NONLINEAR STRINGS BASED ON MASSES AND SPRINGS

Silvin Willemsen

SW Audio
The Netherlands
silvinwillemsen@gmail.com

ABSTRACT

Due to advances in computational power, physical modelling for sound synthesis has gained an increased popularity over the past decades. Although much work has been done to accurately simulate existing physical systems, much less work exists on the use of physical modelling simply for the sake of creating sonically interesting sounds. This work presents a mass-spring network, inspired by existing models of the physical string. Masses have 2 translational degrees of freedom (DoF), and the springs have an additional equilibrium separation term, which together result in highly nonlinear effects. The main aim of this work is to create sonically interesting sounds while retaining some of the natural qualities of the physical string, as opposed to accurately simulating it. Although the implementation exhibits chaotic behaviour for certain choices of parameters, the presented system can create sonically interesting timbres, including nonlinear pitch glides and 'wobbles'.

1. INTRODUCTION

Mass-spring networks for sound synthesis have been investigated for over 40 years. Originally introduced in a musical context by Cadoz *et al.* [1, 2, 3], mass-interaction models have seen recent developments by Leonard and Villeneuve in [4, 5]. The modularity of mass-spring networks and their simple formulation make it an attractive physical modelling technique for creating interesting sounds relatively quickly.

As one is restricted to a finite number of nodes in space (i.e., the masses), other physical modelling techniques have been often used to model the musical string. Over the past 50 years, the string has been modelled using various methods, including physically-inspired methods such as the Karplus-Strong algorithm [6] and digital waveguides [7], as well as modal synthesis [8], and finite-difference time-domain (FDTD) methods [9, 10]. For an ideal string, the latter methods have an equivalent mass-spring formulation as described in [11, Sec 6.1.1], but once one wants to add stiffness, FDTD methods are a much more straightforward alternative.

The above models assume low-amplitude string vibration such that the string can be approximated using a linear model. High-amplitude string vibration results in an initial higher pitch of the string after which it decreases due to an effect called tension modulation [12]. To include this effect in the string model, it needs to be extended to be nonlinear instead. One of the most recent works on nonlinear string modelling is due to Ducceschi & Bilbao in [13], who use a mixed FDTD/modal scheme and energy quadratisation techniques, to model the geometrically exact string.

One could imagine that a proper implementation of the nonlinear string requires a very involved mathematical formulation, and quickly loses simplicity in implementation. As opposed to FDTD methods, a mass-spring formulation treats the discrete nodes of the implementation as separate connected entities, rather than as part of a predefined system, which provides additional flexibility in several aspects. One of these is the relatively easy extension to additional degrees of freedom (DoF) per node. In most traditional FDTD schemes, each node only has one degree of freedom, which in the case of the string is the transverse displacement [11]. If we allow for more degrees of freedom, this could potentially lead to interesting nonlinear effects. Furthermore, every mass and spring can be treated as a separate entity of which parameters can be set independently of each other also potentially leading to interesting sonic qualities.

The aim of this work is not to accurately simulate the nonlinear stiff string, but instead to create a flexible model that can produce string-like sounds with interesting nonlinear sonic qualities. To this end, this work uses a mass-spring formulation due to its flexibility and relatively simple formulation. The main additions with respect to an FDTD implementation of the (damped) ideal string are using 2 DoF for each mass and an equilibrium separation for the springs connecting neighbouring masses. These two additions together, result in pitch glides and interesting timbres not feasible with linear models.

The rest of this paper is structured as follows: Section 2 describes the continuous-time model starting with the description of the 2-DoF mass and extending this to a sequentially connected network resembling a string. Section 3 discretises the model and uses analogies to the FDTD formulation to determine the stability and fundamental frequency of the system. Section 4 provides results and discusses several experiments done using the presented schemes, and Section 5 concludes.

2. MODEL

2.1. Single mass

Before considering a network of masses and springs, first consider a single mass $\mathbf{u} = \mathbf{u}(t)$ with time t (in s), defined over two spatial DoF,

$$\mathbf{u} = \begin{bmatrix} u_x & u_y \end{bmatrix}. \tag{1}$$

Here, $u_x = u_x(t)$ and $u_y(t)$ describe the x and y-location of the mass, respectively (in m). Using the ∂_t operator to denote differentiation with respect to time, the ODE describing the dynamics of a 2-DoF mass-spring system connected to the origin is (such as

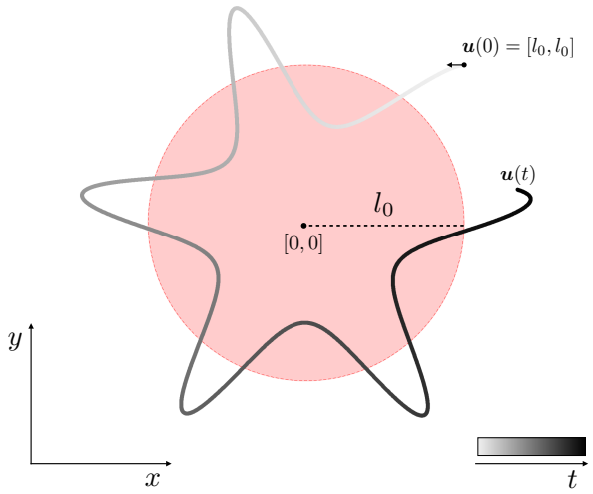


Figure 1: *The effect of l_0 on the trajectory of a 2-DoF mass-spring system. The mass will be pulled towards the origin ($[0, 0]$) if it is outside of the red region. If the mass is in the red region instead, it will be pushed away from the origin. Here, the mass is initialised at $[l_0, l_0]$ and given a small horizontal velocity in the negative x-direction.*

