Modeling the marimba bar with free boundary conditions

Simon Andersen
sand22@student.aau.com

Aalborg University
Department of Architecture, Design and Media Technology
Sydhavn, Denmark

May 3^{rd}, 2024

Abstract

This report explores utilizing finite difference methods to extract the finite difference scheme of the damped ideal bar in order to recreate the excitation module of a marimba bar. In addition, it explores the clamped and free boundary conditions of the ideal bar for the implementation of the update equation. The results of the report shows the behavior of the clamped and free boundary conditions and update equation were successfully implemented.

1 Introduction

Finite difference(FD) methods are numerical techniques that can be used to approximate expressions in physics to differential equations by discretising them into a finite set of points [6]. Meaning the expressions can be implemented as software solutions making them valuable in areas such as physical modeling for sound synthesis. This is because the expressions can be utilised to simulate the behavior of acoustic systems and produce realistic audio. The foundation of this technique is made possible by the partial differential equations(PDEs) describing the propagation of sound waves in the mediums that make up musical instruments [6]. By discretising and numerically solving these PDEs as well as modeling the physical properties of an instrument, a realistic digital replica can be created. In addition to mimicking an instrument, the FD methods provide the ability to adjust the parameters of the physical models which ultimately broadens the expressiveness of the sound that it creates [6]. These parameters can i.e. be material properties, geometry and boundary conditions.

To expand on the importance of the boundary conditions, these are constraints applied to the solution of a differential equation. Using a finite difference scheme distributed in space, they describe how a system behaves at its boundaries or react with the modelled environment [6]. For example, the 1D wave equation's boundaries can be controlled using either the Dirichlet(fixed) or Neumann(free) boundary condition [6]. The end points should always stay zero with the Dirichlet condition but are able to travel freely with the Neumann condition. However, when exploring more complex systems the solution of the boundary conditions changes. In regards to the ideal bar's boundary conditions the choices are the clamped, simply supported or free boundary condition [6]. Now, the ideas of the conditions stay the same but the solutions are different. This report will explore the implementation of the free boundary condition for the ideal bar alongside how it can be related to modeling the marimba bar excitation module.

2 Analysis

2.1 Ideal bar equation

If we want to model the physical properties and propagation of sound waves of a marimba bar the easiest way to go about it is using the ideal bar equation [1]. Since the instrument essentially is constructed of wooden bars placed over tubes [3]. It's dynamics, distributed in time and space, are described by the following partial differential equation [6] (1)

$$\partial_t^2 = -\kappa^2 \partial_x^4 u \tag{1}$$

κ is the stiffness coefficient and $\partial_x^4 u$ is the 4th-order spatial derivative. By discretising the ideal bar's PDE (1) we get the ideal bar's finite difference(FD) scheme (2):

$$\delta_{tt} u_l^n = -\kappa^2 \delta_{xxxx} u_l^n \tag{2}$$

2.2 Physical properties

In equation (1) and (2) the term κ is present. This term accounts for the stiffness parameter of the system and is calculated by (3) [6]:

$$\kappa = \sqrt{\frac{EI}{\rho A}} \tag{3}$$

where ρ is material density, A is cross-sectional area, E is Young's modulus and I is moment of inertia.

In order to model the physical properties it is important to use all parameters in the equation which accounts for the behavior of i.e. the material and so on [6]. With the additional parameters the ideal bar equation becomes(4):

$$\rho A \partial_t^2 u = -EI \partial_x^4 u \tag{4}$$

and discretised to (5)

$$\rho A \delta_{tt} u = -EI \delta_{xxxx} u \tag{5}$$

2.3 Damping

In order to model the physics of an actual marimba we also have to add an expression which accounts for the energy loss in the system [6]. This is done by adding the damping term. The expression for damping is given by (6):

$$-2\sigma_0 \partial_t u + 2\sigma_1 \partial_t \partial_x^2 u \tag{6}$$

Where loss of energy is accounted for with the σ_0 expression, and the frequency dependant damping is accounted for with the σ_1 expression [6]. This expands the expression of the ideal bar equation with the following continuous PDE(7):

$$\rho A\partial_t^2 u = -EI\partial_x^4 u - 2\sigma_0\partial_t u + 2\sigma_1\partial_t\partial_x^2 u \tag{7}$$

and discretised to the FD scheme (8):

$$\rho A\delta_{tt}u_l^n = -EI\delta_{xxxx}u_l^n - 2\sigma_0\delta_t u_l^n + 2\sigma_1\delta_t\delta_{xx}u_l^n \tag{8}$$

