A Real-Time Interactive Physical Model of the Langeleik using Finite Difference Schemes and Web Audio

David Südholt (dsudho20@student.aau.dk) [1], **Søren V. K. Lyster** (slyste20@student.aau.dk) [1], **Oliver B. Winkel** (owinke17@student.aau.dk) [1], and **Stefania Serafin** (sts@create.aau.dk) [2]

[1] *CREATE*, **Aalborg University**, Copenhagen, Denmark
[2] *Multisensory Experience Lab, CREATE*, **Aalborg University**, Copenhagen, Denmark

ABSTRACT

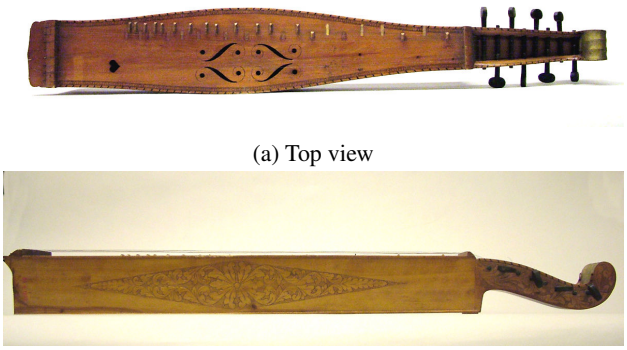
The Langeleik is a stringed folk instrument consisting of one fretted melody string and a number of drone strings. In this paper we present the use of finite difference schemes (FDS) to implement a physical model of the Langeleik as a real-time, interactive web application using JavaScript and Web Audio. Hammer-on/pull-off interactions are modeled using non-iterative collision methods. The user interacts with the model by strumming with a mouse or touchpad, and pressing the frets with a computer keyboard. According to a qualitative evaluation of the application, the sound quality is satisfactory, but the design of the interaction can be improved to provide a more intuitive experience.

1. INTRODUCTION

The Langeleik (pictured in figure 1) is a traditional stringed folk instrument. It consists of a melody string and a number of drone strings. Players strum the strings with a plectrum to create a rhythm and play a melody on one string only, using the hammer-on/pull-off technique. The Langeleik has historically been played in various northern European cultures, but is most closely associated with Norway [1]. While Langeleiks are being built and played in Norway to this day [1], in most other places they are rarely found outside of museums.

By creating a physical model of the Langeleik that is playable in real-time in a web browser, this paper aims to provide an accessible way of interacting with this traditional instrument.

Sound synthesis based on physical models of instruments has a long history. While the efficiency of digital waveguides [3] made them a popular choice for a long time, the more computationally demanding finite difference schemes (FDS) offer a close correspondence between model parameters and physical characteristics of the instrument, and impose few restrictions on the nature of the modeled behavior. This makes them a powerful tool to simulate a variety of different interactions [4–6], including those that go beyond what is possible in the physical world.



(a) Top view

(b) Side view

Figure 1: Photographs of a Langeleik [2]

In the FDS method, the physical behavior of the instrument is described with partial differential equations (PDE), which are then discretized and solved numerically using time-stepping methods.

The rapid increase in computing power in consumer electronics has made it possible to use FDS methods in real-time to construct interactive physical models of instruments. While it is common to develop these applications in C++ to take advantage of the performance benefits of compiled code, this results in an application that users need to download and execute on their device locally. To make the instrument model accessible to a broad audience with as little friction as possible, we choose to implement the physical model using JavaScript Web Audio, allowing it to be played in a web browser at the expense of optimal performance.

In section 2, we present the physical model of the Langeleik. Section 3 shows the methods used to discretize it. Section 4 describes the design and implementation of the interactive web application. We discuss the results of a qualitative evaluation in section 5. Finally, section 6 contains concluding remarks and future work.

[1] Performance example: https://www.youtube.com/watch?v=-i9h28DcTV0

2. INSTRUMENT MODEL

Our model of the Langeleik considers the strings and the body separately. We describe plucking and hammer-on/pull-off interactions for the strings and interaction with the resonating body.

