Thesis:

“Numerical simulation of a violin string motion”

supervisor: Stefano Miccoli
candidate: Marco Bellante
ABSTRACT
A physical model for a bowed violin string was developed. A non linear strain field and a friction force between bow and string dependent on the relative speed were considered. A discrete numerical model was coded in the high level programming language Matlab. The code was validated studying simple cases, for which scientific literature provides an analytical solution. Finally, the code was used to study the string's motion and the sound generated during a common excitation.

SOMMARIO
E' stato sviluppato un modello fisico di una corda di violino eccitata dall'archetto. Sono stati considerati una forza di attrito tra corda e archetto dipendente dalla velocità relativa e una deformazione non lineare. Un modello numerico discreto è stato implementato nel linguaggio di programmazione di alto livello Matlab. Il codice è stato validato studiando semplici casi, per i quali esistono soluzioni analitiche nella letteratura scientifica. Il codice è infine stato utilizzato per studiare il moto della corda di violino e il suono generato durante un'eccitazione comune.
# TABLE OF CONTENTS

1. INTRODUCTION .................................................................................................................. 5
   1.1. good sound.................................................................................................................. 5
   1.2. ...and playability...................................................................................................... 6
   1.3. The violin .................................................................................................................. 6
   1.4. The human ear.......................................................................................................... 8

2. QUALITATIVE DESCRIPTION ......................................................................................... 9

3. LITERATURE ................................................................................................................. 11
   3.2. Raman [4] ............................................................................................................... 11
   3.6. Latest numerical simulations .................................................................................. 14

4. IMPORTANT ASPECTS ................................................................................................. 15

5. PHYSICAL MODEL ....................................................................................................... 16
   5.1. Viscous damping .................................................................................................... 20
   5.2. Excitation ............................................................................................................... 22
   5.3. Forces evaluation ................................................................................................... 27

6. CONSIDERATIONS ....................................................................................................... 29
   6.1. Imposed motion trajectory ..................................................................................... 29
   6.2. Damping ................................................................................................................. 29
   6.3. Coding ................................................................................................................... 30

7. CODE VALIDATION ....................................................................................................... 31
   7.1. Fundamental harmonics ......................................................................................... 33
       7.1.1. 1\textsuperscript{st} fundamental harmonic ......................................................... 34
       7.1.2. 2\textsuperscript{nd} fundamental harmonic .............................................................. 43
       7.1.3. 3\textsuperscript{rd} fundamental harmonic .............................................................. 46
       7.1.4. 4\textsuperscript{th} fundamental harmonic .............................................................. 49
       7.1.5. 5\textsuperscript{th} fundamental harmonic .............................................................. 52
       7.1.6. 6\textsuperscript{th} fundamental harmonic .............................................................. 55
       7.1.7. 7\textsuperscript{th} fundamental harmonic .............................................................. 58
8. THE CODE ................................................................. 89

8.1. job_harmonic................................................................. 90
8.2. job_wave_propagation...................................................... 90
8.3. job_imposed_dof............................................................. 90
8.4. pre_processing............................................................... 91
8.5. string_data................................................................. 92
8.6. harmonic................................................................. 93
8.7. wave_propagation............................................................ 93
8.9. processing................................................................. 94
8.10. Integration............................................................... 96
8.11. bridge_force............................................................. 98
8.12. bow_force.............................................................. 98
8.13. post_processing........................................................... 99
8.15. signal_bellante............................................................ 104
8.16. film........................................................................... 105

9. CONCLUSIONS ..................................................................... 106

10. TABLE OF FIGURES .......................................................... 107

11. REFERENCES ...................................................................... 110
1. INTRODUCTION

The Mechanical Department of Politecnico di Milano has a partnership with the musical acoustics laboratory of the museum “Museo del Violino” in Cremona, and we need a model of the vibrating string for our activities. Link: http://www.museodelviolino.org/laboratori-scientifici/laboratorio-politecnico-di-milano/

One might ask:

- How can you judge a violin’s performance?
- What is a ‘good sound’?
- What does it mean that a violin is ‘easier to play’ than another one?

The violin case acts as a mechanical filter, transforming the string’s vibration into radiating sound, and its transfer function can be measured. One might think the quality of a violin depends on its transfer function, but, surprisingly, not only the cheapest violins on the market show a transfer function really similar to the best ones, but the transfer function is not the only feature to be considered. In fact, unlike other musical instruments, bowed instruments let you control the sound even after generating it, and some of them make this job easier. This may seem confusing, but let us consider a classical example. In a piano, when pushing a key, a hammer hits a string making it vibrate, and, after releasing the key, you are not able to control the sound anymore. You can just control the damping by releasing the key really slowly. In a violin, instead, you generate the sound thanks to the friction force acting between bow and string, and, while the sound is radiating, you can control it through the bow’s movement.

Let me quote Jim Woodhouse [1], a mathematician with a passion for violins:

1.1. good sound…

“Musical instruments raise a variety of problems of classical physics, materials science and engineering, and offer applications of many theoretical and experimental techniques, but the distinctive flavour of the subject comes from the fact that the key questions are posed by subjective judgements: what is ‘good sound’? This survey of violin physics must involve an excursion into perception and psychophysics, the science of giving quantitative form to subjective judgements.”
1.2. ...and playability

“A beginner on the guitar may play the wrong note, but the individual note will still be recognizably ‘musical’. A first attempt to bow a violin string, in contrast, may produce a very un-musical sound. This contrast points to the most fundamental difference in the physics of plucking a string versus bowing one: the plucked string is, to a good approximation, a linear system but a bowed string is strongly nonlinear. Plucking a string elicits a mixture of its natural frequencies, which are approximately harmonically spaced. All the player can do is vary the mixture by the method and position of the pluck. Bowing elicits a self-sustained oscillation through the slip-stick action of friction between the bow-hair and the string.”

1.3. The violin

Figure 1 violin

Sotakeit at the English language Wikipedia [GFDL (http://www.gnu.org/copyleft/fdl.html) or CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/3.0/)], from Wikimedia Commons
The frequency interval between the first natural frequency of a string and the first natural frequency of the next string is 7 semitones, that is:

$$f_{n+1} = 2^{7/12} f_n \ [Hz]$$
1.4. The human ear

The human ear is able to hear sounds from 20 Hz to 20 kHz, the upper limits tends to decrease with age. The average frequency resolution between 1 kHz and 2 kHz is about 3.6 Hz.

Figure 4 human ear


Figure 5 human hearing

By Human_Hearing_Graph.jpg: Daxx4434 derivative work: Dani CS [Public domain], via Wikimedia Commons
2. QUALITATIVE DESCRIPTION

Let us first describe qualitatively how the violin emits a sound:

The player excites the violin’s string through the bow-hair, the latter pushes the former toward the case, and then drives it laterally, thanks to the friction force acting in-between. As the friction exceeds the static limit, a relative sliding between bow-hair and string occurs. Depending on the system state, the friction force can change from static to dynamic, or vice versa (stick & slip). During the motion, the string dissipates mechanical energy into heat because of the viscous damping (axial, shear, bending and torsional) and the dynamic friction between bow-hair and string. The string’s vibration produces reaction forces on the bridge. These reaction forces excite the violin’s case, making it radiate the sound.

Figure 6 violin player

By Dove Bongartz (Own work) [CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0) or GFDL (http://www.gnu.org/copyleft/fdl.html)], via Wikimedia Commons
The player is inside a closed loop [2]:

We can simplify the closed loop in the following way:

1) We study the string motion as if the string were mechanically decoupled from the body (fixed string boundaries)
2) We compute the reaction forces at the bridge
3) We apply the reaction forces at the bridge to the body
4) We study the case transfer function

Since this is a really long research, my thesis activity will focus just on the first two points:

- Bowed string motion
- Reaction forces at the bridge

After the II World War, many physicists gave a qualitative physical interpretation to the most common violin string motion types. Let me remind the most important authors in the scientific literature of bowed musical instruments:
3. LITERATURE

3.1. Helmholtz [3]

- In the most common motion, the so called “Helmholtz corner”, a sharp kink that splits the string into two straight lines, travels back and forth along the string, changing orientation when it reaches the bridge or the finger.
- When the Helmholtz corner passes the bow it triggers the transition from stick to slip and vice versa.

![Helmholtz corner](image)

3.2. Raman [4]

- There can be more than one Helmholtz corner at the same time (“higher types” or “bifurcation events”).
- Velocity at bow always shifts between two fixed values.
- When the string moves periodically and at a natural frequency of the string, the friction force between bow-hair and string is almost constant.
- Raman suggested also a linear model for strings excited by an external harmonic force.
3.3. **Schelleng [5]**

- The Helmholtz corner motion depends mostly on 3 parameters: bow speed, bow force and bow distance from the bridge. The adimensional bow distance from the bridge is conventionally named “\( \beta \)”.
- Fixing bow speed and bow position, the bow force must belong to a certain interval in order to guarantee a steady motion. Schelleng diagram shows the Helmholtz region, that is bow force interval versus bow position for a given bow speed.
- Over this region we have “crunch” (sticking too long), under it instead we have “surface sound” (slipping too soon). Upper and lower region boundaries follow, respectively, a \(-1\) and \(-2\) power laws.
- The closer to the bridge is the bow, the higher is the bow force, the smaller is the bow force interval, and the higher must be the player’s ability.

![Schelleng diagram](image)

3.4. **Cremer and Lazarus [6]**

- Using an electromagnetic sensor, Cremer and Lazarus were able to measure velocity at bow profiles, for different bow forces.
- The closer to the bridge is the bow, the higher is the bow force, the sharper is velocity at bow profile and the “brighter” is the tone.
- Helmholtz corner gets sharper passing through the bow contact point.
- Helmholtz corner gets more rounded during the motion due to bending damping, that is more effective at high frequencies.
3.5. Friedlander [7]

- Friedlander construction is the intersection between Coulomb friction law and the relationship between friction force and speed at bow point in the classical string wave propagation model.
- More than one intersection can be found. In that case, we must choose the solution that generates the smoothest velocity at bow profile. We must choose the “sticking” friction coefficient if and only if the motion condition was already “sticking”, and vice versa. The middle intersection is unstable, and must never be chosen.
- These multiple solutions generate an hysteresis mechanism, that leads to an unexpected effect, the so called “flattening”: the period of the motion is systematically lengthened relative to the natural period of the free string.