done in [4])

$$M\partial_t^2 \mathbf{u} = -Kf\frac{\mathbf{u}}{\|\mathbf{u}\|}, \quad \text{with} \quad f = \|\mathbf{u}\| - l_0, \tag{2}$$

mass $M > 0$ (in kg), spring constant $K > 0$ (in N/m), equilibrium spring separation $l_0 \geq 0$ (in m) and spring force f (in N). Furthermore, $\|\mathbf{u}\|$ describes the magnitude of $\mathbf{u}$ (in m), which when using two DoF is defined as

$$\|\mathbf{u}\| = \sqrt{u_x^2 + u_y^2}\,. \tag{3}$$

The equilibrium separation l_0, can be seen as introducing a zone where the spring pushes the mass away from the equilibrium rather than pulling it towards it. See Figure 1.

One can observe that if $l_0 = 0$, Eq. (2) reduces to a mass-spring system whose dimensions are uncoupled; in other words, l_0 introduces coupling between the two dimensions.

2.2. Multiple masses

Going towards a string-like mass-spring network, one can create a system of N_{mass} masses connected by N_{spring} springs, which are related as follows:

$$N_{\text{mass}} = N_{\text{spring}} + 1. \tag{4}$$

Subscript $m = \{0, \ldots N_{\text{spring}}\}$ will be used to index the masses, i.e., mass m will be described by the state $\mathbf{u}_m = \mathbf{u}_m(t)$. For a system of length L (in m), the masses are initially placed along the x-axis with no displacement in the y-direction according to

$$\mathbf{u}_m(0) = \begin{bmatrix} m\Delta_0 & 0 \end{bmatrix}, \tag{5}$$

where $\Delta_0 = L/N_{\text{spring}}$ is the initial distance between two consecutive masses. Connecting the masses such that the spring forces between masses m and $m+1$ positively affects mass m and the force between masses m and $m-1$ negatively affects mass m yields

$$\begin{aligned} M\partial_t^2 \mathbf{u}_m = {} & Kf_{m+1/2}\frac{\mathbf{u}_{m+1} - \mathbf{u}_m}{\|\mathbf{u}_{m+1} - \mathbf{u}_m\|} \\ & - Kf_{m-1/2}\frac{\mathbf{u}_m - \mathbf{u}_{m-1}}{\|\mathbf{u}_m - \mathbf{u}_{m-1}\|}, \end{aligned} \tag{6}$$

where the spring force between masses m and $m+1$ is

$$f_{m+1/2} = \|\mathbf{u}_{m+1} - \mathbf{u}_m\| - l_0. \tag{7}$$

Notice that M applies to all masses and K and l_0 apply to all springs in the network.

2.3. Boundaries

Although in a mass-spring context one does not usually speak of boundary conditions, conditions for the first and the last mass still need to be defined. These are set to the following states:

$$\mathbf{u}_0(t) = \begin{bmatrix} 0 & 0 \end{bmatrix}, \quad \text{and} \quad \mathbf{u}_{N_{\text{spring}}}(t) = \begin{bmatrix} L & 0 \end{bmatrix}, \quad \forall t. \tag{8}$$

2.4. Damping

Similar to the damped stiff string (see e.g. [11]), one can add damping terms to Eq. (6) according to

$$\begin{aligned} M\partial_t^2 \mathbf{u}_m = {} & \ldots - 2\sigma M\partial_t \mathbf{u}_m \\ & + 2zM\partial_t\left(\mathbf{u}_{m+1} - \mathbf{u}_m\right) - 2zM\partial_t\left(\mathbf{u}_m - \mathbf{u}_{m-1}\right), \end{aligned} \tag{9}$$

with mass damping $\sigma \geq 0$ and spring damping $z \geq 0$ (both in s^{-1}). The spring damping term acts as a frequency-dependent damping term, analogous to the damped stiff string. As the implementation of this damping term is also analogous to its implementation in the stiff string it can be shown to be passive under similar conditions (see Sec. 3.1.2).

Using the following shorthand notation

$$\boldsymbol{\mathcal{U}}_{m+1/2} \triangleq \frac{\mathbf{u}_{m+1} - \mathbf{u}_m}{\|\mathbf{u}_{m+1} - \mathbf{u}_m\|}, \tag{10}$$

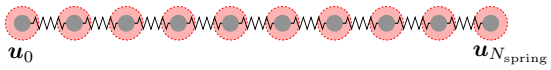
one can simplify Eq. (9) to

$$\begin{aligned} \partial_t^2 \mathbf{u}_m = {} & \frac{K}{M}\left(\mathbf{u}_{m+1} - 2\mathbf{u}_m + \mathbf{u}_{m-1}\right) \\ & - \frac{Kl_0}{M}(\boldsymbol{\mathcal{U}}_{m+1/2} - \boldsymbol{\mathcal{U}}_{m-1/2}) \\ & - 2\sigma\partial_t \mathbf{u}_m + 2z\partial_t\left(\mathbf{u}_{m+1} - 2\mathbf{u}_m + \mathbf{u}_{m-1}\right). \end{aligned} \tag{11}$$