2.1 Stiff Strings

The vertical displacement $u(x,t)$ of a point $x \in [0, L]$ along a damped stiff string of length L at time $t \geq 0$ can be described by the PDE [7]

$$\mathcal{L}_s u = 0\,, \tag{1}$$

where the operator $\mathcal{L}_s$ is defined as follows, using ∂_t and ∂_x to denote partial differentiation with respect to t and x:

$$\mathcal{L}_s = \rho A \partial_t^2 - T\partial_x^2 + EI\partial_x^4 + 2\sigma_0\rho\partial_t - 2\sigma_1\rho\partial_t\partial_x^2\,. \tag{2}$$

Here, σ_0 is a general and σ_1 a frequency-dependent damping parameter, and E is Young's modulus, describing the material stiffness of the string. T refers to the string tension, and ρ to its density. Given the string radius r, $A = \pi r^2$ is the cross-sectional area of the string and $I = \pi r^2/4$ is the moment of inertia.

The wave speed c is related to these physical parameters as $c = \sqrt{T/(\rho A)}$. The boundary conditions follow the simply supported model, such that

$$u = \partial_x^2 u = 0 \quad \text{for} \quad x = 0, L\,. \tag{3}$$

2.2 Plucking

In our model of the Langeleik, each string is individually excitable. A single pluck of a string at location χ_e and starting at time t_0 with maximum force $f_{e,\max}$ may be modeled according to [8] by applying a force $f_e(t)$ for a duration t_e, defined as

$$f_e(t) = \begin{cases} \frac{f_{e,\max}}{2}\left(1 - \cos\left(\frac{t-t_0}{t_e}\pi\right)\right) & t_0 \leq t \leq t_0 + t_e \\ 0 & \text{else} \end{cases} \tag{4}$$

The force model is incorporated into the string model by modifying equation (1) as follows:

$$\mathcal{L}_s u = \mathcal{F}_e\,, \text{ where } \mathcal{F}_e(x,t) = \begin{cases} f_e(t) & x = \chi_e \\ 0 & \text{else} \end{cases} \tag{5}$$

2.3 Fretting, Hammer-ons, Pull-offs

Equation (5) fully models the drone strings of the Langeleik. The melody string however requires consideration of fretting and hammer-on/pull-off interactions. We model the hammer-on through the collision of a finger descending and pressing down on the string. The finger simultaneously acts as excitation force and fret, modifying the string's effective length and thus its pitch while the finger is pressed down. The pull-off interaction is modeled by removing the finger instantaneously, which leads to a renewed excitation of the string snapping back due to its tension.

We introduce a point mass representing a finger for each of the N_f frets. Given the fret locations $\tau_1, \ldots, \tau_{N_f} \in [0, L]$, the i-th point mass interacts with the string at $x = \tau_i$. Its vertical displacement $w_i(t)$ can then be expressed as [4]

$$\mathcal{L}_f w_i = f_{c,i} - f_{p,i} \quad \text{where} \quad \mathcal{L}_f = M_i \frac{d^2}{dt^2}\,. \tag{6}$$

Here, M_i denotes the mass of the i-th finger. $f_{p,i}$ refers to the force with which the players presses the i-th finger down. To model the force $f_{c,i}$, which describes the collision of the i-th finger with the string, we introduce a potential $\phi_i(\eta_i)$ [4]:

$$\phi_i(\eta_i) = \frac{K}{\alpha + 1}\left[\eta_i\right]_+^{\alpha+1}\,. \tag{7}$$

Given the stiffness K of the collision and a nonlinearity coefficient α, the potential ϕ_i penalizes interpenetration of the i-th finger and the string depending on their distance $\eta_i(t) = u(\tau_i, t) - w_i(t)$. Using the operator $[\eta_i]_+ = \max(\eta_i, 0)$ to refer to the positive part of η_i, the potential is non-zero if and only if $w_i(t) < u(\tau_i, t)$, i.e. when the finger displacement is below the displacement of the string at the corresponding fret location.

