \right) \frac{\partial^2 u}{(\sin \theta \partial \varphi)^2} + \frac{\cot \theta}{(\sin \theta)^2} \frac{\partial u}{\partial \theta} \end{aligned}$$

Even:

$$\begin{aligned} \nabla^4 u &= \frac{\partial^4 u}{\partial \theta^4} + 2 \cot \theta \frac{\partial^3 u}{\partial \theta^3} + \left(\frac{(\cos \theta)^2 - 2}{(\sin \theta)^2} \right) \frac{\partial^2 u}{\partial \theta^2} + \frac{\cot \theta}{(\sin \theta)^2} \frac{\partial u}{\partial \theta} \\ &\quad + \frac{\partial^4 u}{(\sin \theta \partial \varphi)^4} + 2 \left(\frac{1 + (\cos \theta)^2}{(\sin \theta)^2} \right) \frac{\partial^2 u}{(\sin \theta \partial \varphi)^2} \\ &\quad + 2 \frac{\partial^4 u}{(\sin \theta \partial \varphi)^2 \partial \theta^2} - 2 \cot \theta \frac{\partial^3 u}{\partial \theta (\sin \theta \partial \varphi)^2} \end{aligned}$$

Remember: $u(\theta, \varphi)$:

$$\varphi \in [0; 2\pi[\quad (+2k\pi)$$

$$\theta \in \left[0; \frac{\pi}{2} \right]$$

We can cut $u(\theta, \varphi) = u(n, k)$ and we get:

$$\frac{\partial u}{\partial \theta}(n, k) \cong \frac{1}{2\delta\theta} (u(n+1, k) - u(n-1, k))$$

$$\frac{\partial^2 u}{\partial \theta^2}(n, k) \cong \frac{1}{\delta\theta^2} (u(n+1, k) - 2u(n, k) + u(n-1, k))$$

$$\begin{aligned}
\frac{\partial^2 u}{\partial \varphi^2}(n, k) &\cong \frac{1}{\delta \varphi^2} (u(n, k+1) - 2u(n, k) + u(n, k-1)) \\
\frac{\partial^3 u}{\partial \theta^3}(n, k) &\cong \frac{3}{8\delta \theta^3} (u(n+2, k) - u(n-2, k) - 2(u(n+1, k) - u(n-1, k))) \\
\frac{\partial^4 u}{\partial \theta^4}(n, k) &\cong \frac{1}{\delta \theta^4} (u(n+2, k) + u(n-2, k) - 4u(n+1, k) - 4u(n-1, k) + 6u(n, k)) \\
\frac{\partial^4 u}{\partial \varphi^4}(n, k) &\cong \frac{1}{\delta \varphi^4} (u(n, k+2) + u(n, k-2) - 4u(n, k+1) - 4u(n, k-1) + 6u(n, k)) \\
\left(\frac{\partial^3 u}{\partial \varphi \partial \theta^2} \right)(n, k) &\cong \frac{1}{2\delta \varphi \cdot \delta \theta^2} (u(n+1, k+1) - 2u(n, k+1) \\
&+ u(n-1, k+1) - u(n+1, k-1) + 2u(n, k-1) - u(n-1, k-1)) \\
\left(\frac{\partial^3 u}{\partial \theta \partial \varphi^2} \right)(n, k) &\cong \frac{1}{2\delta \varphi^2 \delta \theta} (u(n+1, k+1) - 2u(n+1, k) \\
&+ u(n+1, k-1) - u(n-1, k+1) + 2u(n-1, k) - u(n-1, k-1)) \\
\left(\frac{\partial^4 u}{\partial \theta^2 \partial \varphi^2} \right)(n, k) &\cong \frac{1}{\delta \theta^2} \left(\frac{\partial^2 u}{\partial \varphi^2}(n+1, k) - 2\frac{\partial^2 u}{\partial \varphi^2}(n, k) + \frac{\partial^2 u}{\partial \varphi^2}(n-1, k) \right) \\
&\cong \frac{1}{\delta \varphi^2 \delta \theta^2} (u(n+1, k+1) - 2u(n, k+1) + u(n-1, k+1) \\
&- \frac{2}{\delta \varphi^2 \delta \theta^2} (u(n+1, k) - 2u(n, k) + u(n-1, k)) \\
&+ \frac{1}{\delta \varphi^2 \delta \theta^2} (u(n+1, k-1) - 2u(n, k-1) + u(n-1, k-1)))
\end{aligned}$$