![Friedlander curve](image)

*Figure 10 Friedlander curve*
3.6. Latest numerical simulations

Regarding numerical models, let me quote again Jim Woodhouse [1]:

“Exploration of extended models of this kind is in its infancy, but some preliminary results have been published. If nonlinear effects (other than those arising from the frictional interaction) need to be incorporated, that would require a more serious reformulation of the model. However, one might hope to get away with a predominantly linear theory, perhaps incorporating weak non-linear effects via a perturbation approach. This question has so far received only very limited attention in the literature, but the basic equations for coupled transverse and axial vibration of a string are well known.”

Some recent numerical models are based on the ‘electrical circuit similarity’ approach [8]. The violin is treated as an electrical circuit, whose impedances are computed from real measured quantities. These models are accurate for sure, but lack of physical meaning.

Another possible approach is to approximate the string as a discrete system, made by concentrated masses connected in series by means of springs [9].

We could even think of using a FEM code for beams.

We decided to develop a code based on the acoustics non-linear equations that we found in “Theoretical Acoustics” by Morse and Ingard [9].
4. IMPORTANT ASPECTS

The previous models are not able to give quantitative results about the string motion, but they give a physically meaningful interpretation of the string mechanical behavior, and they can be used to validate a numerical model.

A complete physical model should take in account many aspects:

- **kinematics**
  - torsional motion
- **boundaries**
  - string part from the tailpiece to the bridge
  - string and body mechanical coupling
  - player’s finger
- **internal forces**
  - shear
  - bending
  - torsion
- **load**
  - closed control loop
  - bow force applied on the string surface, leading to torsional motion
  - load distributed on a finite string surface
  - different bow movements
  - thermal effects on friction
- **damping**
  - viscous damping
  - structural damping
  - aerodynamic damping

From now on we will make these approximation:

- Only the string part from the bridge to the peg is considered
- String and body are mechanically decoupled
- Only axial internal forces
- The excitation is an imposed motion of the bow-string contact node
5. PHYSICAL MODEL

Let us consider a orthonormal “right hand” base \((\vec{i}, \vec{j}, \vec{k})\):

- \(\vec{i}\) versor going from the bridge to the pegs (longitudinal or x direction)
- \(\vec{j}\) versor parallel to the case and perpendicular to \(\vec{i}\) (transversal or y direction)
- \(\vec{k} = \vec{i} \times \vec{j}\) (vertical or z direction)

![physical model frame](image)

Let us consider a string long \(l_0\), that elongates to \(l_t\) applying a static tension:

\[
\varepsilon_t = \int_{l_0}^{l_t} \frac{dl}{l} = \ln \frac{l_t}{l_0} \quad \rightarrow \quad l_t = e^{\varepsilon_t}l_0 = kl_0 \quad k = e^{\varepsilon_t}
\]

Let us consider a generic point distant \(x_0\) from the bridge in the untensed string. Its positions in the untensed, tensed, and bowed string are, respectively:

\[
\vec{s}_{0(x_0)} = x_0 \vec{i} + 0 \vec{j} + 0 \vec{k} \quad \text{untented string}
\]
\[
\vec{s}_{t(x_0)} = x_t \vec{i} + 0 \vec{j} + 0 \vec{k} \quad x_t = kx_0 \quad \text{tensed string}
\]
\[
\vec{s}_{(x_0,t)} = (x_t + \alpha(t)) \vec{i} + \beta(t) \vec{j} + \gamma(t) \vec{k} \quad \text{bowed string}
\]

Let us consider an infinitesimal string element, long \(dx_0\) in the untensed string:
Displacement field:

\[
\vec{s}(x_0 + dx_0,t) = \vec{s}(x_0,t) + \frac{\partial \vec{s}(x_0,t)}{\partial x_0} \, dx_0 + \mathcal{O}(dx_0) \quad \text{Taylor expansion}
\]

Elongation field:

\[
l(x_0,t) = |\vec{s}(x_0 + dx_0,t) - \vec{s}(x_0,t)| = \left| \frac{\partial \vec{s}(x_0,t)}{\partial x_0} \right| \, dx_0
\]

Strain field:

\[
\varepsilon(x_0,t) = \int_{dx_0} \frac{dl(x_0,t)}{l(x_0,t) \, dx_0} = \ln \frac{l(x_0,t)}{dx_0} = \ln \left| \frac{\partial \vec{s}(x_0,t)}{\partial x_0} \right| \, dx_0 = \ln \left| \frac{\partial \vec{s}(x_0,t)}{\partial x_0} \right|
\]

Stress field:

\[
\sigma(x_0,t) = E \varepsilon(x_0,t)
\]
Tangential direction field:

\[ \mathbf{t}_{(x_0,t)} = \frac{\partial \mathbf{s}_{(x_0,t)}}{\partial x_0} \left| \frac{\partial \mathbf{s}_{(x_0,t)}}{\partial x_0} \right| \]

Internal axial force field:

\[ \vec{N}_{(x_0,t)} = N_{(x_0,t)} \mathbf{t}_{(x_0,t)} = A \sigma_{(x_0,t)} \mathbf{t}_{(x_0,t)} \]

\[ \vec{N}_{(x_0+dx_0,t)} = \vec{N}_{(x_0,t)} + \frac{\partial \vec{N}_{(x_0,t)}}{\partial x_0} dx_0 + \sigma_{(dx_0)} \text{ Taylor expansion} \]

Computing mechanical equilibrium on the infinitesimal element:

\[ dM \left( \frac{\partial}{\partial t} \right)^2 \mathbf{s}_{(x_0,t)} = \vec{N}_{(x_0+dx_0,t)} - \vec{N}_{(x_0,t)} = \frac{\partial \vec{N}_{(x_0,t)}}{\partial x_0} dx_0 \]

\[ \left( \frac{\partial}{\partial t} \right)^2 \mathbf{s}_{(x_0,t)} = \frac{dx_0}{dM} \frac{\partial \vec{N}_{(x_0,t)}}{\partial x_0} = \frac{EA dx_0}{dM} \frac{\partial}{\partial x_0} (\varepsilon_{(x_0,t)} \mathbf{t}_{(x_0,t)}) \]

\[ \left( \frac{\partial}{\partial t} \right)^2 \mathbf{s}_{(x_0,t)} = \frac{E}{\rho} \frac{\partial}{\partial x_0} (\varepsilon_{(x_0,t)} \mathbf{t}_{(x_0,t)}) \]

We can approximate the acceleration field with finite differences over elements long \( h \):

\[ \left( \frac{\partial}{\partial t} \right)^2 \mathbf{s}_{(x_0,t)} = \frac{E}{\rho \ h} \left[ \ln \left| \frac{1}{h} \left( \mathbf{s}_{(x_0+h,t)} - \mathbf{s}_{(x_0,t)} \right) \right| \right. \]

\[ - \ln \left| \frac{1}{h} \left( \mathbf{s}_{(x_0,t)} - \mathbf{s}_{(x_0-h,t)} \right) \right| \]

\[ \frac{1}{h} \left( \mathbf{s}_{(x_0+h,t)} - \mathbf{s}_{(x_0,t)} \right) \]

\[ \left. \frac{1}{h} \left( \mathbf{s}_{(x_0,t)} - \mathbf{s}_{(x_0-h,t)} \right) \right] \]
\[
\left( \frac{\partial}{\partial t} \right)^2 \tilde{s}(x_0,t) = \frac{E}{\rho h} \ln \left[ \frac{1}{h} \left( \tilde{s}(x_0+h,t) - \tilde{s}(x_0,t) \right) \right] \left( \frac{\tilde{s}(x_0+h,t) - \tilde{s}(x_0,t)}{\tilde{s}(x_0+h,t) - \tilde{s}(x_0,t)} \right) \\
- \ln \left[ \frac{1}{h} \left( \tilde{s}(x_0,t) - \tilde{s}(x_0-h,t) \right) \right] \left( \frac{\tilde{s}(x_0+h,t) - \tilde{s}(x_0-h,t)}{\tilde{s}(x_0,t) - \tilde{s}(x_0-h,t)} \right) 
\]
5.1. Viscous damping

Let us refine our physical model considering a linear viscous damping:

\[
\sigma(x_0, t) = E(\varepsilon(x_0, t) + c \dot{\varepsilon}(x_0, t))
\]

\[
\dot{\varepsilon}(x_0, t) = \frac{1}{l(x_0, t)} \frac{1}{d x_0} \dot{l}(x_0, t) = \frac{1}{l(x_0, t)} \dot{l}(x_0, t)
\]

Only the tangential velocity component gives a contribution to elongation, while a constant velocity field gives no contribution at all:

\[
\dot{i}(x_0, t) = \left( \dot{s}(x_0 + dx_0) - \dot{s}(x_0) \right) * \dot{t}(x_0, t)
\]

\[
\dot{s}(x_0 + dx_0) = \dot{s}(x_0) + \frac{\partial \dot{s}(x_0)}{\partial x_0} dx_0 + \sigma(dx_0)
\]

\[
\dot{i}(x_0, t) = \left( \frac{\partial \dot{s}(x_0)}{\partial x_0} * \dot{t}(x_0, t) \right) dx_0
\]

We will make the following approximation:

\[
\dot{\varepsilon}(x_0, t) = \frac{l(x_0, t)}{l(x_0, t)} \dot{i}(x_0, t) \approx \frac{\dot{l}(x_0, t)}{d x_0} \dot{l}(x_0, t) \approx \frac{\partial \dot{s}(x_0, t)}{\partial x_0} * \dot{t}(x_0, t)
\]

Adding the viscous contribution to the physical model:

\[
\left( \frac{\partial}{\partial t} \right)^2 \ddot{s}(x_0, t) = \frac{E}{\rho} \frac{\partial}{\partial x_0} \left[ (\varepsilon(x_0, t) + c \dot{\varepsilon}(x_0, t)) \ddot{t}(x_0, t) \right]
\]
Applying finite differences on elements long $h$:

\[
\left( \frac{\partial}{\partial t} \right)^2 \tilde{s}(x_0, t) = \frac{E}{\rho} \frac{1}{h} \left\{ \ln \left| \frac{1}{h} \left( \tilde{s}(x_0 + h, t) - \tilde{s}(x_0, t) \right) \right| + \frac{c}{h} \left( \dot{\tilde{s}}(x_0 + h, t) - \dot{\tilde{s}}(x_0, t) \right) \right. \\
\left. \times \frac{\tilde{s}(x_0 + h, t) - \tilde{s}(x_0, t)}{\tilde{s}(x_0 + h, t) - \tilde{s}(x_0, t)} \right| \frac{\tilde{s}(x_0 + h, t) - \tilde{s}(x_0, t)}{\tilde{s}(x_0 + h, t) - \tilde{s}(x_0, t)} \\
- \left[ \ln \left| \frac{1}{h} \left( \tilde{s}(x_0, t) - \tilde{s}(x_0 - h, t) \right) \right| + \frac{c}{h} \left( \dot{\tilde{s}}(x_0, t) - \dot{\tilde{s}}(x_0 - h, t) \right) \right. \\
\left. \times \frac{\tilde{s}(x_0, t) - \tilde{s}(x_0 - h, t)}{\tilde{s}(x_0, t) - \tilde{s}(x_0 - h, t)} \right| \frac{\tilde{s}(x_0, t) - \tilde{s}(x_0 - h, t)}{\tilde{s}(x_0, t) - \tilde{s}(x_0 - h, t)} \right\} 
\]
5.2. Excitation

The bow excitation will be studied as an imposed motion of the contact node. We can suppose that the friction force acts only along the transversal direction, so we must define an imposed motion that can be guaranteed just by a transversal force.