The collision force $f_{c,i}$ can then be defined as the derivative of the potential:

$$f_{c,i} = \phi_i' = \frac{d}{d\eta_i}\phi_i \tag{8}$$

Equation (5) can then be modified to take the interaction into account:

$$\mathcal{L}_s u = \mathcal{F}_e - \mathcal{F}_c\,, \tag{9}$$

where the collision model $\mathcal{F}_c$ is defined as

$$\mathcal{F}_c(x,t) = \begin{cases} f_{c,i}(t) & x = \tau_i \text{ for } i = 1, \ldots, N_f \\ 0 & \text{else} \end{cases} \tag{10}$$

2.4 Body

Following the approach used in [5], we model the body as a two-dimensional plate. The body model is however not part of the real-time instrument model. Instead, we excite it with an impulse at approximately the location of the bridge, generating an impulse response (IR) in advance, which can then be used in convolution reverberation (cf. section 4) to simulate the resonance of the body.

2.5 Complete Instrument

Our real-time model of the Langeleik consists of 7 independent and uncoupled drone strings modeled according to equation (5). An additional melody string is interacting with N_f point masses according to the following system:

$$\begin{cases} \mathcal{L}_s u = \mathcal{F}_e - \mathcal{F}_c & \text{(11a)} \\ \mathcal{L}_f w_i = f_{c,i} - f_{p,i} & \text{for } i = 1, \ldots, N_f \quad \text{(11b)} \\ \eta_i = u(\tau_i, t) - w_i(t) & \text{for } i = 1, \ldots, N_f \quad \text{(11c)} \end{cases}$$

3. DISCRETIZATION

We discretize our model using FDTD methods. We will first consider equation (5), describing the drone strings, and then incorporate the interaction with the melody string described in system (11) by modelling hammer-ons, pull-offs and fretting user finger-string collisions.

3.1 Drone Strings

We introduce the grid function $u_l^n \approx u(x,t)$, which is a discrete approximation to the continuous string model. The relationship between the time index n and continuous time is given by $t = nk$ with $n \geq 0$, where k is the size of the time step in seconds, usually externally given by the audio sampling rate f_s as $k = 1/f_s$.

The grid location l is related to continuous space by $x = lh$, where h is the distance between grid points in meters and $l \in [0, \ldots, N_x]$ with $N_x = L/h$ being the total number of grid points.

To obtain the discrete approximation ℓ_s of the operator $\mathcal{L}_s$, partial derivatives in time and space are approximated through finite differences. These methods are described in detail in [7].

Numerical stability requires $h \geq h_{\min}$. In the case of the stiff string, $h_{\min}$ can be shown to be [7]:

$$h_{\min} = \sqrt{\frac{k}{2}\left(\frac{Tk}{\rho A} + 4\sigma_1 + \sqrt{\left(\frac{Tk}{\rho A} + 4\sigma_1\right)^2 + \frac{16EI}{\rho A}}\right)} \tag{12}$$

For the most accurate approximation permitted by this stability condition, N_x is chosen as $N_x = \lfloor L/h_{\min} \rfloor$, from which the grid spacing follows as $h = L/N_x$.

The discrete plucking force f_e^n can be obtained by simply sampling the continuous force model given in equation (4) in time. To apply the force at a particular location to the corresponding grid points, we introduce the linear spreading operator $J_l(x_c)$, which is used to apply a force to a particular location x_c between two grid points $l_c = \lfloor x_c/h \rfloor$ and $l_c + 1$. Writing $\alpha_c = x_c/h - l_c$, the spreading operator for a particular grid index l is given by [7]

$$J_l(x_c) = \frac{1}{h}\begin{cases} 0 & l < l_c \\ (1-\alpha_c) & l = l_c \\ \alpha_c & l = l_c + 1 \\ 0 & l > l_c + 1 \end{cases} . \tag{13}$$

The discrete counterpart of the string model from equation (5) is then given by

$$\ell_s u_l^n = J_l(\chi_e) f_e^n . \tag{14}$$

By expanding ℓ_s, an explicit update equation for u_l^{n+1} can be obtained from equation (14). The drone strings can then be modelled by simulating the desired number of grids independently in parallel.