2.4 Expansions

We can use finite difference operators to make approximations to the PDE's derivatives [6]. To generate an explicit finite difference scheme we'll use a centred time difference approximation for the σ_0 expression [6]. For the mixed derivative in the σ_1 expression the backward time difference will be used for the temporal derivative [6].

$$\rho A\delta_{tt}u_l^n = -EI\delta_{xxxx}u_l^n - 2\sigma_0\delta_{t.}u_l^n + 2\sigma_1\delta_{t-}\delta_{xx}u_l^n \tag{9}$$

Now expansions to each section can be applied for generating the update equation which is able to be implemented. The expansions for each section is given by (10, 11, 12 & 16).

$$\delta_{tt}u_l^n = \frac{1}{k^2}(u^{n+1} - 2u_l^n + u_l^{n-1}) \tag{10}$$

$$\delta_{xxxx}u_l^n = \frac{1}{h^4}(u_{l+2}^n - 4u_{l+1}^n + 6u_l^n - 4u_{l-1}^n + u_{l-2}^n) \tag{11}$$

$$\delta_{t.}u_l^n = \frac{1}{2k}(u_l^{n+1} - u_l^{n-1}) \tag{12}$$

2.4.1 Mixed derivative

$$\delta_{t-}\delta_{xx}u_l^n \tag{13}$$

A mixed derivative as seen in (13) can be approximated by applying each operator one by one in no specific order [6]. In this case, the backward time difference was applied first to give (14).

$$\delta_{t-}\delta_{xx}u_l^n = \frac{1}{k}(\delta_{xx}u_l^n - \delta_{xx}u_l^{n-1}) \tag{14}$$

and then the second-order difference in space to (15).

$$\delta_{t-}\delta_{xx}u_l^n = \frac{1}{k}(\frac{u_{l+1}^n - 2u_l^n - 2u_{l+1}^n}{h^2} - \frac{u_{l+1}^{n-1} - 2u_l^{n-1} - 2u_{l+1}^{n-1}}{h^2}) \tag{15}$$

and reduced to:

$$\delta_{t-}\delta_{xx}u_l^n = \frac{1}{kh^2}(u_{l+1}^n - 2u_l^n - 2u_{l+1}^n - u_{l+1}^{n-1} - 2u_l^{n-1} - 2u_{l+1}^{n-1}) \tag{16}$$

2.5 Update equation

With the expansions made, these could be substituted into the equation (17).

$$
\begin{aligned}
\rho A \frac{1}{k^2}(u_l^{n+1} - 2u_l^n + u_l^{n-1}) = \\
&- \frac{EI}{h^4}(u_{l+2}^n - 4u_{l+1}^n + 6u_l^n - 4u_{l-1}^n + u_{l-2}^n) \\
&- \frac{2\sigma_0}{k}(u_l^{n+1} - u_l^{n-1}) \\
&+ \frac{2\sigma_1}{kh^2}(u_{l+1}^n - 2u_l^n + u_{l-1}^n - u_{l+1}^{n-1} + 2u_l^{n-1} - u_{l-1}^{n-1})
\end{aligned}
\tag{17}
$$

Solving for u_l^{n+1} yields (18).

$$
\begin{aligned}
u_l^{n+1} = \\
&2u_l^n - u_l^{n-1} + \frac{k^2}{\rho A} \\
&(-\frac{EI}{h^4}(u_{l+2}^n - 4u_{l+1}^n + 6u_l^n - 4u_{l-1}^n + u_{l-2}^n) \\
&- \frac{2\sigma_0}{k}(u_l^{n+1} - u_l^{n-1}) \\
&+ \frac{2\sigma_1}{kh^2}(u_{l+1}^n - 2u_l^n + u_{l-1}^n - u_{l+1}^{n-1} + 2u_l^{n-1} - u_{l-1}^{n-1}))
\end{aligned}
\tag{18}
$$

The equation is implicit caused by the centered time difference operator. To make the equation explicit we can introduce the factor $2\sigma_0 \frac{k}{\rho A}$ to remove the unnecessary x_l^{n+1} and rearrange accordingly which gives the final update equation for the system [6] (19):

$$
\begin{aligned}
u_l^{n+1} = \\
&\frac{2u_l^n - u_l^{n-1}}{1 + 2\sigma_0 \frac{k}{\rho A}} + \frac{k^2}{\rho A} \\
&(-\frac{EI}{h^4}(u_{l+2}^n - 4u_{l+1}^n + 6u_l^n - 4u_{l-1}^n + u_{l-2}^n) \\
&+ \frac{2\sigma_1}{kh^2}(u_{l+1}^n - 2u_l^n + u_{l-1}^n - u_{l+1}^{n-1} + 2u_l^{n-1} - u_{l-1}^{n-1}))
\end{aligned}
\tag{19}
$$