The following picture represents a generic imposed motion of the contact node $\vec{B}$ going to position $\vec{B}'$, thanks to a generic force $\vec{F}$.

The contact node divides the string AC into 2 segments, a shorter one (segment BC, long $s_s$) and a longer one (segment BA, long $l_s$), that elongate after the imposed motion (respectively: segment B'C, long $s_d$; segment B'A, long $l_d$).

Contact point displacement can be decomposed into a longitudinal and a transversal direction. Let us use the frame shown in the previous picture:

$$\vec{B}' - \vec{B} = x\hat{i} + y\hat{j}$$

Let us neglect inertia forces for the moment. The elastic forces are:

$$\overrightarrow{T_l} = AE\varepsilon_l \frac{\hat{A} - \hat{B}'}{|\hat{A} - \hat{B}'|} = AE\varepsilon_l \left[ -\frac{l_s + x}{l_d} \hat{i} - \frac{y}{l_d} \hat{j} \right]$$

$$\overrightarrow{T_s} = AE\varepsilon_s \frac{\hat{C} - \hat{B}'}{|\hat{C} - \hat{B}'|} = AE\varepsilon_s \left[ -\frac{s_s - x}{s_d} \hat{i} - \frac{y}{s_d} \hat{j} \right]$$
We compute force “F” applying mechanical equilibrium at contact point:

\[ \vec{F} = - (\vec{T}_l + \vec{T}_s) = AE \left[ \left( \varepsilon_l \frac{l_s + x}{l_d} - \varepsilon_s \frac{s_s - x}{s_d} \right) \hat{i} + y \left( \frac{\varepsilon_l}{l_d} + \frac{\varepsilon_s}{s_d} \right) \hat{j} \right] = F_x \hat{i} + F_y \hat{j} \]

Considering static tension, the two string segments untensed lengths are:

\[ l = \frac{l_s}{k} \quad s = \frac{s_s}{k} \]

We can compute the strain during the imposed motion:

\[ \varepsilon_l = \ln \frac{l_d}{l} = \ln \left( \frac{\sqrt{(l_s + x)^2 + y^2}}{l_s} k \right) \]

\[ \varepsilon_s = \ln \frac{s_d}{s} = \ln \left( \frac{\sqrt{(s_s - x)^2 + y^2}}{s_s} k \right) \]

Substituting in the force equation, we obtain the force longitudinal component:

\[ F_{x(x,y,l_s,s_s,k)} = AE \left( \frac{l_s + x}{\sqrt{(l_s + x)^2 + y^2}} \ln \left( \frac{\sqrt{(l_s + x)^2 + y^2}}{l_s} \right) \right) \]

\[ - \frac{s_s - x}{\sqrt{(s_s - x)^2 + y^2}} \ln \left( \frac{\sqrt{(s_s - x)^2 + y^2}}{s_s} \right) \]

Equating to zero the previous equation, we impose a bow force that keeps the transversal direction. Let us baptize this equation iso-clinamic (from the ancient Greek “iso”=equal and the Latin “clinamen”=slope):

\[ F_{x(x,y,l_s,s_s,k)} = 0 \]
Let us rewrite the previous equation in a more convenient form:

\[
\frac{l_s + x}{\sqrt{(l_s + x)^2 + y^2}} \ln \left( \frac{\sqrt{(l_s + x)^2 + y^2}}{l_s} \right) - \frac{s_s - x}{\sqrt{(s_s - x)^2 + y^2}} \ln \left( \frac{\sqrt{(s_s - x)^2 + y^2}}{s_s} \right) = 0
\]

\[
\ln \left[ \left( \frac{\sqrt{(l_s + x)^2 + y^2}}{l_s} \right)^{\frac{l_s + x}{\sqrt{(l_s + x)^2 + y^2}}} \right] - \ln \left[ \left( \frac{\sqrt{(s_s - x)^2 + y^2}}{s_s} \right)^{\frac{s_s - x}{\sqrt{(s_s - x)^2 + y^2}}} \right] = 0
\]

\[
f_{(x,y,l_s,s_s,k)} = \left( \frac{\sqrt{(l_s + x)^2 + y^2}}{l_s} \right)^{\frac{l_s + x}{\sqrt{(l_s + x)^2 + y^2}}} - \left( \frac{\sqrt{(s_s - x)^2 + y^2}}{s_s} \right)^{\frac{s_s - x}{\sqrt{(s_s - x)^2 + y^2}}} = 0
\]

\[
f_{(x,y,l_s,s_s,k)}\text{ seems to depend on 5 parameters: the first four are dimensionally lengths, the last one is adimensional. Let us apply Buckingham π theorem to generate a relationship among 4 adimensional parameters:}
\]

\[
\beta \triangleq \frac{s_s}{t_s} \quad s_s = \beta t_s \quad l_s = t_s - s_s = (1 - \beta) t_s \quad 0 \leq \beta < 1
\]

\[
\bar{x} = \frac{x}{t_s} \quad \bar{y} = \frac{y}{t_s}
\]

\[
f_{(\bar{x},\bar{y},\beta,k)} = \left( \frac{k}{1 - \beta} \sqrt{(1 + \bar{x} - \beta)^2 + \bar{y}^2} \right)^{\frac{1+\bar{x}-\beta}{\sqrt{(1+\bar{x}-\beta)^2+\bar{y}^2}}} - \left( \frac{k}{\beta} \sqrt{(\beta - \bar{x})^2 + \bar{y}^2} \right)^{\frac{\beta - \bar{x}}{\sqrt{(\beta - \bar{x})^2 + \bar{y}^2}}}
\]

Once \(\bar{y}\) is numerically computed for each \(\bar{x}\), the isoclinamic trajectory depends just on \(\beta\) and \(k\).
The following is an isoclinamic trajectory computed for $\beta = 0.25$, $k=1.01$. It starts in the origin and is symmetric respect to the x axis. From now on we will consider just the positive quadrant.

Let us study the isoclinamic trajectory varying $\beta$ and $k$:

![Image 14](isoclinamic trajectory)

*Figure 14  isoclinamic trajectory*

![Image 15](isoclinamic trajectory, k dependancy)

*Figure 15  isoclinamic trajectory, k dependancy*

![Image 16](isoclinamic trajectory, beta dependancy)

*Figure 16  isoclinamic trajectory, beta dependancy*
Considering $\beta$ and $k$ of our case, the dimensional isoclinamic trajectory is the following:

As you can see, the isoclinamic trajectory is almost a straight transversal line, even for huge transversal displacement (1 cm). We can therefore approximate the isoclinamic trajectory with a completely transversal one.
5.3. Forces evaluation

We had to evaluate bow and bridge reaction forces. Introducing a concentrated load directly in the infinitesimal string element was difficult, so we found a trick to get over this mathematical obstacle.

We can consider string elements as springs with a certain mass amount, on which inertia and elastic forces act. Decreasing the elements’ length, the individual spring mass amount and inertia forces tend to vanish, while the elastic forces remain:

\[
\lim_{\text{elements} \to +\infty} (F_{el} + F_{in}) = \lim_{\text{elements} \to +\infty} [AE\varepsilon\vec{n} + m\vec{a}_{in}] = \lim_{h \to 0^+} A[E\varepsilon\vec{n} + \rho h\vec{a}_{in}] = AE\varepsilon\vec{n}
\]

This means that, using a proper number of nodes, forces can be evaluated by a nodal mechanical equilibrium of elastic forces. A generic elastic force has the form:

\[
\vec{T} = AE\varepsilon\vec{n}
\]

Where \(\vec{n}\) is the versor going out of the element.

Considering the pictures in the next page:

\[
\vec{F} = -(\vec{T}_1 + \vec{T}_2)
\]

\[
\vec{R} = -\vec{T}
\]
Figure 18  bow force

Figure 19  bridge force
6. CONSIDERATIONS

6.1. Imposed motion trajectory

One could say, since we approximated the isoclinamic trajectory with a completely transversal one, that in our model the friction force has a longitudinal component.

Actually, after developing the isoclinamic theory, we found experimental evidence that the friction force acts also along the longitudinal direction. Considering this experimental evidence, a non isoclinamic trajectory is physically possible.

In our analysis, we will make the following approximations:

- The contact node remains the same during the motion
- The contact node moves only transversally, even during slip motion
- Longitudinal friction is neglected when computing stick-slip transitions

These approximations lead to transversal displacement values that are realistic, but can differ from the real ones. In fact, in real cases:

- The bow follows a generic trajectory
- The bow has a tilt angle
- The contact point changes during slip

6.2. Damping

We can think of three forms of damping:

- Viscous
- Structural
- Aerodynamic

In literature we can find estimates of viscous damping and, sometimes, structural damping, while aerodynamic damping is completely negligible.