3.2 Melody String

Analogously to the spreading operator defined in equation (13), we introduce the linear interpolation operator $I(x_c)u^n = (1-\alpha_c)u_{l_c}^n + \alpha_c u_{l_c+1}^n$. With ℓ_f denoting the discrete approximation of $\mathcal{L}_s$, system (11) can be discretized as

$$\begin{cases} \ell_s u_l^n = J_l(\chi_e) f_e^n - \sum_{i=1}^{N_f} J_l(\tau_i) f_{c,i}^n & \text{(15a)} \\ \ell_f w_i^n = f_{c,i}^n - f_{p,i}^n \quad \text{for } i = 1, \ldots, N_f & \text{(15b)} \\ \eta_i^n = I(\tau_i) u^n - w_i^n \quad \text{for } i = 1, \ldots, N_f & \text{(15c)} \end{cases}$$

The force $f_{p,i}(t)$ representing the player pressing the fingers down is an external input to the system and can simply be sampled in time to obtain $f_{p,i}^n$.

The discretization $f_{c,i}^n$ of the continuous collision force $f_{c,i}(t)$ is less straightforward. $\phi_i'(\eta_i)$ can be rewritten to $\dot{\phi}_i/\dot{\eta}_i$ using the chain rule, but as demonstrated in [9], directly applying finite difference methods to approximate these time derivatives results in implicit update equations that require iterative solvers.

The same paper also proposes an approach to arrive at a non-iterative discretization, avoiding the computational cost associated with iterative solvers. We employ this approach for the collision calculation in our model. Assuming that the frets are sufficiently distant from one another that their associated grid points do not overlap, this reduces the collision calculation to solving a full-rank system of two linear equations for each finger, which can be done explicitly. The detailed derivation is shown in appendix A. Another example of using this approach in the context of a real-time physical model can be seen in [5].

3.3 Interaction with the Model

It remains to answer the question of how the input force $f_{p,i}^n$ representing the player pressing down on the i-th finger should be set. In our model, each finger is given a target velocity v_i^{target} at which it should hit the string. When the hammer-on is initiated at time step n_0, the finger will start moving downward from its initial position $w_i^{n_0} = w_i^0$ somewhere above the string. Once the finger reaches a defined displacement $w_i^{\text{target}} < 0$, we set $v_i^{\text{target}} = 0$, indicating that the finger should stay at that displacement, pressing down on the string.

At time steps $n > n_0$ we calculate the update equation w_i^{n+1} and approximate the finger's current velocity as

$$v_i^n = \frac{w_i^{n+1} - w_i^{n-1}}{2k} . \tag{16}$$

According to Newton's second law of motion, the force needed to change the finger's velocity from v_i^n to v_i^{target} over the course of one time step may be expressed as

$$M_i \frac{v_i^n - v_i^{\text{target}}}{k} . \tag{17}$$

Additionally, the collision force $f_{c,i}^n$ is pushing the finger upward once it makes contact with the string. To keep the finger's velocity approximately constant while it is descending and to keep it stationary while pressing down at the target displacement, we add an opposite force to the

finger at the next time step. These two forces are combined to calculate $f_{p,i}^{n+1}$, acting negatively on the finger at the next time step:

$$f_{p,i}^{n+1} = f_{c,i}^{n} + M_i \frac{v_i^n - v_i^{\text{target}}}{k} \tag{18}$$

The finger mass, the target velocity and target displacement are model parameters that can be adjusted to affect the sound of the hammer-on excitation.