2.6 Boundary Conditions

Because of the 4th-order spatial derivative of the damped ideal bar system, two virtual grid points needs to be handled at the boundaries [6]. The stencil of the damped ideal bar system is shown on figure 1. It becomes clear each boundary require at least 2 virtual grid points. Specifically 2 at each side and 4 in total for the current time-step. As well as 1 at each side or 2 in total for the previous time-step. The virtual grid points to be accounted for are the marked points on figure 1. 3 boundary conditions are available for the ideal bar. This includes:

- clamped: $u_l^n = \delta_{x\pm} u_l^n = 0$
- simply supported: $u_l^n = \delta_{xx} u_l^n = 0$
- free: $\delta_{xx} u_l^n = \delta_{x\cdot}\delta_{xx} u_l^n = 0$

This report will only focus on the clamped and free boundary conditions.

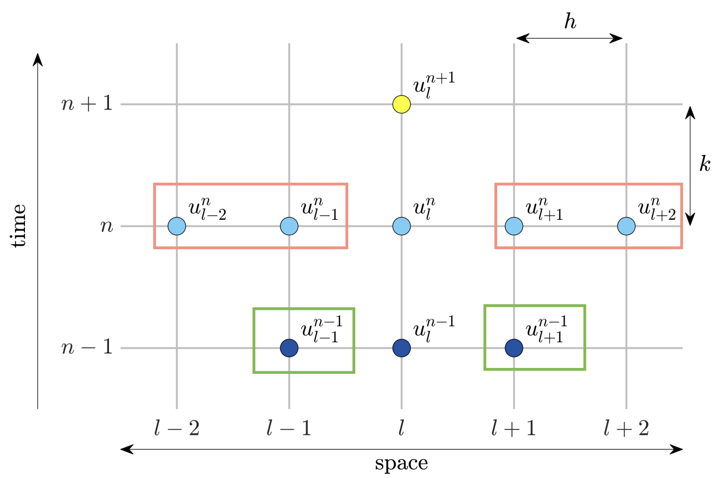


Figure 1: Ideal bar stencil: The red squares marks the current time step's virtual grid points and the green marks the previous time step's virtual grid points (courtesy of [6] and modified)

2.6.1 Clamped boundary

Expanding the operators of the FD scheme of the clamped boundary condition shows each boundary should stay 0 at all times [6]. This can be accounted for in the implementation by reducing the range of calculation to $l \in \{2, ..., N-2\}$ and initializing the boundaries to 0 [6].

2.6.2 Free boundary

The free boundary condition is often related to the ideal bar [6]. In comparison to the clamped condition, the free requires all points to be calculated which means the range stays $l \in \{0, ..., N\}$ [6]. As each respective boundary require at least 2 virtual grid points, these are defined as: u_{-1}^n & u_{-2}^n for the left side and u_{N+1}^n & u_{N+2}^n for the right. Using the operators to expand the FD scheme for the boundaries we get the following definitions:

$$
\begin{aligned}
u_{-1}^n &= 2u_0^n - u_1^n \\
u_{-2}^n &= u_2^n - u_1^n + 2u_{-1}^n \\
u_{N+1}^n &= 2u_N^n - u_{N-1}^n \\
u_{N+2}^n &= u_{N-2}^n - 2u_{N-1}^n + 2u_{N+1}^n
\end{aligned} \tag{20}
$$

2.7 Parameters

With the update equation and boundary conditions defined we can take a look at the physical attributes of modelling the sound of a marimba. These attributes will result in the parameters being used in the implementation. The bars residing on the marimba are either made out of wood or a synthetic fiberglass material [4]. Rosewood is often used to get the most desirable sound of the instrument [4]. The density or ρ (kg/m3) of rosewood is around 800-880 [5] and the Young's modulus or E is around 2.45387e13 [2]. The marimba bars or keys comes in different lengths that each represent different a pitch. Taking the bar of C3 pitch the length is around 440 mm. Now, A was defined as the cross sectional area and I which depends on the geometry of the bar cross-section [1]. Silvin Willemsen, PostDoc at TU/e, stated that these parameters would be difficult to fine-tune, so he provided initial values to start from which included: $A = 7.85e - 07$ and $I = 4.91e - 14$. The same was proposed for the damping terms which were $\sigma_0 = 0.2$ and $\sigma_1 = 0.0697$.