In our model we took into account axial viscous damping through a linear relationship:

\[ N = A\sigma \]
\[ \sigma = E(\varepsilon + c\dot{\varepsilon}) \]

Considering the limited data available and the low influence of damping, we decided to neglect it completely.
6.3. Coding

In this section we will speak about the most relevant aspects specific of programming. We used Matlab version 2015b and took advantage of its library functions. We used function “ode45” to integrate the equation of motion and record velocities and positions at each integration instant. Ode45 computes on its own the best time discretization step in order to achieve a stable and accurate solution, this means that, for long integration times, by default, it records a huge amount of data. For example, integrating for 0.1 s with a mesh of 1024 nodes, ode45 would probably exceeds the RAM of a common PC (4GB), stopping the running job. To deal with this problem, we made a decimation of the solution, that is we recorded the solution given by ode45 just at some integration instants. We set the “decimation frequency” to 50 kHz to stay on the safe side, knowing by experience that common mechanical problems do not exceed this frequency. Decimation does not imply ode45 made bigger integration steps (achieving a wrong solution), it means we recorded just the amount of data we needed for our analyses.

ode45 does not record the evaluations of the differential equation to be integrated, in our case, it did not record accelerations. We made an attempt to record accelerations: we forced the differential equation function to record its evaluations at each call of ode45. Unluckily, not all its calls are effective in generating the solution. In fact, ode45 evaluates the differential equations, but proceeds with the integration if and only if the convergence criterion is satisfied. Recording all the differential equation evaluations led us to record accelerations and times that were not even used to generate the solution. Ultimately, it was not possible to store accelerations using ode45. We could have coded a function on our own instead of using a library function, but we considered also that a Matlab library function is always faster and more reliable than a function you write on your own, so we decided to give up recording accelerations (friction values et cetera) and be satisfied with decimated positions and velocities.
7. CODE VALIDATION

After developing the model and writing the code, we had to validate the code. We did it by running analyses of simple cases, for which scientific literature provides an exact analytical solution, and comparing analytical and numerical results:

1) Fundamental harmonics
2) Wave propagation
3) Helmholtz corner

For the following analyses we will consider the same violin $E_5$ string:

- $E_5 = \omega_0 \frac{rad}{s} = 4142.3 \ \frac{rad}{s} = 659.26 \ Hz$
- $E = 220000 \ MPa$
- $\rho = 7700 \ \frac{kg}{m^3}$
- $d = 0.25 \ mm$
- $l_T = 325 \ mm \ (from \ the \ bridge \ to \ the \ pegs)$

<table>
<thead>
<tr>
<th>fundamental harmonic #</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
<th>V</th>
<th>VI</th>
<th>VII</th>
<th>VIII</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega \ [\frac{rad}{s}]$</td>
<td>4142.3</td>
<td>8284.5</td>
<td>12427</td>
<td>16569</td>
<td>20711</td>
<td>24854</td>
<td>28996</td>
<td>33138</td>
</tr>
<tr>
<td>$f \ [Hz]$</td>
<td>659.26</td>
<td>1318.52</td>
<td>1977.78</td>
<td>2637.04</td>
<td>3296.30</td>
<td>3955.56</td>
<td>4614.82</td>
<td>5274.08</td>
</tr>
<tr>
<td>$T \ [s]$</td>
<td>0.0015169</td>
<td>0.00075843</td>
<td>0.00050562</td>
<td>0.00037921</td>
<td>0.00030337</td>
<td>0.00025281</td>
<td>0.00021669</td>
<td>0.00018961</td>
</tr>
</tbody>
</table>
Before starting, we must “tune” this string, that is we must compute the static preload $N$ in order to guarantee its pitch. Considering a linear model of a hinged-hinged string [3]:

$$v_{(x,t) \text{ stationary}} = A \sin(y_k l_T) \cos(\omega_{0k} t) \quad [m] \quad A \to 0^+ \quad [m]$$

$$y_k = k \frac{\pi}{l_T} \quad [m^{-1}] \quad k \in \mathbb{N} / k > 0 \quad [-]$$

$$c \triangleq \sqrt{\frac{N}{\rho A}} \quad \left[ \frac{m}{s} \right] \quad \rho \triangleq \frac{M}{V} \quad \left[ \frac{kg}{m^3} \right]$$

$$\omega_{0k} = c y_k \quad \left[ \frac{rad}{s} \right]$$

$$T_k = \frac{2\pi}{\omega_{0k}} \quad [s]$$

Considering the first fundamental harmonic:

$$y_1 = \frac{\pi}{l_T} = 9.6664 \quad m^{-1}$$

$$c = \frac{\omega_{01}}{y_1} = 428.52 \quad \frac{m}{s}$$

$$N = \rho A c^2 = 69.407 \quad N$$

$$\sigma = \frac{N}{A} = \rho c^2 = 1413.9 \quad MPa$$

$$\epsilon_t = \frac{\sigma}{E} = 0.0064268 \quad [-]$$

$$k = e^{\epsilon_t} = 1.0064 \quad [-]$$

$$l_0 = \frac{l_T}{k} = 322.92 \quad mm$$
7.1. **Fundamental harmonics**

We will now study the first 8 fundamental harmonics, considered the most important ones in terms of sound generation by music tradition, and also the less affected by damping.

Initial conditions were imposed in such a way that, at the first instant, the string has a sine shape in the x-y plane, and null speed.

A very low sine amplitude was chosen ($\frac{l^2}{10^6}$) not to excite non-linearities.

Initial conditions were put shifting the nodes transversally respect to their equilibrium position. This approach is not completely correct, because in the real world string points moves also along the longitudinal direction during a harmonic motion. Anyway, for small displacements, the longitudinal displacement is negligible compared to the transversal one, and this is our case.

Spectral analysis of the motion hourly law will tell us if the computed motion matches with the fundamental harmonic. The sampling frequency was 50 kHz, much higher than $2 \times \omega_{0viu}$, satisfying Shannon theorem and letting us detect also compressive vibration modes.

Errors in the results are due to the following approximations:

- Finite amplitude
- Harmonics longitudinal displacement neglected
- Discrete model

In the next pages, we will call “bow node” the first space peak of the harmonic along x direction, the one marked with a red asterisk.
7.1.1. 1\textsuperscript{st} fundamental harmonic

Figure 20  first harmonic spatial initial condition

Figure 21  first harmonic bridge force
Figure 22  first harmonic bow node position

Figure 23  first harmonic bow node velocity
Figure 24  first harmonic bow node position x PSD

Figure 25  first harmonic bow node position y PSD
Figure 26  first harmonic bow node position y spectrum
Figure 27  first harmonic bow node velocity x PSD

Figure 28  first harmonic bow node velocity y PSD
Figure 29  first harmonic bow node velocity spectrum
Figure 30  first harmonic bridge force x PSD

Figure 31  first harmonic bridge force y PSD
Figure 32  first harmonic bridge force y spectrum
7.1.1.1. comments

We can make the following considerations:

- The spectrums of bow node position, bow node velocity and bridge reaction force $y$ component show a peak at the same frequency, that is 2 Hz bigger than the first natural frequency. This is actually quite a small error, considering that human ear’s frequency resolution is less than 4 Hz.
- There is no motion along $z$ direction, as we would expect.
- There is a motion and a bridge reaction force variation along $x$ direction, though negligible respect to the $y$ direction. This was expected, and is related to the approximated spacial initial conditions of the first fundamental harmonic, for which the initial displacement in $x$ direction was neglected.
- Spectrums of bow node position, bow node velocity and bridge reaction force along $x$ direction show a peak at about 10 kHz. This may be due to a compression vibration mode, that is activated by the approximated initial conditions.

In the next pages, we will show the most meaningful results for each harmonic studied.
7.1.2. 2nd fundamental harmonic

Figure 33  second harmonic spacial initial conditions
Figure 34  second harmonic bow node position

Figure 35  second harmonic bow node position x PSD

Figure 36  second harmonic bow node position y PSD
Figure 37  second harmonic bridge force

Figure 38  second harmonic bridge force x PSD

Figure 39  second harmonic bridge force y PSD
7.1.3. 3rd fundamental harmonic

Figure 40  third harmonic spatial initial conditions
Figure 41  third harmonic bow node position

Figure 42  third harmonic bow node position x PSD

Figure 43  third harmonic bow node position y PSD
Figure 44  third harmonic bridge force

Figure 45  third harmonic bridge force x PSD

Figure 46  third harmonic bridge force y PSD
7.1.4. 4\textsuperscript{th} fundamental harmonic
Figure 48  fourth harmonic bow node position

Figure 49  fourth harmonic bow node position x PSD

Figure 50  fourth harmonic bow node position y PSD
Figure 51  fourth harmonic bridge force

Figure 52  fourth harmonic bridge force x PSD

Figure 53  fourth harmonic bridge force y PSD
7.1.5. 5th fundamental harmonic

Figure 54  fifth harmonic spatial initial conditions
Figure 55  fifth harmonic bow node position

Figure 56  fifth harmonic bow node position x PSD

Figure 57  fifth harmonic bow node position y PSD
Figure 58  fifth harmonic bridge force

Figure 59  fifth harmonic bridge force x PSD

Figure 60  fifth harmonic bridge force y PSD
7.1.6. 6th fundamental harmonic

Figure 61 sixth harmonic spacial initial conditions
Figure 62  sixth harmonic bow node position

Figure 63  sixth harmonic bow node position x PSD

Figure 64  sixth harmonic bow node position y PSD
Figure 65  sixth harmonic bridge force

Figure 66  sixth harmonic bridge force x PSD

Figure 67  sixth harmonic bridge force y PSD
7.1.7. 7th fundamental harmonic

Figure 68  seventh harmonic spatial initial conditions
Figure 69  seventh harmonic bow node position

Figure 70  seventh harmonic bow node position x PSD

Figure 71  seventh harmonic bow node y position PSD
Figure 72  seventh harmonic bridge force

Figure 73  seventh harmonic bridge force x PSD

Figure 74  seventh harmonic bridge force y PSD
7.1.8.  \textbf{8}^{\text{th}} \text{ fundamental harmonic}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure75}
\caption{eighth harmonic spatial initial conditions}
\end{figure}
Figure 76  eighth harmonic bow node position

Figure 77  eighth harmonic bow node position x PSD

Figure 78  eighth harmonic bow node position y PSD
Figure 79  
**eighth harmonic bridge force**

Figure 80  
**eighth harmonic bridge force x PSD**

Figure 81  
**eighth harmonic bridge force y PSD**
7.1.9. comments

The peaks of power spectral density diagrams related to the bridge force y component match with the ones related to the bow node position y component. The error related the first eight natural frequencies is completely negligible, as you can see from the following table.