4. IMPLEMENTATION

This section describes the implementation of an interactive real-time simulation of the Langeleik in a web browser. It has been tested for desktop computers using the Chrome browser[2] and is accessible on `https://smc804.github.io`. The code is publicly available at `https://github.com/SMC804/SMC804.github.io`.

The JavaScript Web Audio API [10] provides an interface to design custom audio nodes. This requires the implementation of an `AudioWorkletProcessor`, which exposes a `process` method, so that it can form part of the audio routing graph. Our implementation makes use of custom nodes for the FDS calculations (and combines them with predefined Web Audio nodes to further process the synthesized audio from the FDS. A graphical user interface (GUI) written in JavaScript enables real-time interaction with the model.

4.1 String Nodes

Our implementation introduces two custom processing nodes, which are defined to have one channel of input and one channel of output. The input value at time step n is interpreted to be the excitation force f_e^n, which affects a position along the string determined by the mouse position using linear interpolation, and the output channel contains the audio generated by "listening" to the string at an externally defined position.

The `StringProcessor` implements the FDS of a stiff string under tension as defined in equation (14), and the `MelodyStringProcessor` inherits the `StringProcessor` class to add the hammer-on/pull-off interaction to the stiff string FDS.

To allow the user to control the hammer-on/pull-off interaction, we introduce a finger for each fret that initially hovers over the string and presses down when the user presses the fret. This causes both an initial hammer-on excitation and a continuous pitch shift of the string corresponding to shortening it to the distance between the bridge and the fret while the finger is pressed down.

When the user releases the fret, the finger is instantly "teleported" back to its starting position by rewriting its saved previous positions and values for η. The sudden release of the string causes an excitation imitating the pull-off interaction.

[2] Using non-Chromium based browsers such as Firefox currently results in corrupted audio quality.

Name	Symbol (unit)	Value
Young's modulus	$E\ (Pa)$	$180 \cdot 10^9$
Radius	$r\ (m)$	$4.6 \cdot 10^{-3}$
Material density	$\rho\ (kg \cdot m^{-3})$	6000
Overall damping	$\sigma_0\ (s^{-1})$	1.38
Freq.-dependent damping	$\sigma_1\ (m^2 \cdot s^{-1})$	$1.3 \cdot 10^{-4}$

Table 1: Physical parameters of the Langeleik strings.

The seven drone strings and the melody string are simulated in parallel and mixed together. The synthesized audio is convolved with an IR previously generated from a model of the body to simulate its resonance.

4.2 Interface

Playing the Langeleik requires two types of interaction: plucking and fretting. To approximate these interactions in a web browser, using only standard available human-computer interfaces, we decided to use mouse/touchpad interaction for strumming, which is defined as plucking multiple strings with the same motion, and keyboard interaction for fretting. The interface is depicted in figure 2.

The interface is implemented using HTML, CSS, and JavaScript. The strings of the Langeleik are animated to visualize the strumming of strings. Frets are drawn on the melody string with their corresponding notes, and an indicator is shown when a fret is pressed.

The strum force $f_{e,\max}$ as defined in equation (4) is based on the duration between the `onmouseenter` and `onmouseleave` HTML DOM events, so that the loudness of the pluck is effectively controlled by the speed of the user's mouse.

4.3 Physical Parameters and Tuning

The physical parameters of the strings have been approximated by gathering information from different sources [1,2,11], and by perceptually fine-tuning the parameters to match performances accessible online. There appear to be no Langeleik measurements or tunings that could be characterized as "standard". The selected physical parameters are shown in table 1.

The drone strings are tuned to, in sequential order, A3, A3, A3, E4, A4, C#4 and E4, with a reference pitch of A4 $= 440$Hz. A desired fundamental frequency f_0 for a given string can be achieved by setting the tension according to $T = \rho A \cdot (2Lf_0)^2$. The first three drone strings are slightly detuned to simulate slight imperfections.