3 Implementation

The implementation of the damped ideal bar is divided into two categories. Namely, parameter initialization and time loop.

3.1 Parameter initialization

The physical and damping parameters was entered for each attribute. As well as the system properties defining the length of calculation, spatial and temporal step sizes, and array initialization. For the excitation of the system, a hanning window was used to apply the initial energy.

3.2 Time loop

The time loop is constructed by two `for loops`. The first one is accounting for the sample steps and the second is for the spatial and temporal steps. Since both the clamped and free boundary conditions were implemented they required different calculation ranges.

3.2.1 Clamped boundary condition

To account for the virtual grid points, that in the clamped condition should remain at a value of zero, the range of calculation was shortened to `for i = (3:N-1)`.

This meant the virtual grid points were accounted for and could run the update equation throughout the loop and update the previous and current time step accordingly.

3.2.2 Free boundary condition

To account for the virtual grid points in the free boundary condition the virtual grid points should be calculated and used within the update equation. The range of calculation was set to `for i (1:N)` since all points had to be calculated. To introduce the calculation of each virtual grid point at the ends, an `if` statement was used. This determined, when the time step was at either end, the virtual grid points should be calculated. So, at the left side when `l==1`, both u^n_{-1}, u^n_{-2}, and $uPrev^n_{-1}$ was calculated and substituted into the update equation. But when `l==2`, only u^n_{-1} needs to be calculated and substituted. The same went for the right side. When `l==N+1`, then u^n_{N+1}, u^n_{N+2}, and $uPrev^n_{N+1}$ had to be calculated and substituted but only u^n_{N+1} when `l==N`.

The substitution was implemented by adding a variable for storing the calculated grind points and adding each version of the update equation that used the variable. I.e. the virtual grid points was saved to `uRight_1` and `uRight_2` and substituted respectively with `u(l+1)` and `u(l+2)` when `l==1` and so on.

The fully implemented update equation can be seen in appendix A listing 1 and the statement surrounding the free boundary condition can be seen listing 2.

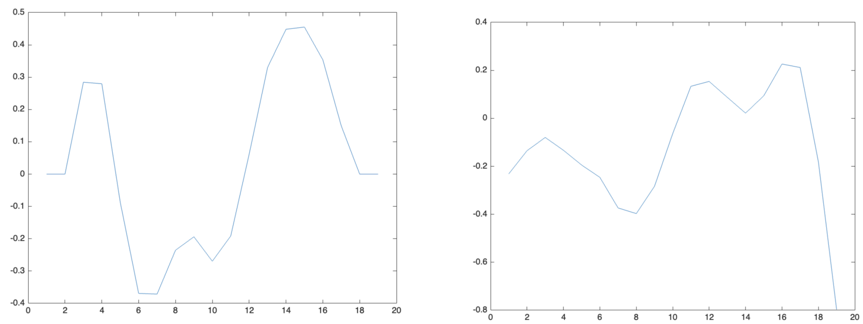
4 Results

The behavior of both clamped and free boundaries was plotted during runtime using the `plot()` and `drawnow` function in MATLAB and can be seen on figure 2. As well as the actual signal plotted in MATLAB on figure 3.

5 Discussion

The sound produced by both boundary conditions is very similar and can be characterised by a hybrid version of the xylophone's metal bar and the marimba wooden bar. Despite that, the two graphs on figure 2 clearly shows the expected behavior of the free and clamped boundary conditions within the system. A Butterworth filter was also applied to the signal to reduce some of the metallic 'clink' to preview the wooden body of the sound.

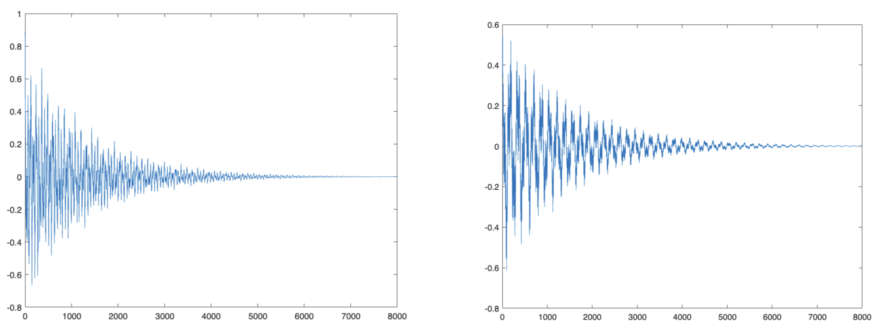
To comprehensively model the behavior of the marimba, additional factors must be taken into account. The marimba is constructed of multiple tone plates with metal resonator pipes affixed underneath [3]. Whereas, the exciter of the instrument is the interaction of the wooden bar and a mallet. The length of the