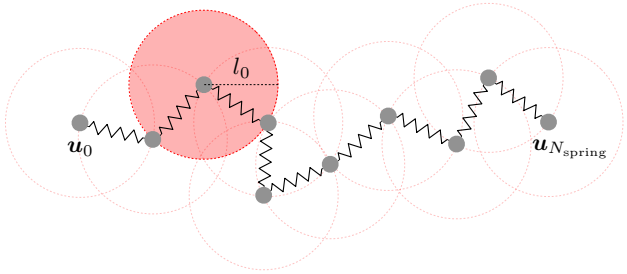
Figure 2 illustrates possible stable states (initially excited and after damping) of the 2-DoF mass-spring network described in this section for low and high values of l_0.

3. DISCRETE TIME

One can discretise continuous time according to $t = nk$, with time index $n = 0, 1, \ldots$, time step $k = 1/f_s$ (in s) and sample rate f_s (in Hz). The position of mass m can then be discretised at



(a) *Small value of* l_0.



(b) *Large value of* l_0.

Figure 2: *Possible stable states for different values of* l_0.

time t according to $\mathbf{u}_m(t) \approx \mathbf{u}_m^n$. To approximate the continuous-time derivatives used in the previous section, the following finite-difference (FD) operators need to be introduced:

$$\partial_t \mathbf{u}_m \approxeq \begin{cases} \delta_{t+}\mathbf{u}_m^n \triangleq \dfrac{1}{k}\left(\mathbf{u}_m^{n+1} - \mathbf{u}_m^n\right), & \text{(12a)} \\ \delta_{t-}\mathbf{u}_m^n \triangleq \dfrac{1}{k}\left(\mathbf{u}_m^{n} - \mathbf{u}_m^{n-1}\right), & \text{(12b)} \\ \delta_{t\cdot}\mathbf{u}_m^n \triangleq \dfrac{1}{2k}\left(\mathbf{u}_m^{n+1} - \mathbf{u}_m^{n-1}\right), & \text{(12c)} \end{cases}$$

which are the forward, backward, and centred difference respectively. The former two are first-order accurate, whereas the latter is second-order accurate [11]. A second-order differentiation can be approximated using

$$\partial_t^2 \mathbf{u}_m \approxeq \delta_{tt}\mathbf{u}_m^n \triangleq \frac{1}{k^2}\left(\mathbf{u}_m^{n+1} - 2\mathbf{u}_m^n + \mathbf{u}_m^{n-1}\right), \tag{13}$$

which is also second-order accurate.

Using these definitions, Eq. (11) can then be discretised to the following scheme:

$$\begin{aligned} \delta_{tt}\mathbf{u}_m^n = {} & \frac{K}{M}\left(\mathbf{u}_{m+1}^n - 2\mathbf{u}_m^n + \mathbf{u}_{m-1}^n\right) \\ & - \frac{Kl_0}{M}\left(\boldsymbol{\mathcal{U}}_{m+1/2}^n - \boldsymbol{\mathcal{U}}_{m-1/2}^n\right) \\ & - 2\sigma\delta_{t\cdot}\mathbf{u}_m^n + 2z\delta_{t-}\left(\mathbf{u}_{m+1}^n - 2\mathbf{u}_m^n + \mathbf{u}_{m-1}^n\right), \end{aligned} \tag{14}$$

where

$$\boldsymbol{\mathcal{U}}_{m+1/2}^n \triangleq \frac{\mathbf{u}_{m+1}^n - \mathbf{u}_m^n}{\|\mathbf{u}_{m+1}^n - \mathbf{u}_m^n\|}, \tag{15}$$

is Eq. (10) discretised. Notice that the first-order FD operators are chosen to yield the highest accuracy, while keeping the scheme explicit. Appendix 7 provides an alternative, implicit discretisation.

To implement scheme (14), it needs to be expanded to an update equation, or recursion:

$$\begin{aligned} (1+\sigma k)\mathbf{u}_m^{n+1} = {} & \left(2 - \frac{2Kk^2}{M} - 4zk\right)\mathbf{u}_m^n \\ & + \left(\frac{Kk^2}{M} + 2zk\right)\left(\mathbf{u}_{m+1}^n + \mathbf{u}_{m-1}^n\right) \\ & - \frac{Kl_0k^2}{M}\left(\boldsymbol{\mathcal{U}}_{m+1/2}^n - \boldsymbol{\mathcal{U}}_{m-1/2}^n\right) \\ & + (\sigma k + 4zk - 1)\mathbf{u}_m^{n-1} - 2zk\left(\mathbf{u}_{m+1}^{n-1} + \mathbf{u}_{m-1}^{n-1}\right), \end{aligned} \tag{16}$$

which, after division by $(1+\sigma k)$, can be solved for $\mathbf{u}_m^{n+1}$.

3.1. Analogies to FDTD schemes

If one is familiar with FDTD methods, it is easy to see the resemblance between scheme (14) and FDTD schemes of the (damped) 1D wave equation. Despite some differences (being the term including l_0 and the possibility for additional DoF), this resemblance can still be used to find definitions for the fundamental frequency and stability, as will be presented in this section. For completeness, the FDTD scheme of the 1D wave equation is given here.