<table>
<thead>
<tr>
<th>fundamental harmonic #</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
<th>V</th>
<th>VI</th>
<th>VII</th>
<th>VIII</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{\text{analytical}} ) [Hz]</td>
<td>659.26</td>
<td>1318.52</td>
<td>1977.78</td>
<td>2637.04</td>
<td>3296.30</td>
<td>3955.56</td>
<td>4614.82</td>
<td>5274.08</td>
</tr>
<tr>
<td>( f_{\text{computed}} ) [Hz]</td>
<td>661</td>
<td>1323</td>
<td>1984</td>
<td>2646</td>
<td>3307</td>
<td>3968</td>
<td>4630</td>
<td>5291</td>
</tr>
<tr>
<td>error [Hz]</td>
<td>1.74</td>
<td>4.48</td>
<td>6.22</td>
<td>8.96</td>
<td>10.7</td>
<td>12.44</td>
<td>15.18</td>
<td>16.92</td>
</tr>
<tr>
<td>% error [-]</td>
<td>0.26 %</td>
<td>0.34 %</td>
<td>0.31 %</td>
<td>0.34 %</td>
<td>0.32 %</td>
<td>0.31 %</td>
<td>0.33 %</td>
<td>0.32 %</td>
</tr>
</tbody>
</table>

\[
\text{error} = f_{\text{computed}} - f_{\text{analytical}} \text{ [Hz]}
\]

\[
\% \text{error} = \frac{\text{error}}{\text{analytical value}} \times 100\%
\]

From the second fundamental harmonic, the power spectral density diagrams of bow node position x component show a peak at about 25 kHz. This peak may be related to a compression vibration mode.
7.2. Wave propagation

A node at $\beta = 0.2$ was given an initial transversal speed. A very low speed was chosen ($\frac{c_{\text{string}}}{10^6}$) not to excite non-linearities.

The following pictures represents the propagating wave in its early life, the node with initial speed is represented in red, and will be called “bow node” for this analysis.

![Wave propagation spatial initial conditions](image)

*Figure 82* wave propagation spatial initial conditions
Figure 83    wave propagation bow node position

Figure 84    wave propagation bow node velocity
Figure 85   wave propagation bridge force
Figure 86 wave propagation bow node position x PSD

Figure 87 wave propagation bow node position y PSD

Figure 88 wave propagation bow node position y spectrum
Figure 89  wave propagation bow node velocity x PSD

Figure 90  wave propagation bow node velocity y PSD

Figure 91  wave propagation bow node velocity y spectrum
Figure 92  wave propagation bridge force x PSD

Figure 93  wave propagation bridge force y PSD

Figure 94  wave propagation bridge force y spectrum
7.2.1. Comments

The power spectral density diagram of bow node position x component shows a peak at about 25 kHz, similarly to the fundamental harmonics. This peak may be related to a compression vibration mode.

The power spectral density diagram of bow node position y component shows peaks corresponding to the fundamental harmonics. Since the wave propagation is a free motion of a system that can be considered linear for such small displacements and speeds, it can be described as a superposition of the system’s vibration modes, so as a combination of its fundamental harmonics.
7.3. **Helmholtz corner**

We are now considering the simplest excitation: bowing with constant Schelleng parameters.

### 7.3.1. Convergence

We ran many analyses keeping the same input parameters and integration time, but increasing the number of nodes, in order to assess the results convergence. The following variables were taken into account:

- Bow node displacement along y direction (absolute value)
- Bow force (modulus)
- Bridge force oscillation around its static value (modulus)

For each analysis, we recorded the maximum value assumed by each of these variables in time domain. The values related to a certain variable were then normalized respect to the maximum value recorded of that variable. The following is a normalized convergence diagram, that shows that 1024 nodes give an acceptable error.

![Convergence Diagram](image)
7.3.2. Computational time

Computational times were very long for this kind of analysis, we had to estimate them before running the code.

\[ t_{\text{computational}} = f(\text{number of nodes},\text{integration time},\text{CPU frequency}) \]

Fixing integration time and CPU frequency, we recorded computational times varying the number of nodes. Finally, we used Matlab fitting tool to estimate the relationship between number of nodes and computational time.

\[ t_{\text{computational}} = p_1 \text{nodes}^2 + p_2 \text{nodes} + p_3 \ [s] \]

Supposing a linear relationship respect to integration time and an inversely proportional relationship respect to CPU frequency:

\[ t_{\text{computational}} = (p_1 \text{nodes}^2 + p_2 \text{nodes} + p_3) \left( \frac{T_{\text{integration}}}{T_{\text{ref}}} \right) \left( \frac{f_{\text{ref}}}{f_{\text{CPU}}} \right) \ [s] \]

reference integration time = 0.01 s  reference CPU frequency = 2.40 GHz
7.3.3. **Helmholtz corner sequence**

During the motion many Helmholtz corners occurred, we are showing some frames of the first one:
Figure 98  Helmholtz corner sequence
7.3.4. Time profiles and spectral analysis

We are now showing the first instants of the motion, when you can see the first stick-slip transitions.

Figure 99 Helmholtz corner bow node position

Figure 100 Helmholtz corner bow node velocity
Figure 101  Helmholtz corner bridge force

Figure 102  Helmholtz corner bow force
Figure 103  Helmholtz corner bow node position $y$

Figure 104  Helmholtz corner bow node velocity $y$
Figure 105 Helmholtz corner bow node position y PSD

Figure 106 Helmholtz corner bow node velocity y PSD
Figure 107  Helmholtz corner bridge force x

Figure 108  Helmholtz corner bridge force y

Figure 109  Helmholtz corner bridge force z
Figure 110  Helmholtz corner bridge force x PSD

Figure 111  Helmholtz corner bridge force y PSD

Figure 112  Helmholtz corner bridge force z
Figure 113  Helmholtz corner bow force x

Figure 114  Helmholtz corner bow force y

Figure 115  Helmholtz corner bow force z
7.3.5. Comments

We ran many analyses of this kind, choosing Schelleng parameters in order to get a steady Helmholtz corner. The most meaningful analysis had the following parameters:

- **Bow speed** = $0.1 \text{ m/s}$
- $\beta = 0.2$
- **pressing depth** = $2 \times d_{\text{string}}$

The bow speed is relatively low, in this way there were bigger time integration steps and a lower computational time.

$\beta$ has a high value: in this way, there is a wider bow force interval that generates a steady Helmholtz corner. In this way, the chosen bow force was more likely to fall into this interval. This shrewdness saved a lot of time, since we did not have much quantitative information about Schelleng diagram.

![Schelleng diagram](image)

Since our physical model is based on an imposed motion, and not on an external force, we had to express the maximum bow force as a distance. So, we expressed the maximum bow force as the maximum depth at which the bow presses the violin string along $-z$ direction. In this case, the pressing depth was equal to 2 times the string diameter.

Even in the first instants, it is clear the stick and slip mechanism. In the first four pictures of 8.2, for example, you can see many transitions from stick to slip, and a corresponding drop in the bow force, bridge force, and bow node position along $y$ direction. In the picture showing the bow node velocity along $y$ direction instead, it is clear that the bow node keeps the bow speed during stick, and after a transition from stick to slip there is an oscillation around this value. The bow node velocity along $y$ direction may have a flat profile around another value in case the friction force acting between bow and string balances the string elastic force.
The change from stick to slip condition triggers the Helmholtz corner, so, as the bowing goes on, more and more Helmholtz corner appear, and propagate along the string part from the bow contact point to the peg. Watching the video generated by the code, the first Helmholtz corner seems to have a period nearly equal to 0.0012 s. This period is lower than the period of the first fundamental harmonic (0.0015 s), and the reason is that the first fundamental harmonic propagates along the entire string length $l_t$, while Helmholtz corner propagates along $(1 - \beta)l_t$:

$$\frac{T_{Helmholtz\ corner}}{T_{\text{first fundamental harmonic}}} = \frac{(1 - \beta)l_t}{l_t} = (1 - \beta) = 0.8 = \frac{0.0012\ s}{0.0015\ s}$$

Power spectral density diagrams of bridge force and bow force seem to share the same harmonically spaced peaks.
8. THE CODE

The numerical model was implemented in the high level programming language Matlab 2015b, and it is made up by the following files:

![Figure 120 - code files](image)

The analysis starts after running “job” files, the ones containing the analysis information. In the next pages, we are showing the code.
8.1. job_harmonic

```matlab
% excitation parameters
excitation='harmonic';
har=8; % fundamental harmonic 
beta=1/(2*har);

% integration parameters
nodes=1024;
T=1;

pre_processing
processing
save harmonic
post_processing

quit
% system('shutdown -s')
```

8.2. job_wave_propagation

```matlab
% excitation parameters
excitation='wave propagation';
beta=0.2;

% integration parameters
nodes=1024;
T=1;

pre_processing
processing
save wave_propagation
post_processing

quit
% system('shutdown -s')
```

8.3. job_imposed_dof

```matlab
% Schelleng parameters
excitation='imposed dof';
beta=0.2;
pressing_depth=1*(0.25/1000);
bow_speed=0.1;

% integration parameters
nodes=1024;
T=3;

pre_processing
processing
save imposed_dof
post_processing

quit
% system('shutdown -s')
```
8.4. pre_processing

%% string data
string_data

%% space discretization
h=10/(nodes-1); % untensed element length [m]
m=rho*A*h; % mass [kg]
Erho=E/rho;
Erhoh=Erho/h;
dof=3*nodes; % each node has 3 dof (position caertesian components)
doft=2*dof; % 6 independent variables considering velocity
first=1:1:dof; % velocities indexes within y, acceleration indexes within
y_dot=f(t,y)
second=first+dof; % positions indexes within y, velocities indexes within
y_dot=f(t,y)

%% mesh + I.C. @ stable static equilibrium position
y0=zeros(1,doft); % y0=[v0,x0]
y0((dof+1):3:(doft-2))=linspace(0,lt,nodes); % nodes evenly distributed along x direction