The melody string is tuned to A3, and its frets are spaced to provide two octaves of an A-major scale, while ensuring that frets do not overlap in terms of the grid points they affect.

4.4 Performance

The platform choice of Web Audio and JavaScript trades off optimal performance for convenience. However, we found the application to run without issues in chromium-based browsers on consumer-grade laptops.

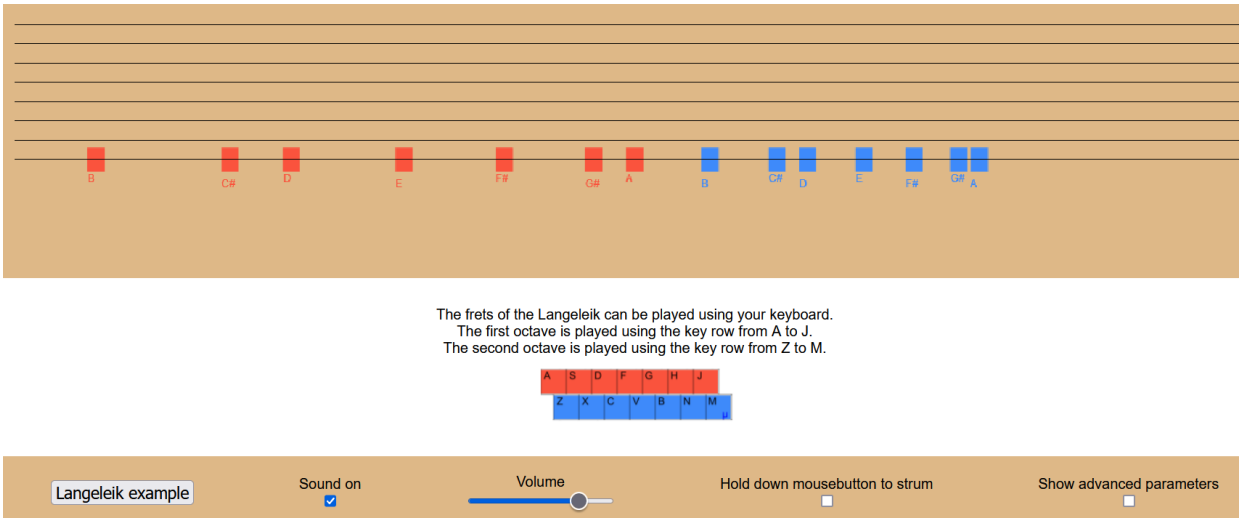


Figure 2: The graphical user interface of the web application.

On a laptop with 16 virtual cores and a clock speed of 2.9 GHz, the render capacity reported by Chrome's Web Audio analysis tool never exceeded 50% even with all frets pressed at the same time.

On another laptop with 4 virtual cores and a clock speed of 2.5 GHz, the render capacity stayed around the same mark while Chrome was the only application running, but reached close to 100% and caused some buffer underruns when other applications were competing for resources.

5. EVALUATION

A qualitative evaluation of the application was conducted remotely with five participants. The participants did not experience any performance issues and generally rated the audio quality favorably.

To evaluate the interface, the participants were asked to play invididual notes followed by a simple melody, for which they were provided the letter names of the notes.

All participants, when prompted to play a given note, initially moved their mouse to strum at the position of the corresponding fret indicator, as opposed to using the keyboard to press the fret and then strumming the string. We addressed this by featuring the instructions more prominently and permanently below the instrument.

Some participants also strummed between the fret and the "nut" end of the string, instead of between the fret and the "bridge" end. Since the excitation was applied at the location of the mouse, this causes the generated sound to be almost inaudible. We addressed this by fixing the excitation between the last fret and the bridge.

6. CONCLUSION AND FUTURE WORK

We presented an FDS model of the Langeleik and implemented it as an interactive, real-time web application. The sound quality of the application was rated favorably, but we would like to iterate further on the design of the interface and the interaction to create a more intuitive experience. Additionally, the application should be expanded to run smoothly on all major browsers and possibly mobile devices.