The transverse displacement of an ideal string of length L (in m) can be described by state variable $u = u(x,t)$ (in m), which is defined over space $x \in [0, L]$ (in m) and time t (in s). Space x is subdivided into N_{FD} equally sized intervals according to $x = lh$ with spatial index $l = \{0, \ldots, N_{\text{FD}}\}$, and grid spacing h (in m). Time t is discretised according to the same definitions presented at the beginning of this section. Using these definitions, $u(x,t)$ can be approximated by grid function u_l^n.

Introducing the following FD operator, which approximates a second-order spatial derivative

$$\partial_x^2 u \approx \delta_{xx}u_l^n \triangleq \frac{1}{h^2}\left(u_{l+1}^n - 2u_l^n + u_{l-1}^n\right), \tag{17}$$

the discrete damped 1D wave equation can be described by the following scheme [11]:

$$\delta_{tt}u_l^n = c^2\delta_{xx}u_l^n - 2\sigma_0\delta_{t\cdot}u_l^n + 2\sigma_1\delta_{t-}\delta_{xx}u_l^n. \tag{18}$$

Here, c is the wave speed (in m/s), and σ_0 and σ_1 are the frequency-independent (in s^{-1}) and the frequency-dependent damping coefficients (in m^2/s), respectively.

3.1.1. Fundamental frequency

To obtain the fundamental frequency (in Hz) of a string of length L (in m) fixed at the ends, one uses

$$f_0 = \frac{c}{2L}. \tag{19}$$

Ignoring the damping terms for now (as these do not influence f_0), one can compare the update equation of the 1D wave equation with that of the mass spring network in Eq. (16):

$$\begin{aligned} u_l^{n+1} &= \left(2 - \frac{2c^2k^2}{h^2}\right)u_l^n - u_l^{n-1} + \frac{c^2k^2}{h^2}\left(u_{l+1}^n + u_{l-1}^n\right), \\ \mathbf{u}_m^{n+1} &= \left(2 - \frac{2Kk^2}{M}\right)\mathbf{u}_m - \mathbf{u}_l^{n-1} + \frac{Kk^2}{M}\left(\mathbf{u}_{m+1}^n + \mathbf{u}_{m-1}^n\right). \end{aligned}$$

One can observe that the following combination of variables is analogous to each other:

$$\frac{c^2}{h^2} \iff \frac{K}{M}. \tag{21}$$

As the mass-spring network does not make use of h we can use $h = L/N_{\text{FD}}$ (in the case of the Courant number $\lambda = ck/h = 1$ for the 1D wave equation), and substitute this to yield

$$\frac{c^2 N_{\text{FD}}^2}{L^2} = \frac{K}{M}. \tag{22}$$

As N_{FD} describes the number of intervals (rather than the number of grid points) in the FDTD scheme, this is analogous to the number of springs in the mass-spring network N_{spring}. Using its relation to the number of masses N_{mass} in Eq. (4), rearranging Eq. (22) in terms of c, and substituting this into Eq. (19) yields

$$f_0 = \frac{\sqrt{K/M}}{N_{\text{mass}} - 1}. \tag{23}$$

It is interesting to note that the fundamental frequency is solely determined by the spring constant, the mass, and the number of masses in the system. Therefore, the length L no longer has an influence on the fundamental frequency of the system.

3.1.2. Stability

The stability condition for Eq. (18) can be shown to be [14]

$$\frac{c^2 k^2}{h^2} + \frac{4\sigma_1 k}{h^2} \leq 1. \tag{24}$$

Comparing Eqs. (14) and (18) again, an analogy between the following variables can be made

$$\sigma_1/h^2 \iff z,$$

which, after including Eq. (21), one can rewrite (24) the following stability condition for the mass-spring network

$$\frac{Kk^2}{M} + 4zk \leq 1.$$

As is the case for the 1D wave equation, the closer this condition is to being satisfied with equality, the higher the simulation bandwidth. As we would like to have control over this condition later on, this is rewritten to

$$\frac{Kk^2}{M} + 4zk \leq \Lambda, \tag{25}$$

where $0 < \Lambda \leq 1$ determines the bandwidth limit of the simulation.

3.2. Implementation

Assuming that the fundamental frequency is known, Eq. (23) can be rewritten to

$$N_{\text{mass}} = \frac{\sqrt{K/M}}{2f_0} + 1. \tag{26}$$

Rewriting Eq. (25) in terms of K/M,

$$\frac{K}{M} \leq \frac{(\Lambda - 4zk)}{k^2},$$

and substituting this into Eq. (26), yields

$$N_{\text{mass}} \geq \frac{\sqrt{(\Lambda - 4zk)}}{2f_0 k} + 1. \tag{27}$$

As N_{mass} is an integer, a rounding operation needs to be performed on Eq. (27) that also satisfies the condition. The following can therefore be used to calculate the number of masses

$$N_{\text{mass}} = \left\lfloor \frac{\sqrt{(\Lambda - 4zk)}}{2f_0 k} \right\rfloor + 1, \tag{28}$$

where $\lfloor \cdot \rfloor$ is the flooring operation. Finally, either M or K can be fixed to an arbitrary value, and the other can be calculated by rewriting Eq. (26)

$$K = M(2f_0(N_{\text{mass}} - 1))^2. \tag{29}$$

Here, M is kept fixed and K is changed, analogous to changing the tension of a string and keeping the mass per unit length fixed.