To consider the time, we need to add the damping term:

$$\rho h \frac{\partial^2 u}{\partial t^2} = -\frac{D}{R^4} \nabla^4 u - \sigma \frac{\partial u}{\partial t}$$

And take into account.

$$\begin{aligned}
\frac{\partial u}{\partial t}(n, k, t) &\cong \frac{1}{2\delta t} (u(n, k, t+1) - u(n, k, t-1)) \\
\frac{\partial^2 u}{\partial t^2}(n, k, t) &\cong \frac{1}{\delta t^2} (u(n, k, t+1) - 2u(n, k, t) + u(n, k, t-1))
\end{aligned}$$

So:

$$\begin{aligned}
\frac{\rho h}{\delta t^2} (u(n, k, t+1) - 2u(n, k, t) + u(n, k, t-1)) + \frac{\sigma}{2\delta t} (u(n, k, t+1) - u(n, k, t-1)) &= -\frac{D}{R^4} \nabla^4 u(t) \\
\left(\frac{\rho h}{\delta t^2} + \frac{\sigma}{2\delta t} \right) u(n, k, t+1) &= -\frac{D}{R^4} \nabla^4 u(n, k, t) + 2 \frac{\rho h}{\delta t^2} u(n, k, t) + \left(-\frac{\rho h}{\delta t^2} + \frac{\sigma}{2\delta t} \right) u(n, k, t-1)
\end{aligned}$$

For the equations at the limits, we keep that at the top we are fixed.

- At $\theta = 0$ the vibration is null: $\forall t \quad \forall \varphi \quad u(0, \varphi, t) = 0$
- On the big circle $\theta = \frac{\pi}{2}$ the vibration is free, following θ : $\forall t \quad \forall \varphi \quad \frac{\partial u}{\partial \theta}(\frac{\pi}{2}, \varphi, t) = 0$

The assumption at $\theta = 0$ is easy to understand. The mechanical fixation prevents any vibration. The assumption at $\theta = \frac{\pi}{2}$ is more difficult to grasp. It means that, at these points, the vibration does not vary concerning θ .

This would need to be computed in Python or Matlab. I tried both cases. Matlab's calculation was very long, and I needed to take very small steps. In Python, I did not finish the code in time because I had to check that all the calculations were correct before running the code, which was too long to compute. I managed to change how the matrices were formed to reduce the computation time, but I did not have more time to correct the code in more depth.

0.9 Conclusion

0.9.1 General conclusion

FEM and FDM In general, the measurement error is negligible since these models are primarily used to compare with real-world conditions, where the precision of measurements is inherently limited. When performing an FFT analysis using basic instruments, such as a standard microphone with a sampling rate of 44100 Hz, the accuracy of the frequency extraction is constrained.

It is also noteworthy that computation time depends significantly on the implementation. From my observations, the computational time for both methods is approximately equivalent. While the FDM is generally faster, the FEM tends to be more efficient when handling a higher number of points. Additionally, FDM converges for larger time intervals compared to FEM, indicating that FDM reaches a stable solution more quickly than FEM.

From a mathematical and implementation perspective, FDM is more straightforward but requires knowledge of the object's shape, necessitating recalculations for different geometries. In contrast, while FEM is conceptually more complex to understand and implement, once the initial derivations are established, the object type and boundary conditions are incorporated into the stiffness (K) and mass (M) matrices. This allows for a more generalized approach, where a single FEM algorithm can be adapted for different objects simply by modifying these matrices.

Furthermore, given the availability of open-source FEM software, it is evident that in an industrial context, FEM would be more cost-effective and flexible. Developing a single general FEM algorithm and adapting it by modifying the K and M matrices would be more efficient for large-scale applications requiring frequent modifications. Conversely, FDM remains the more practical choice for smaller and more specialized applications due to its relative simplicity and ease of implementation.

Models It is important to note that, through my work, I have demonstrated that both the models, modeling the physical propagation of the wave, for the guitar string and the cycling bell are accurate and effectively represent reality.

0.9.2 Additional work possible

Many things can be done to continue this project.