%% load
bow_node_x=beta*lt; % bow node equilibrium position in x direction
bow_node=round(bow_node_x/h)+1; % bow node
switch bow_node % bow node can be neither the first nor the last node because of BC
    case 1
        bow_node=2;
    case nodes
        bow_node=nodes-1;
end

bow_node_first_index=(1:1:3)+3*(bow_node-1); % bow node velocity index within y, acceleration index within y_dot=f(t,y)
bow_node_second_index=bow_node_first_index+dof; % position velocity
bow_node_x=y0(bow_node_second_index(1)); % bow node equilibrium position in x direction considering space discretization

switch excitation
    case 'harmonic'
        harmonic
    case 'wave propagation'
        wave_propagation
    case 'imposed dof'
        imposed_dof
    otherwise
        disp('invalid excitation')
end
8.5. string_data

% violin E5 string
unit

% acoustic properties
f_E5=659.26; % first fundamental frequency [Hz]
w_E5=2*pi*f_E5; % first fundamental frequency [rad/s]
period1=2*pi/w_E5; % first fundamental harmonic period [s]

% material
E=0.220*10^(12); % Young modulus [Pa]
rho=7700; % density [kg/m^3]

% geometry
lt=0.325; % unloaded length [m]
d=0.25/1000; % diameter [m]
r=d/2; % radius [m]
A=pi*r^2; % area [m^2]

% tuning
N=rho*A*(w_E5/(pi/lt))^2; % static tension [N]
AE=A*E;
eps0=N/(AE); % static strain [-]
k=exp(eps0); % static elongation factor [-]
l0=lt/k; % static length [m]

c_string=sqrt(N/(rho*A));

% friction
mu_s=1.2; % adimensional static friction coefficient [-]
u_d=0.375; % dynamic [-]
v=1000 -0.30 -0.18 -0.13 -0.08 -0.04 -0.0250 -0.02 -0.0150 -0.01 -0.0150 -0.0250 -0.0250 -0.02 -0.13 0.18 0.30 1000;
mu=[0.375 0.375 0.42 0.47 0.55 0.66 0.73 0.77 0.825 0.9 1.2 -1.2 -0.9 -0.825 -0.77 -0.73 -0.66 -0.55 -0.47 -0.42 -0.375 -0.375];
8.6. harmonic

imposed=0;
amplitude=lt*1e-6;
shape=0:1:(nodes-1);
shape=shape*har*pi/(nodes-1);
shape=sin(shape)*amplitude;

interval=(dof+2):3:(doft-1);  % nodes position y component indexes within y0
y0(interval)=shape;           % assigning space profile in y0

clear amplitude shape interval

8.7. wave_propagation

imposed=0;

y0(bow_node_first_index(2))=c_string*1e-6;  % bow node initial velocity y component

8.8. imposed_dof

imposed=1;

bow_velocity_z=-bow_speed;
bow_velocity_y=bow_speed;

pressing_time=pressing_depth/bow_speed;
if (T<pressing_time)||(T==pressing_time)
    T=pressing_time+2e-5;
end

bow_length=0.5;
% driving_time=bow_length/bow_velocity_y+pressing_time;
driving_time=T;
8.9. processing

%% numerical integration

sampling_frequency=25000; % 25 kHz sampling, human ear is not sensitive to higher frequencies
dT=1/sampling_frequency;

computational_time=tic;
% bar=waitbar(0,'analysis start');
if imposed
    instant=0;
    attachment=1;
    if pressing_time<2*dT
        sampling_frequency=10/pressing_time;
        dT=1/sampling_frequency;
    end

tspan=0:dT:pressing_time;
y0(bow_node_first_index)=[0 0 bow_velocity_z];
[tout,yout]=integration(tspan,y0,nodes,dof,doft,first,second,h,Erhoh,imposed,AE,
                  pressing_time,driving_time,bow_velocity_y,bow_velocity_z,attachment,bow_node,bow
                  _node_first_index,bow_node_second_index,v,mu,v_stick);

    if (T-pressing_time>2*dT)||((T-pressing_time==2*dT)
        tspan=tout(end):dT:T;
        y0=yout(end,:);
        y0(bow_node_first_index)=[0 bow_velocity_y 0];
        tout(end)=[];
        yout(end,:)=[];
        [tout2,yout2]=integration(tspan,y0,nodes,dof,doft,first,second,h,Erhoh,imposed,AE,
                pressing_time,driving_time,bow_velocity_y,bow_velocity_z,attachment,bow_node,bow
                _node_first_index,bow_node_second_index,v,mu,v_stick);
        tout=[tout;tout2];
        yout=[yout;yout2];
        clear tout2 yout2
    end
else
    if T<2*dT
        sampling_frequency=10/T;
    end
    dT=1/sampling_frequency;
    tspan=0:dT:T;
    [tout,yout]=integration(tspan,y0,nodes,dof,doft,first,second,h,Erhoh,0,0,0,0,0
                  ,0,0,0,0,0,0);
end
% close(bar);
computational_time=toc(computational_time); % integration elapsed time
% in case you want to resume integration
y0=yout(1,:);
tspan=tout(end);

velocities=yout(:,first);
yout(:,first)=[];
positions=yout;

bow_node_index=bow_node_first_index;

clear yout bar

%% misc

% bow node hourly law
bow_node_position=zeros(length(tout),4);
bow_node_position(:,1:3)=positions(:,bow_node_index);
bow_node_position(:,1)=bow_node_position(:,1)-bow_node_x;
bow_node_position(:,4)=(sum((bow_node_position(:,1:1:3)).^2,2)).^(0.5);

% bow node velocity
bow_node_velocity=zeros(length(tout),4);
bow_node_velocity(:,1:3)=velocities(:,bow_node_index);
bow_node_velocity(:,4)=(sum((bow_node_velocity(:,1:1:3)).^2,2)).^(0.5);

% bow force
if imposed
    [F_bow]=bow_force(positions,h,AE,bow_node_index);
end

% reaction forces
[F_bridge]=bridge_force(positions,h,AE);
8.10. Integration

```matlab
function [tout, yout] = integration(tspan, y0, nodes, dof, doft, first, second, h, Erhoh, imposed, AE, pressing_time, driving_time, bow_velocity_y, bow_velocity_z, attachment, bow_node, bow_node_first_index, bow_node_second_index, v, mu, v_stick) % T bar

[tout, yout] = ode45(@(f, tspan, y0);

function acc = f(t, y)
acc = zeros(doft, 1);
acc(second) = y(first); % identity equations related to velocities
y_dot = reshape(y(first), 3, nodes);
y = reshape(y(second), 3, nodes);
o = zeros(3, 1);

dsdx = ([y, o] - [o, y]) / h;
dsdx(:, [1 end]) = [];
c = sqrt(sum(dsdx.^2, 1));
tangent = dsdx ./ [c; c; c];
epsilon = log(c);

% damping correction
% damping = 1e-2;
% dv = ([y_dot, o] - [o, y_dot]);
% dv(:, [1 end]) = [];
% epsilon_dot = dv .* tangent;
% epsilon = epsilon + epsilon_dot;
% epsilon = [epsilon; epsilon; epsilon] .* tangent;
% acceleration = Erhoh * ([epsilon_t, o] - [o, epsilon_t]);
% acceleration(:, [1 end]) = [o, o];

acceleration = reshape(acceleration, dof, 1);
acc(first) = acceleration;

% imposed dof
% attachment_current = attachment;
o = o';
if imposed
if t < driving_time
    bow_node_free_acc = acc(bow_node_first_index);
    if attachment % stick friction
        acc(bow_node_first_index) = 0; % constant velocity
    end
    imposed motion
    if t < pressing_time;
        acc(bow_node_second_index) = [0 0 bow_velocity_z]; % string pressed in - z direction
    else
        acc(bow_node_second_index) = [0 bow_velocity_y 0];
```
\begin{verbatim}
s1=y(:,bow_node-1)';
s2=y(:,bow_node)';
s3=y(:,bow_node+1)';  % F node
s_l=s2-s1;
s_r=s2-s3;  % spacial gradients
l_l=sum(s_l.^2,2).^(0.5);
l_r=sum(s_r.^2,2).^(0.5);
t_l=s_l./[l_l l_l l_l];
t_r=s_r./[l_r l_r l_r];
eps_l=log(l_l/h);
eps_r=log(l_r/h);
F=AE*[eps_r eps_r eps_r.*t_r+eps_l eps_l eps_l.*t_l];
friction_value=abs(F(2)/F(3));

if isinf(friction_value)
  % singularities arise in imposed motion direction changes, they must be neglected
  if friction_value>1.2
    % static friction
    attachment=0;
    acc(bow_node_first_index(2))=bow_node_free_acc(2);
  end
else
  % slip friction
  acc(bow_node_first_index([1 3]))=[0 0];  % no motion in x and z directions
  acc(bow_node_second_index([1 3]))=[0 0];
  v_rel=acc(bow_node_second_index(2))-bow_velocity_y;
  if abs(v_rel)<v_stick
    attachment=1;
    acc(bow_node_first_index(2))=bow_velocity_y;
  else
    friction_value=interp1(v,mu,v_rel);
    acc(bow_node_first_index(2))=bow_node_free_acc(2)+(friction_value)*abs(bow_node_free_acc(3));
  end
end
else
  attachment=0;
end
end

end
\end{verbatim}
### 8.11. bridge_force

```matlab
function [F]=bridge_force(positions,h,AE)

s=positions(:,[4 5 6]); % distance vector from bridge node to second node
l=sum(s.^2,2).^(0.5); % first element length
t=s./[1 1 1]; % first element versor
eps=log(l/h); % first element strain
F=AE*([eps eps eps).*t);
F=[F sum(F.^2,2).^(0.5)];
```

### 8.12. bow_force

```matlab
function [F]=bow_force(positions,h,AE,bow_node_index)

s1=positions(:,bow_node_index-3);
s2=positions(:,bow_node_index); % F node
s3=positions(:,bow_node_index+3);

s_l=s2-s1; % left and right elements
s_r=s2-s3;

l_l=sum(s_l.^2,2).^(0.5);
l_r=sum(s_r.^2,2).^(0.5);
t_l=s_l./[l_l l_l l_l];
t_r=s_r./[l_r l_r l_r];

eps_l=log(l_l/h);
eps_r=log(l_r/h);
F=AE*([eps_r eps_r eps_r).*t_r+[eps_l eps_l eps_l).*t_l);
F=[F sum(F.^2,2).^(0.5)];
```
8.13. post_processing