3.3. Output

One can obtain the output of the system by selecting one mass and following its state over time. For an interesting stereo effect, the longitudinal (x) and transverse (y) dimensions, can be mapped to the left and right channel, respectively. As $u^n_{x,m}$ has an initial non-zero location due to Eq. (5) this needs to be corrected for, resulting in

$$o_{\text{left}}(n) = u^n_{x,m_{\text{o}}} - m_{\text{o}}\Delta_0 \quad \text{and} \quad o_{\text{right}}(n) = u^n_{y,m_{\text{o}}}, \tag{30}$$

where $m_{\text{o}} \in \{0, N_{\text{spring}}\}$ is the index of the mass selected for the output.

3.4. Extension to anisotropic systems

Another advantage of using a mass-spring formulation as opposed to a FDTD one, is that it is relatively straightforward to use different parameter values for different parts of the network. Although this subsection will not be further discussed in the next sections, it is interesting to mention a simple extension of Eq. (14) to be anisotropic.

One can rewrite Eq. (9) to allow for different values for M and K along the network.[1] Using M_m to denote the mass of mass m, and $K_{m+1/2}$ to describe the spring force between masses m and $m+1$,

$$\begin{aligned} M_m \partial_t^2 \mathbf{u}_m = {} & K_{m+1/2} f_{m+1/2} \boldsymbol{\mathcal{U}}_{m+1/2} \\ & - K_{m-1/2} f_{m-1/2} \boldsymbol{\mathcal{U}}_{m-1/2} \\ & - 2\sigma M_m \partial_t \mathbf{u}_m \\ & + 2z M_{m+1/2} \partial_t \left(\mathbf{u}_{m+1} - \mathbf{u}_m\right) \\ & - 2z M_{m-1/2} \partial_t \left(\mathbf{u}_m - \mathbf{u}_{m-1}\right). \end{aligned} \tag{31}$$

Here,

$$M_{m+1/2} = \frac{1}{2}\left(M_{m+1} + M_m\right) \tag{32}$$

is the average mass of two neighbouring masses. Please note that the fundamental frequency calculation in Eq. (23) does not hold for an anisotropic system.

[1] In principle, σ, z, and l_0 could also have been chosen to vary along the network, but are left constant for the sake of brevity.

3.4.1. Stability

In order to keep the system stable, the stability condition in Eq. (25) needs to be adapted to challenge the condition most:

$$\frac{K_{\max}k^2}{M_{\min}} + 4zk \leq \Lambda, \tag{33}$$

where

$$K_{\max} = \max_{m\in\{0,\ldots,N_{\text{spring}}\}} K_{m+1/2} \quad \text{and} \quad M_{\min} = \min_{m\in\{0,\ldots,N_{\text{mass}}\}} M_m. \tag{34}$$

4. RESULTS AND DISCUSSION

This section presents the results of several simulations using Eq. (16) with different parameter values and discusses these. Sound examples can be found online [15].

4.1. Simulation setup

As according to Eq. (23) the length of the system L does not change the eventual behaviour of the system, we can set $L = N_{\text{spring}}$ such that $\Delta_0 = 1$ and Eq. (5) simplifies to

$$\mathbf{u}_m(0) = \begin{bmatrix} m & 0 \end{bmatrix}.$$

This way, the value of l_0 can be seen as a ratio of the initial difference between two consecutive masses (i.e., $l_0 = 0.5$ yields a resting length of half the initial distance between two masses). The system is then excited by giving mass $m_{\text{e}} \in \{0, \ldots, N_{\text{spring}}\}$ an initial displacement of e in both x and y directions:

$$\mathbf{u}^0_{m_{\text{e}}} = \mathbf{u}^1_{m_{\text{e}}} = \begin{bmatrix} m_{\text{e}} + e & e \end{bmatrix}. \tag{35}$$

Notice that with this setup, the output in Eq. (30) has to be normalised (divided by e to yield output in the $[-1, 1]$ range. Table 1 shows the other parameters used for the experiments and provides usable ranges for some.

4.2. Chaotic behaviour

The first thing to note, is that for values of Λ close to 1 in Eq. (25), non-zero values of l_0 cause increasingly chaotic behaviour, which causes the system to produce 'buzzing' output. This is in line with what Bilbao mentions in [11, p. 229], stating that

> "*... for a nonlinear system, anomalous behavior may be observed when the grid spacing is chosen close to the stability bound.*"

The behaviour is most likely caused by a numerical integration error of the nonlinear term, which is (most probably) why the implicit implementation in Appendix 7 shows slightly improved behaviour in this regard. It is important to note that the chaotic behaviour does not imply that the implementation is unstable; although the system might never fully decay, it does not exhibit explosive behaviour!

The system has been tested for different values of l_0, z and Λ to see whether it would either exhibit chaotic behaviour or decay instead. These tests have been repeated at $f_{\text{s}} = 44100$ and $f_{\text{s}} = 88200$. A full overview of the results can be found in Figure 3. All generated sounds can be found via [15] and were obtained through Eq. (30).