- Compare the complexity of FEM and FDM
- Finish the Python code of the FDM of the Bell
- Compute the FEM of the bell on python or on an online FEM application [1]
- Compare the frequencies of the string of a guitar if you add the modulus of elasticity and show that it is not important
- Explore other models like BEM: Boundary element method

Bibliography

- [1] Y. Konaté, “Calcul par éléments finis des structures en coque,” Tech. Rep., 1986, A study on shell structure modeling using the finite element method, analyzing theoretical aspects and computational implementation.
- [2] E. Kaselouris, M. Bakarezos, M. Tatarakis, N. A. Papadogiannis, and V. Dimitriou, “A review of finite element studies in string musical instruments,” *Acoustics*, 2022, A comprehensive review of FEM modeling applied to string musical instruments, covering soundboards, resonance boxes, and fluid-structure interactions.
- [3] M. Raibaud, “Modélisation et simulation de systèmes discréétisés par la méthode des éléments finis dans le formalisme des systèmes hamiltoniens portuels : Application à la synthèse sonore,” Tech. Rep., 2018, A master’s thesis on the modeling and simulation of nonlinear infinite-dimensional systems using the finite element method and port Hamiltonian systems, with applications to sound synthesis.
- [4] V. Manet, *Méthode des éléments finis : Vulgarisation des aspects mathématiques et illustration de la méthode*. Creative Commons 3.0 France, 2013, This book presents a detailed theoretical overview of the finite element method, providing engineers with a solid mathematical foundation and practical applications.
- [5] Anonymous, “L’équation des ondes de d’alembert et ses applications,” *Physique des Ondes*, A study on the one-dimensional D’Alembert wave equation, its derivation, solutions, and applications in vibrating strings and mechanical waves.
- [6] Anonymous, “Correction dl4 : Corde vibrante et propagation des ondes,” Tech. Rep., A correction document discussing wave propagation in vibrating strings, including the mathematical formulation of the wave equation and boundary conditions.
- [7] A. Joussen, “De la guitare...,” *Bulletin de l’Union des Physiciens*, 1990, An acoustic study of the guitar detailing string vibrations, resonance, and the physical characteristics affecting sound production.
- [8] F. Martin, “Étude d’un instrument à cordes,” *Sciences et Vie Junior*, 1989, An educational exploration of the physics of string instruments, discussing string vibrations, sound wave propagation, and tuning techniques.
- [9] K. Polisano and M. Jedouaa, *Tp méthodes numériques : Modélisation d’une corde de guitare et du tympan*, This lab work explores the physical and numerical modeling of vibrational phenomena, particularly guitar strings, and shell., ENSIMAG, 2016.
- [10] Anonymous, “Td ondes mécaniques - vibrations des cordes et équation de d’alembert,” *TD Ondes Mécaniques*, 2017, A tutorial covering the equations governing vibrating strings, wave propagation, and stationary wave solutions, including the application of D’Alembert’s equation.
- [11] G. Derveaux, “Modélisation numérique de la guitare acoustique,” Doctoral thesis on the numerical modeling of acoustic guitars, including structural vibrations, coupling with air, and finite element methods for sound synthesis., Ph.D. dissertation, École Polytechnique, 2002.
- [12] Anonymous, *Physique appliquée : Étude des instruments à cordes*, A lab manual covering the study of the physics of string instruments, including the analysis of vibrations, harmonics, and sound propagation in guitars., 2023.
- [13] S. Deghboudj, *Méthode des éléments finis*. Université Larbi Tébessi - Tébessa, 2018, A comprehensive textbook on the finite element method, covering theoretical foundations and applications in mechanical and civil engineering.

- [14] S. Deghboudj, *Cours sur la méthode des éléments finis (livre complet)*. Édition universitaire, 2023, A book covering the fundamentals of the finite element method, intended for engineering students.
- [15] Y. Rocard, “Sur l’amortissement des ondes sonores dans un milieu gazeux homogène,” *Journal de Physique et le Radium*, 1930, 1 (12), pp.426-437. DOI: 10.1051/jphysrad:01930001012042600. A theoretical study on the damping of sound waves due to various effects such as diffusion, thermal conductivity, and viscosity in gaseous media.
- [16] G. Derveaux, “Étude sur les ondes acoustiques dans les systèmes complexes,” *Revue de Physique*, 2022, Theoretical and practical exploration of the acoustic properties of waves.