7. REFERENCES

[1] H. Panum, *Langelegen som Dansk Folkeinstrument*. Lehmann & Stage, 1918.

[2] The Crosby Brown Collection of Musical Instruments. (1889). [Online]. Available: https://www.metmuseum.org/art/collection/search/502451

[3] J. Smith, "Physical modeling using digital waveguides," *Computer Music Journal*, vol. 16, p. 74, 1992.

[4] S. Bilbao and A. Torin, "Numerical Modeling and Sound Synthesis for Articulated String/Fretboard Interactions," *Journal of the Audio Engineering Society*, vol. 63, no. 5, pp. 336–347, May 2015.

[5] S. Willemsen, S. Serafin, S. Bilbao, and M. Ducceschi, "Real-time Implementation of a Physical Model of the Tromba Marina," in *Proceedings of the 17th Sound and Music Computing Conference*, June 2020.

[6] M. G. Onofrei, S. Willemsen, and S. Serafin, "Real-Time Implementation of a Friction Drum Inspired Instrument using Finite Difference Schemes," in *Proceedings of the 24th International Conference on Digital Audio Effects (DAFx20in21)*, September 2021.

[7] S. Bilbao, *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics*. Wiley Publishing, 2009.

[8] G. Derveaux, A. Chaigne, P. Joly, and E. Bécache, "Time-domain simulation of a guitar: Model and method," *The Journal of the Acoustical Society of America*, vol. 114, no. 6, pp. 3368–3383, 2003.

[9] M. Ducceschi and S. Bilbao, "Non-Iterative Solvers for Nonlinear Problems: The Case of Collisions," in *Proceedings of the 22nd Conference of Digital Audio Effects (DAFx-19)*, Sep. 2019.

[10] developer.mozilla.org. (2021) Web audio api. https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API.

[11] J. Ritchie, *The Dulcimer Book*. Oak Publications, 1974. [Online]. Available: https://books.google.dk/books?id=KY7IDgAAQBAJ

A. DERIVATION OF FINGER COLLISIONS

This section demonstrates in detail how we arrive at the update equation for the melody string.

A.1 Discrete Force Model

This section is a summary of the approach described in [9], which presents a discretization of the collision force that will allow us to model the collision non-iteratively and keep the update equation explicit.

We define identity and temporal shift operators as

$$1\eta^n = \eta^n, \quad e_{t+}\eta^n = \eta^{n+1}, \quad e_{t-}\eta^n = \eta^{n-1}. \tag{19}$$

A first-order time derivative can then be approximated by the forward or the centered time difference operator, defined as

$$\delta_{t+} = \frac{e_{t+} - 1}{k} \quad \text{and} \quad \delta_{t\cdot} = \frac{e_{t+} - e_{t-}}{2k}, \tag{20}$$

respectively. The forwards and backwards averaging operator are defined as

$$\mu_{t+} = \frac{e_{t+} + 1}{2} \quad \text{and} \quad \mu_{t-} = \frac{1 + e_{t-}}{2}. \tag{21}$$

The collision potential from equation (7) can be rewritten as $\phi_i' = \psi_i \psi_i'$, where $\psi_i = \sqrt{2\phi_i}$. Using the chain rule, ψ_i' can then be expressed as

$$\psi_i' = \frac{d\psi_i}{d\eta_i} = \frac{d\psi_i}{dt}\frac{dt}{d\eta_i} = \frac{\left(\frac{d\psi_i}{dt}\right)}{\left(\frac{d\eta_i}{dt}\right)} = \frac{\dot{\psi}_i}{\dot{\eta}_i}. \tag{22}$$