(a) Clamped boundary

(b) Free boundary

Figure 2: Damped ideal bar with clamped(top) and free(bottom) boundaries



(a) Clamped boundary signal

(b) Free boundary signal

Figure 3: Damped ideal bar with clamped(top) and free(bottom) boundaries

resonator pipe is also dependant on the pitch of the tone plate. The lower the note, the longer pipe and bar [3]. If the resonator pipe is not present in both the real instrument and digital system, the expected sound would just be a soft popping sound [3], which in fact, is the resulting sound of the implemented version.

For further development a mass-spring mallet model could be coupled together with the implemented ideal bar model for the complete excitation and an acoustic tube connection for the resonator part. Alongside accounting for the bars of varying cross-sectional area [1].

6 Conclusion

The aim of this report was to physical model the marimba bar excitation using the damped ideal bar with clamped and free boundary condition. The results show that both boundary conditions were effectively implemented as well as the update equation for the ideal bar. Ultimately, concluding the task of the report. However, the actual sound produced by the system is quite metallic sounding which does not fit with the expected behavior and would need some fine-tuning adjustments to fully shape the sound.

A Code examples

Listing 1: Update equation

```
uNext(l) = (2 * u(l) - uPrev(l)) / (1 + 2 * ...
sigma0 * (k / (rho * A))) + (k^2 / (rho * A)) ...
* (- (E * I/h^4) * (u(l+2) - 4*u(l+1) + 6*u(l) ...
- 4*u(l-1) + u(l-2)) + (2 * sigma1 / k * h^2) ...
* (u(l+1) - 2 * u(l) + u(l-1) - uPrev(l + 1) + 2  ...
* uPrev(l) - uPrev(l - 1)));
```

Listing 2: Update eq. & virtual grid points

```
for l = 1:N+1
    if l == 1 % left side
        uLeft_1 = 2 * u(1) - u(2);
        uLeft_2 = u(3) - 2 * u(2) + 2 * uLeft_1;
        uPrevLeft = 2 * uPrev(1) - uPrev(2);
        uNext(l) = ...
    elseif l == 2 % left side
        uLeft_1 = 2 * u(1) - u(2);
        uNext(l) = ...
    elseif l == N+1 % right side
        uRight_1 = 2 * u(N+1) - u(N);
        uRight_2 = u(N-1) - 2 * u(N) + 2 * uRight_1;
        uPrevRight = 2 * uPrev(N+1) - uPrev(N);
        uNext(l) = ...
    elseif l == N % right side
        uRight_1 = 2 * u(N+1) - u(N);
        uNext(l) = ...
    else
        uNext(l) = ...
    end
end
```

References

[1] Stefan Bilbao. *Numerical sound Synthesis: Finite Difference Schemes and simulation in musical Acoustics.* John Wiley & Sons, September 2009. ISBN

978-0-470-74902-9. tex.googlebooks: 3q5nGRd4UysC.

[2] Modalys Editorial. Material properties for wooden bars - modalys, unknown. URL https://support.ircam.fr/docs/Modalys/current/Objects/ObjectProperties/object_properties_wood.html.

[3] Yamaha Editorial. The structure of the marimba:inside and outside the resonator pipes - musical instrument guide - yamaha corporation, unknown. URL https://www.yamaha.com/en/musical_instrument_guide/marimba/mechanism/mechanism002.html.

[4] Efraín Figueroa Lemus. Wood, the essential element in the mesoamerican marimba, Aug 2019. URL https://www.odiseascentroamericanas.com/single-post/2019/08/10/wood-the-essential-element-in-the-mesoamerican-marimba.

[5] Matmatch. Density of wood in kg/m3, g/cm3, lb/ft3 – the ultimate guide - matmatch, unknown. URL https://matmatch.com/learn/property/density-of-wood.

[6] Silvin Willemsen. The emulated ensemble: Real-time simulation of musical instruments using finite-difference time-domain methods. *unknown*, 2021.