% sound
if (~(sampling_frequency>1e9)
    bits=32;
    audiowrite('bridge_Fy.wav',1000*F_bridge(:,2),sampling_frequency,'BitsPerSample',bits,'Artist','Marco Bellante')
    clear bits
end

if imposed
    red=find((tout-pressing_time)==min(abs(tout-pressing_time)));
else
    red=find((tout-period1)==min(abs(tout-period1)));
end
red=1:1:(5*red);
if isempty(red)
    red=1:1:length(tout);
end

figure('units','normalized','outerposition',[0 0 1 1])

% time analysis

% bow node position
plot(tout(red),bow_node_position(red,1),'r-',tout(red),bow_node_position(red,2),'k-',tout(red),bow_node_position(red,3),'b-')
xlabel('time [s]')
ylabel('bow node position [m]')
legend('Sx','Sy','Sz')
title('bow node position')
saveas(gcf,'bow_node_S','bmp')

plot(tout,bow_node_position(:,1),'k-')
xlabel('time [s]')
ylabel('bow node position x [m]')
title('bow node position x')
saveas(gcf,'bow_node_Sx','bmp')

plot(tout,bow_node_position(:,2),'k-')
xlabel('time [s]')
ylabel('bow node position y [m]')
title('bow node position y')
saveas(gcf,'bow_node_Sy','bmp')

plot(tout,bow_node_position(:,3),'k-')
xlabel('time [s]')
ylabel('bow node position z [m]')
title('bow node position z')
saveas(gcf,'bow_node_Sz','bmp')
% bow node velocity
plot(tout(red),bow_node_velocity(red,1),'r-',tout(red),bow_node_velocity(red,2),'k-',tout(red),bow_node_velocity(red,3),'b-')
xlabel('time [s]')
ylabel('bow node velocity [m/s]')
legend('Vx','Vy','Vz')
title('bow node velocity')
saveas(gcf,'bow_node_V','bmp')

plot(tout,bow_node_velocity(:,1),'k-')
xlabel('time [s]')
ylabel('bow node velocity x [m/s]')
title('bow node velocity x')
saveas(gcf,'bow_node_Vx','bmp')

plot(tout,bow_node_velocity(:,2),'k-')
xlabel('time [s]')
ylabel('bow node velocity y [m/s]')
title('bow node velocity y')
saveas(gcf,'bow_node_Vy','bmp')

plot(tout,bow_node_velocity(:,3),'k-')
xlabel('time [s]')
ylabel('bow node velocity z [m/s]')
title('bow node velocity z')
saveas(gcf,'bow_node_Vz','bmp')

% bridge force
plot(tout(red),F_bridge(red,1)-N,'r-',tout(red),F_bridge(red,2),'k-',tout(red),F_bridge(red,3),'b-')
xlabel('time [s]')
ylabel('bridge force [N]')
legend('Fx-N','Fy','Fz')
title('bridge force')
saveas(gcf,'bridge_F','bmp')

plot(tout,F_bridge(:,1),'k-')
xlabel('time [s]')
ylabel('bridge force x [N]')
title('bridge force x')
saveas(gcf,'bridge_Fx','bmp')

plot(tout,F_bridge(:,2),'k-')
xlabel('time [s]')
ylabel('bridge force y [N]')
title('bridge force y')
saveas(gcf,'bridge_Fy','bmp')

plot(tout,F_bridge(:,3),'k-')
xlabel('time [s]')
ylabel('bridge force z [N]')
title('bridge force z')
saveas(gcf,'bridge_Fz','bmp')

if imposed
% bow force
plot(tout(red), F_bow(red,1), 'r.-', tout(red), F_bow(red,2), 'k.-', tout(red), F_bow(red,3), 'b.-')
xlabel('time [s]')
ylabel('bow force x [N]')
legend('Fx', 'Fy', 'Fz')
title('bow force')
saveas(gcf, 'bow_F', 'bmp')

plot(tout, F_bow(:,1), 'k.-')
xlabel('time [s]')
ylabel('bow force x [N]')
title('bow force x')
saveas(gcf, 'bow_Fx', 'bmp')

plot(tout, F_bow(:,2), 'k.-')
xlabel('time [s]')
ylabel('bow force y [N]')
title('bow force y')
saveas(gcf, 'bow_Fy', 'bmp')

plot(tout, F_bow(:,3), 'k.-')
xlabel('time [s]')
ylabel('bow force z [N]')
title('bow force z')
saveas(gcf, 'bow_Fz', 'bmp')
end

% PSD

frequency_range=[];
spectrogram(bow_node_position(:,1), [], [], [], [], [], [], [], [], [], sampling_frequency); title('bow node position x PSD')
saveas(gcf, 'bow_node_Sx_PSD', 'bmp')

spectrogram(bow_node_position(:,2), [], [], [], [], [], [], [], [], [], [], frequency_range, sampling_frequency); title('bow node position y PSD')
saveas(gcf, 'bow_node_Sy_PSD', 'bmp')

spectrogram(bow_node_position(:,3), [], [], [], [], [], [], [], [], [], [], [], sampling_frequency); title('bow node position z PSD')
saveas(gcf, 'bow_node_Sz_PSD', 'bmp')

spectrogram(bow_node_velocity(:,1), [], [], [], [], [], [], [], [], [], [], sampling_frequency); title('bow node velocity x PSD')
saveas(gcf, 'bow_node_Vx_PSD', 'bmp')

spectrogram(bow_node_velocity(:,2), [], [], [], [], [], [], [], [], [], [], frequency_range, sampling_frequency); title('bow node velocity y PSD')
saveas(gcf, 'bow_node_Vy_PSD', 'bmp')

spectrogram(bow_node_velocity(:,3), [], [], [], [], [], [], [], [], [], [], [], sampling_frequency); title('bow node velocity z PSD')
saveas(gcf, 'bow_node_Vz_PSD', 'bmp')
spectrogram(F_bridge(:,1),[],[],[],sampling_frequency)
title('bridge force x PSD')
saveas(gcf,'bridge_Fx_PSD','bmp')

spectrogram(F_bridge(:,2),[],[],[],frequency_range,sampling_frequency)
title('bridge force y PSD')
saveas(gcf,'bridge_Fy_PSD','bmp')

spectrogram(F_bridge(:,3),[],[],[],sampling_frequency)
title('bridge force z PSD')
saveas(gcf,'bridge_Fz_PSD','bmp')

if imposed
  spectrogram(F_bow(:,1),[],[],[],sampling_frequency)
title('bow force x PSD')
saveas(gcf,'bow_Fx_PSD','bmp')

  spectrogram(F_bow(:,2),[],[],[],frequency_range,sampling_frequency)
title('bow force y PSD')
saveas(gcf,'bow_Fy_PSD','bmp')

  spectrogram(F_bow(:,3),[],[],[],sampling_frequency)
title('bow force z PSD')
saveas(gcf,'bow_Fz_PSD','bmp')
end
close all

% spectrum

figure('units','normalized','outerposition',[0 0 1 1])
signalbellante(tout,bow_node_position(:,2));
title('bow node position y spectrum')
saveas(gcf,'bow_node_Sy_spectrum','bmp')

signalbellante(tout,bow_node_velocity(:,2));
title('bow node velocity y spectrum')
saveas(gcf,'bow_node_Vy_spectrum','bmp')

signalbellante(tout,F_bridge(:,2));
title('bridge force y spectrum')
saveas(gcf,'bridge_Fy_spectrum','bmp')

if imposed
  signalbellante(tout,F_bow(:,2));
title('F bridge force y spectrum')
saveas(gcf,'F_bridge_Fy_spectrum','bmp')
end
close
clear red
% animation
frame_skip=1;
if imposed

[bow_position]=bow_eom(tout,bow_node_x,pressing_time,bow_velocity_z,driving_time,
bow_velocity_y);
  film(tout,positions,bow_node,frame_skip,bow_position,imposed)
  clear bow_position
else
  film(tout,positions,bow_node,frame_skip,0,0)
end
clear frame_skip

8.14. bow_eom

function
[bow_position]=bow_eom(tout,bow_node_x,pressing_time,bow_velocity_z,driving_time,
bow_velocity_y)

number=length(tout);

bow_position=zeros(number,3);

depth=pressing_time*bow_velocity_z;

for a=1:1:number
  t=tout(a);
  if (t<pressing_time)||(t==pressing_time)
    bow_position(a,:)=[bow_node_x 0 bow_velocity_z*t];
  else
    if t<driving_time
      bow_position(a,:)=[bow_node_x bow_velocity_y*(t-pressing_time)
depth];
    end
  end
end
8.15. signal_bellante

function [spectrum]=signalbellante(t,y)

l=length(y); % samples number
T=t(l)-t(1); % sampling period [s]
dt=T/(l-1); % sampling time resolution [s]
f=1/dt; % sampling frequency [Hz]

y=y.*kaiser(l); % Kaiser-Bessel window
v=fft(y); % Matlab library fft transform
n=length(v);

if rem(n,2)==0 % even samples numbers
    spectrum=v(1:(n/2+1)); % Nyquist frequency at position (n/2)+1
    spectrum=spectrum*2/n; % normalization
    spectrum(1)=spectrum(1)/2; % average at position 1
else % odd samples numbers
    spectrum=v(1:ceil(n/2)); % Nyquist frequency position
    spectrum=spectrum*2/n; % normalization
    spectrum(1)=spectrum(1)/2; % average at position 1
end

f=0:(length(spectrum)-1); % frequency interval with resolution 1/T
f=f/T;

constant=2*pi;
omega=constant*f;
modulus=abs(spectrum);
rad=angle(spectrum);
deg=radtodeg(rad);

% discarding low amplitude frequencies
fr=f(2:end);
omegar=omega(2:end);
modulusr=modulus(2:end);
radr=rad(2:end);
degr=deg(2:end);

treshold=100;
treshold=max(modulusr)/treshold;
cut2=max(find(modulusr>treshold));
cut1=min(find(modulusr>treshold));
fr=f(cut1:cut2);
omegar=omega(cut1:cut2);
modulusr=modulus(cut1:cut2);
radr=rad(cut1:cut2);
degr=deg(cut1:cut2);

g%figure()
subplot(2,1,1)
semilogy(fr,modulusr,'k-') % modulus
xlabel('frequency [Hz]')
grid on
grid minor
subplot(2,1,2)
plot(fr,degr,'b-') % phase
xlabel('frequency [Hz]')
ylabel('phase [deg]')
grid on
grid minor
function []=film(tout,positions,bow_node,frame_skip,bow_position,imposed)