Table 1: *Parameter values divided into static parameters used to generate the results and parameters that can be tweaked to generate different behaviour.*

Parameter	Symbol (unit)	Value
Static parameters		
Mass	M (kg)	0.01
Fundamental freq.	f_0 (Hz)	100
Spring stiffness	K (N/m)	Eq. (29)
Mass damping	σ (s^{-1})	1
Initial displacement	e (m)	100
Initial inter-mass dist.	Δ_0 (m)	1
Excited mass	m_{e} (-)	$m_{\text{e}} = \lfloor 0.63 N_{\text{spring}} \rfloor$
Output mass	m_{o} (-)	10
Number of masses	N_{mass} (-)	Eq. (28)
Altered parameters		
Stability bound	Λ (-)	$0 < \Lambda \leq 1$
Spring damping	z (s^{-1})	$z \in [0, 5^*]$
Equilibrium sep.	l_0 (m)	$l_0 \in [0, 2^*]$
Sample rate	f_{s} (Hz)	$f_{\text{s}} \in [44100, 88200]^*$

*these numbers are to determine usable ranges, but the parameters are not bounded by these values.

The results indicate that lower values for Λ and l_0 are the main factors for preventing chaotic behaviour. Similar results are obtained for both sample rates (even slightly in favour of $f_{\text{s}} = 44100$). This is because the simulation is not actually oversampled; more masses are added according to Eq. (28) due to a decrease in k. If one instead oversamples without changing any other parameters, this automatically decreases Λ in Eq. (25) resulting in reduced chaotic behaviour. Although this retains a high simulation bandwidth, it does increase the computational cost.

If the eventual goal is to implement this algorithm in real time, a better option would be to reduce Λ manually. Although this decreases the simulation bandwidth, it decreases chaotic behaviour without increasing the computational complexity (even reducing it by reducing the number of masses in springs through Eq. (28)!). Results show that if one chooses $z \geq 3$, values for $\lambda \leq 0.1$ result in non-chaotic behaviour for $l_0 \in [0, 2]$ (and probably higher).

The fact that an increase in the spring damping z decreases chaotic behaviour follows from the fact that the chaotic oscillations cause rapid extensions and contractions of the springs. This will be more damped for higher values of z. Furthermore, a higher value for z also increases the speed that the simulation reaches a stable equilibrium.

4.3. Frequency-domain behaviour

Spectrograms of the simulation with parameters $l_0 = 0.75$, $z = 2$, $\Lambda = 0.1$ can be seen in Figure 4. The results and discussion below assume that Λ and z are chosen such that the system does not behave chaotically.

All non-zero values of l_0 have a tension-reduction effect on the string in the transverse (y) direction. In the longitudinal (x) direction, however, the effect of l_0 on the frequency depends on its value with respect to the initial distance between the masses Δ_0. For $l_0 \leq \Delta_0$, the equilibrium separation in the springs eventually (after damping) cancel each other out, and the system will – after a slight 'wobble' downwards in pitch – return to its original funda-