We can then approximate ψ_i and ψ_i' in discrete time with the previously introduced operators as

$$\psi_i \approx \mu_{t+}\psi_i^{n-1/2} \quad \text{and} \quad \psi_i' \approx \frac{\delta_{t+}\psi_i^{n-1/2}}{\delta_{t\cdot}\eta_i^n}, \tag{23}$$

where $\psi_i^{n-1/2} = \mu_{t-}\psi_i^n$ denotes an interleaved grid point. Thus, we can discretize the continuous force $f_{c,i} = \phi_i' = \psi_i\psi_i'$ as

$$f_{c,i}(nk) \approx f_{c,i}^n = \mu_{t+}\psi_i^{n-1/2}\frac{\delta_{t+}\psi_i^{n-1/2}}{\delta_{t\cdot}\eta_i^n} \tag{24}$$

We define

$$g_i^n = \frac{\delta_{t+}\psi_i^{n-1/2}}{\delta_{t\cdot}\eta_i^n} \tag{25}$$

and can write equation (24) as

$$f_{c,i}^n = \mu_{t+}\psi_i^{n-1/2} g^n. \tag{26}$$

When we consider the identity

$$\mu_{t+}\psi_i^{n-1/2} = \frac{k}{2}\delta_{t+}\psi_i^{n-1/2} + \psi_i^{n-1/2}, \tag{27}$$

which follows from the definition of the grid operators, and rearrange equation (25) to

$$\delta_{t+}\psi_i^{n-1/2} = g_i^n \delta_{t\cdot}\eta_i^n, \tag{28}$$

we can insert this rearranged definition into the identity in equation (27) and thus write equation (26) as

$$f_{c,i}^n = \left(\frac{k}{2} g_i^n \delta_{t\cdot}\eta_i^n + \psi_i^{n-1/2}\right) g_i^n \tag{29}$$

As shown in [9], calculating g_i^n explicitly by evaluating the analytic expression for $\psi_i' = \frac{\phi_i'}{\sqrt{2\phi_i}}$ at $\eta_i = \eta_i^n$ leads to a numerically stable scheme. Written out, this results in

$$g_i^n = \sqrt{\frac{K(\alpha+1)}{2}}\,[\eta_i^n]_+^{\frac{\alpha-1}{2}}. \tag{30}$$

A.2 Discrete System for Finger Collision

Since equation (29) makes use of the time difference and thus depends on η_i^{n+1}, it might not be immediately clear that using this force model makes system (15) explicit.

When we consider a specific finger i, it is clear through the spreading operator used in equation (15a) that the collision force only affects the two grid points closest to the location of the finger. Since we assume that the frets are sufficiently distant from one another that any single grid point is never affected by two or more finger collisions, we can consider the interaction of finger i at location τ_i, affecting grid points l_{τ_i} and $l_{\tau_i}+1$, in isolation:

$$\begin{cases} h\ell_s u_{l_{\tau_i}}^n = (1-\alpha_e) f_e^n - (1-\alpha_{\tau_i}) f_{c,i}^n & \text{(31a)} \\ h\ell_s u_{l_{\tau_i}+1}^n = \alpha_e f_e^n - \alpha_{\tau_i} f_{c,i}^n & \text{(31b)} \\ \ell_f w_i^n = f_{c,i}^n - f_{p,i}^n & \text{(31c)} \\ \eta_i^n = (1-\alpha_{\tau_i}) u_{l_{\tau_i}}^n + \alpha_{\tau_i} u_{l_{\tau_i}+1}^n - w_i^n & \text{(31d)} \end{cases}$$

Here, α_e and α_{τ_i} result from the linear spreading operator as defined in equation (13), and its linear interpolation counterpart. Equations (31a-b) can be transformed into the system of two linear equations that can be solved for $u_{\tau_i}^{n+1}$ and $u_{\tau_i+1}^{n+1}$ by a single matrix division by

1. Expanding the operators ℓ_s and ℓ_f
2. Inserting the collision force definition from equation (29) into equations (31a-c)
3. Inserting the definition for η_i^n from equation (31d) into equations (31a-c) and expanding the centered difference operator
4. Obtaining a definition for w_i^{n+1} from equation (31c) and inserting it into equations (31a-b)