[rows,columns]=size(positions);

interval_x=1:3:(columns-2); % positions x component indexes
interval_y=2:3:(columns-1); % positions y component indexes
interval_z=3:3:columns; % positions z component indexes

% computing plot spacial limit
lt=positions(:,interval_x(end));
max_y=max(max(abs(positions(:,interval_y))));
max_z=max(max(abs(positions(:,interval_z))));
max_r=max([max_y max_z]); % plot radial limits

if imposed
    x=[bow_position(:,1) bow_position(:,1) bow_position(:,1) bow_position(:,1) bow_position(:,1)];
    z=[bow_position(:,3) bow_position(:,3) bow_position(:,3) bow_position(:,3) bow_position(:,3)];
    y=ones(rows,1)*max_r;
    y=[-6*y -4*y -2*y 0*y 0.3*y];
    y=y+[bow_position(:,2) bow_position(:,2) bow_position(:,2) bow_position(:,2) bow_position(:,2)];
end

% initializing movie
writerObj=VideoWriter('string','MPEG-4'); % video name
writerObj.FrameRate=100; % frames/s
open(writerObj);

% trajectory graphical representation
figure('units','normalized','outerposition',[0 0 1 1])
for a=1:frame_skip:rows
    if imposed
        plot3(x(a,:),y(a,:),z(a,:), 'b-v',positions(a,interval_x),positions(a,interval_y),positions(a,interval_z),'k-',[0 lt],[0 0],[0 0],'bo',positions(a,interval_x(bow_node)),positions(a,interval_y(bow_node)),positions(a,interval_z(bow_node)),'r*' ) % nodes plot
    else
        plot3(positions(a,interval_x),positions(a,interval_y),positions(a,interval_z),'k-',[0 lt],[0 0],[0 0],'bo',positions(a,interval_x(bow_node)),positions(a,interval_y(bow_node)),positions(a,interval_z(bow_node)),'r*' ) % nodes plot
    end
    axis([0 lt -max_r max_r -max_r max_r])
    xlabel('x [m]')
    ylabel('y [m]')
    zlabel('z [m]')
    title(num2str(tout(a))) % plot named after the step instant
    % legend('string','boundary conditions','bow node')
    % view(2); % view from z axis
    writeVideo(writerObj,getframe(gcf));
end
close(writerObj); % close and save the movie

9. CONCLUSIONS

When we chose this thesis activity, we were aware that we did not have much experimental data. We really missed information like non-linear elasticity and viscous damping, accurate friction curves for the different motion modes et cetera. Some of these missing parameters were given a “realistic” value when it was absolutely necessary for our analyses (for example, Shelleng’s parameters). Considering the limited information, we were still able to deliver a realistic simulation.

Considering analytical models, we were able to take into account many more aspects, like non-linearity or Friedlander curves.

Considering other numerical models that can be found in literature, mostly based on purely mathematical or electrical models, we were able to deliver a model with a more direct physical meaning.

This thesis activity was quite a creative job:

- We had to take a look into literature to understand the state of the art of analytical and numerical models.
- We had to think of ways to improve the most recent analytical models, considering the experimental data we had.
- We had to write a specific code able to achieve more accurate and reliable results with a faster convergence respect to a generic commercial FEM software. We did a “tailor’s job”.
- We had to do a “labor limae” on the code to increase computational efficiency. This was absolutely necessary in order to deliver results in a humane amount of time.

We really enjoyed this thesis activity, which was a mix between dynamics, analytical and numerical methods, programming et cetera.
# Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>violin</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>pegbox</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>violin strings</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>human ear</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>human hearing</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>violin player</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>player closed loop</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>Helmholtz corner</td>
<td>11</td>
</tr>
<tr>
<td>9</td>
<td>Schelleng diagram</td>
<td>12</td>
</tr>
<tr>
<td>10</td>
<td>Friedlander curve</td>
<td>13</td>
</tr>
<tr>
<td>11</td>
<td>physical model frame</td>
<td>16</td>
</tr>
<tr>
<td>12</td>
<td>infinitesimal string element</td>
<td>17</td>
</tr>
<tr>
<td>13</td>
<td>imposed motion</td>
<td>22</td>
</tr>
<tr>
<td>14</td>
<td>isoclinamic trajectory</td>
<td>25</td>
</tr>
<tr>
<td>15</td>
<td>isoclinamic trajectory, k dependancy</td>
<td>25</td>
</tr>
<tr>
<td>16</td>
<td>isoclinamic trajectory, beta depandancy</td>
<td>25</td>
</tr>
<tr>
<td>17</td>
<td>isoclinamic trajectory, current case</td>
<td>26</td>
</tr>
<tr>
<td>18</td>
<td>bow force</td>
<td>28</td>
</tr>
<tr>
<td>19</td>
<td>bridge force</td>
<td>28</td>
</tr>
<tr>
<td>20</td>
<td>first harmonic spacial initial condition</td>
<td>34</td>
</tr>
<tr>
<td>21</td>
<td>first harmonic bridge force</td>
<td>34</td>
</tr>
<tr>
<td>22</td>
<td>first harmonic bow node position</td>
<td>35</td>
</tr>
<tr>
<td>23</td>
<td>first harmonic bow node velocity</td>
<td>35</td>
</tr>
<tr>
<td>24</td>
<td>first harmonic bow node position x PSD</td>
<td>36</td>
</tr>
<tr>
<td>25</td>
<td>first harmonic bow node position y PSD</td>
<td>36</td>
</tr>
<tr>
<td>26</td>
<td>first harmonic bow node position y spectrum</td>
<td>37</td>
</tr>
<tr>
<td>27</td>
<td>first harmonic bow node velocity x PSD</td>
<td>38</td>
</tr>
<tr>
<td>28</td>
<td>first harmonic bow node velocity y PSD</td>
<td>38</td>
</tr>
<tr>
<td>29</td>
<td>first harmonic bow node velocity spectrum</td>
<td>39</td>
</tr>
<tr>
<td>30</td>
<td>first harmonic bridge force x PSD</td>
<td>40</td>
</tr>
<tr>
<td>31</td>
<td>first harmonic bridge force y PSD</td>
<td>40</td>
</tr>
<tr>
<td>32</td>
<td>first harmonic bridge force y spectrum</td>
<td>41</td>
</tr>
<tr>
<td>33</td>
<td>second harmonic spacial initial conditions</td>
<td>43</td>
</tr>
<tr>
<td>34</td>
<td>second harmonic bow node position</td>
<td>44</td>
</tr>
<tr>
<td>35</td>
<td>second harmonic bow node position x PSD</td>
<td>44</td>
</tr>
<tr>
<td>36</td>
<td>second harmonic bow node position y PSD</td>
<td>44</td>
</tr>
<tr>
<td>37</td>
<td>second harmonic bridge force</td>
<td>45</td>
</tr>
<tr>
<td>38</td>
<td>second harmonic bridge force x PSD</td>
<td>45</td>
</tr>
<tr>
<td>39</td>
<td>second harmonic bridge force y PSD</td>
<td>45</td>
</tr>
<tr>
<td>40</td>
<td>third harmonic spacial initial conditions</td>
<td>46</td>
</tr>
<tr>
<td>41</td>
<td>third harmonic bow node position</td>
<td>47</td>
</tr>
<tr>
<td>42</td>
<td>third harmonic bow node position x PSD</td>
<td>47</td>
</tr>
<tr>
<td>43</td>
<td>third harmonic bow node position y PSD</td>
<td>47</td>
</tr>
<tr>
<td>44</td>
<td>third harmonic bridge force</td>
<td>48</td>
</tr>
<tr>
<td>45</td>
<td>third harmonic bridge force x PSD</td>
<td>48</td>
</tr>
</tbody>
</table>
Figure 95  convergence ................................................................. 72
Figure 96  Figure 97  computational time fit ............................ 73
Figure 98  Helmholtz corner sequence ............................................. 78
Figure 99  Helmholtz corner bow node position ................................ 79
Figure 100  Helmholtz corner bow node velocity .................................. 79
Figure 101  Helmholtz corner bridge force ........................................... 80
Figure 102  Helmholtz corner bow force ............................................ 80
Figure 103  Helmholtz corner bow node position y ................................ 81
Figure 104  Helmholtz corner bow node velocity y ................................ 81
Figure 105  Helmholtz corner bow node position y PSD ....................... 82
Figure 106  Helmholtz corner bow node velocity y PSD ....................... 82
Figure 107  Helmholtz corner bridge force x ....................................... 83
Figure 108  Helmholtz corner bridge force y ....................................... 83
Figure 109  Helmholtz corner bridge force z ....................................... 83
Figure 110  Helmholtz corner bridge force x PSD ............................... 84
Figure 111  Helmholtz corner bridge force y PSD ............................... 84
Figure 112  Helmholtz corner bridge force z ....................................... 84
Figure 113  Helmholtz corner bow force x ......................................... 85
Figure 114  Helmholtz corner bow force y ......................................... 85
Figure 115  Helmholtz corner bow force z ......................................... 85
Figure 116  Helmholtz corner bow force x PSD ................................. 86
Figure 117  Helmholtz corner bow force y PSD ................................. 86
Figure 118  Helmholtz corner bow force z PSD ................................. 86
Figure 119  Schelleng diagram ..................................................... 87
Figure 120  code files ..................................................................... 89
11. REFERENCES

[1] “The acoustics of the violin: a review”
   by Jim Woodhouse, Reports on Progress in Physics (2014)


   by Helmholtz H V and Ellis A J, London, Longmans, Green, and Co. (1885)

   by Raman C V, Bull. Indian Assoc. Cultivation Sci. (1918)


[8] “Enhanced simulation of the bowed cello string”
   (Stockholm, Sweden) (2013)


    by Cronhjort A, KTH Vetenskap Och Konst Quarterly Progress and Status

Link: http://www.museodelviolino.org/laboratori-scientifici/laboratorio-politecnico-di-milano/