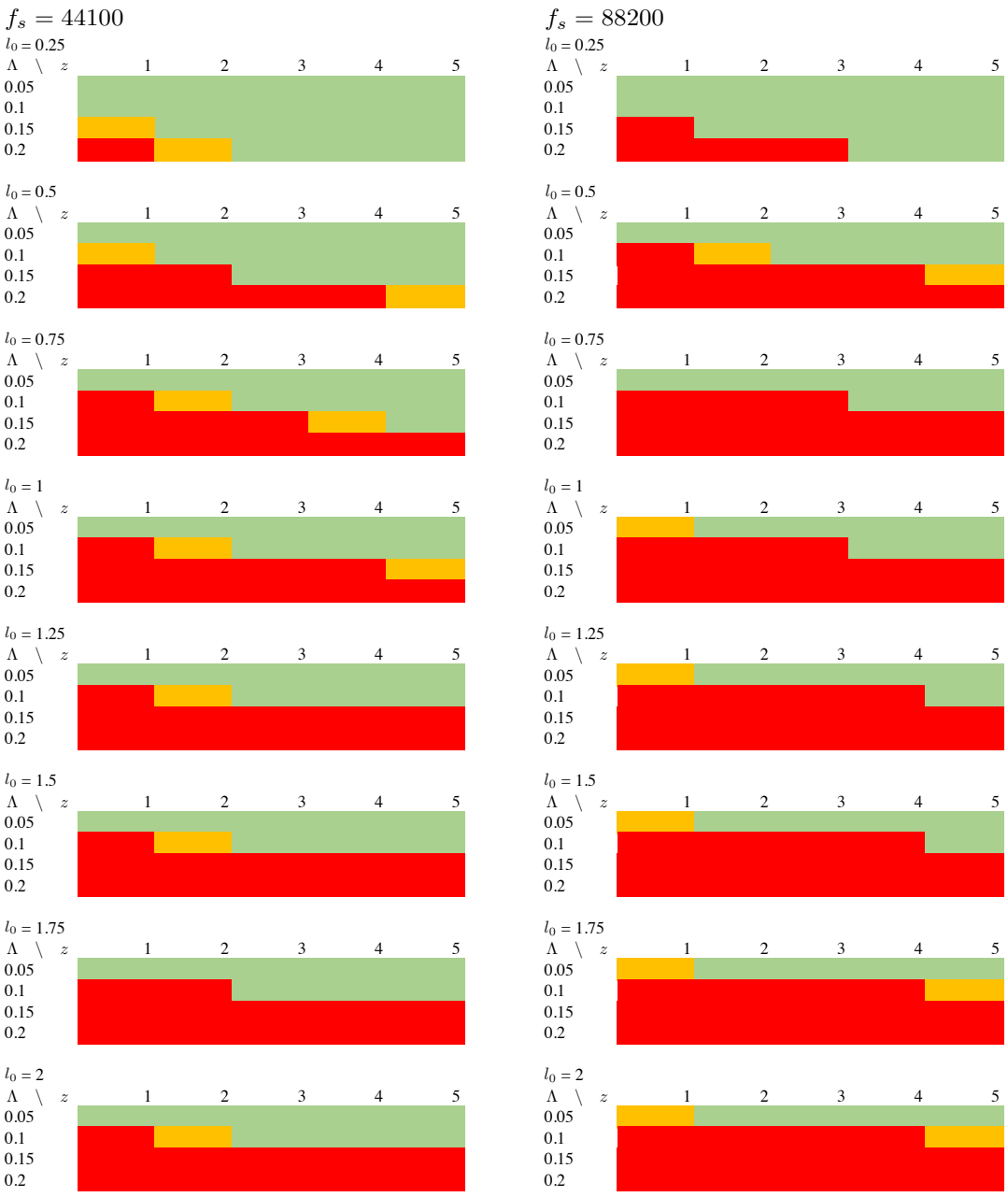


Figure 3: *Simulation results for* $\Lambda = \{0.05, 0.1, 0.15, 0.2\}$, $z = \{1, \ldots 5\}$ *and* $l_0 = \{0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2\}$. *Simulations lasted for 5 seconds and* ***green cells*** *indicate that the output decays within this time.* ***Red cells*** *indicate that the output does not decay within this time, and the system is therefore considered to exhibit chaotic behaviour.* ***Orange cells*** *are boundary cases, where the decay is slightly longer than usual, but the output decays within 5 s.*

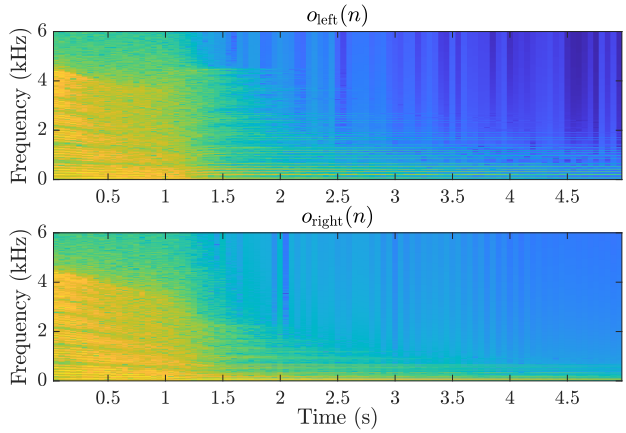


Figure 4: *Spectrograms of the left (longitudinal) and right (transverse) output in Eq.* (30) *with parameters from Table 1 and* $l_0 = 0.75$*,* $z = 2$*,* $\Lambda = 0.1$*, and* $f_s = 44100$*.*

mental frequency. For l_0 values larger than the initial equilibrium separation ($l_0 > \Delta_0$), the string will reach a stable 'loose' state such as depicted in Figure 2b. These results are summarised in Table 2.

Preliminary tests show that $l_0 \approx 0.75\Delta_0$ causes a pitch glide to $f_0/2$ (the subharmonic) and $l_0 \approx 0.94\Delta_0$ to $f_0/4$. More work needs to be done to find the exact relationship between l_0 and f_0. As opposed to the nonlinear schemes presented in e.g. [11, Ch. 8], the pitch glides do not start higher than the original fundamental frequency, but instead move towards 0 Hz, due to the tension-reduction effect of l_0.

Table 2: *Effect of equilibrium separation* l_0 *on output (Eq.* (30)*) if behaviour is not chaotic.*

Value for l_0	Effect on $o_{\text{left}}(n)$	Effect on $o_{\text{right}}(n)$
$l_0 \leq \Delta_0$	Glide down and back up.	Glide to lower f_0
$l_0 > \Delta_0$	Glide to $f_0 = 0$.	Glide to $f_0 = 0$

4.4. Note on only using the longitudinal direction

If one chooses to only excite the longitudinal direction, i.e.

$$\mathbf{u}^0_{m_\text{e}} = \mathbf{u}^1_{m_\text{e}} = \begin{bmatrix} m_\text{e} + e & 0 \end{bmatrix},$$

the following holds:

$$u^n_{y,m} = 0, \quad \forall m, \forall n.$$

In other words, the system behaves as if there was no transverse dimension. For $l_0 \leq \Delta_0$ pitch gliding effects still occur, but chaotic behaviour already occurs for much lower values of l_0. For $l_0 > \Delta_0$, due to the lack of the transversal dimension, the masses effectively have "nowhere to go", resulting in chaotic behaviour at all times.

4.5. Note on computational complexity

Compared to an implementation of the damped 1D wave equation, the update equation in Eq. (16) introduces additional computations in two different ways. First is the obvious extension to 2 DoF, doubling the number of computations with respect to a 1-DoF implementation. The second and more important contribution to computational complexity comes from the calculation of $\boldsymbol{\mathcal{U}}$; more specifically the calculation of the Euclidian distance between two neighbouring masses ($\|\cdot\|$), which adds N_{spring} square-root operations every time step.

However, due to the already computationally inexpensive implementation of the damped 1D wave equation, these additional aspects should definitely not prevent a real-time implementation of the model presented here.

5. CONCLUSION

This paper presents a mass-spring network configured like a string. The masses in the network can move in 2 DoF and the springs connecting the masses have an equilibium separation. These properties cause nonlinear behaviour in the system, such as wobbles and pitch glides.

Although parameters could be chosen that yield chaotic behaviour, results show that one can prevent this by choosing parameters away from the stability bound and including spring damping. Future work includes to find a more precise definition for parameter ranges for which the implementation does not exhibit chaotic behaviour, as well as a relationship for the fundamental frequency and equilibrium spring separation. Finally, it would be interesting to see how this model compares to other already existing models of nonlinear strings or modular mass-spring networks, such as the CORDIS-ANIMA software [3].

6. REFERENCES

[1] C. Cadoz, *Synthèse sonore par simulation de mécanismes vibratoires*, Ph.D. thesis, Thèse de Docteur Ingénieur, I.N.P.G. Grenoble, France, 1979.

[2] C. Cadoz, A. Luciani, and J.-L. Florens, "Responsive input devices and sound synthesis by simulation of instrumental mechanisms: the CORDIS system," *Computer Music Journal*, vol. 8, no. 3, pp. 60–73, 1983.

[3] C. Cadoz, A. Luciani, and J.-L. Florens, "CORDIS-ANIMA: a modeling and simulation system for sound and image synthesis: the general formalism," *Computer Music Journal*, vol. 17, no. 1, pp. 19–29, 1993.

[4] J. Villeneuve and J. Leonard, "Mass-interaction physical models for sound and multi-sensory creation: Starting anew," in *Proceedings of the 16th Sound and Music Computing Conference*, 2019.

[5] J. Leonard and J. Villeneuve, "MI-GEN~: An efficient and accessible mass interaction sound synthesis toolbox," *Proceedings of the 16th Sound and Music Computing Conference (SMC)*, 2019.

[6] K. Karplus and A. Strong, "Digital synthesis of plucked-string and drum timbres," *Computer Music Journal*, vol. 7, pp. 43–55, 1983.

[7] J. O. Smith, "Physical modeling using digital waveguides," *Computer Music Journal*, vol. 16, no. 4, pp. 74–91, 1992.

[8] J. D. Morrison and J.-M. Adrien, "Mosaic: A framework for modal synthesis," *Computer Music Journal*, vol. 17, no. 1, pp. 45–56, 1993.

[9] P. Ruiz, "A technique for simulating the vibrations of strings with a digital computer," M.S. thesis, University of Illinois, 1969.

[10] L. Hiller and P. Ruiz, "Synthesizing musical sounds by solving the wave equation for vibrating objects: Part I," *Journal of the Audio Engineering Society (JASA)*, vol. 19, no. 6, pp. 462–470, 1971.

[11] S. Bilbao, *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics*, John Wiley & Sons, 2009.

[12] T. Tolonen, V. Välimäki, and M. Karjalainen, "Modeling of tension modulation nonlinearity in plucked strings," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 3, pp. 300–310, 2000.

[13] M. Ducceschi and S. Bilbao, "Simulation of the geometrically exact nonlinear string via energy quadratisation," *Journal of Sound and Vibration*, vol. 534, 2022.

[14] S. Willemsen, *The Emulated Ensemble: Real-Time Simulation of Musical Instruments using Finite-Difference Time-Domain Methods*, Ph.D. thesis, Aalborg University Copenhagen, Jul. 2021.

[15] S. Willemsen, "Nonlinear Mass Spring DAFx23," Available at https://github.com/SilvinWillemsen/NonlinearMassSpring_DAFx23, accessed April 7, 2023.

7. APPENDIX A: AN ALTERNATIVE DISCRETISATION

Using the centred averaging operator

$$\mu_{t\cdot} u^n = \frac{1}{2}\left(u^{n+1} + u^{n-1}\right), \tag{36}$$

an alternative discretisation of the nonlinear term in Eq. (11) (as opposed to Eq. (14)) can be written according to[2]

$$\begin{aligned}\delta_{tt}\mathbf{u}_m^n = &\frac{K}{M}\left(\mathbf{u}_{m+1}^n - 2\mathbf{u}_m^n + \mathbf{u}_{m-1}^n\right)\\ &- \frac{Kl_0}{M}\left(\mu_{t\cdot}\boldsymbol{\mathcal{U}}_{m+1/2}^n - \mu_{t\cdot}\boldsymbol{\mathcal{U}}_{m-1/2}^n\right)\\ &- 2\sigma\delta_{t\cdot}\mathbf{u}_m^n + 2z\delta_{t\cdot}\left(\mathbf{u}_{m+1}^n - 2\mathbf{u}_m^n + \mathbf{u}_{m-1}^n\right).\end{aligned} \tag{37}$$

Although this makes the system fully implicit, preliminary results show that this can reduce chaotic behaviour in some situations. However, due to its implicit nature, this scheme takes much longer to compute than the scheme in Eq (14).

[2] Notice that the last term in Eq. (37) now also uses a $\delta_{t\cdot}$ operator.