LARGE-SCALE REAL-TIME MODULAR PHYSICAL MODELING SOUND SYNTHESIS

Stefan Bilbao
Acoustics and Audio Group
University of Edinburgh, Edinburgh, UK
sbilbao@ed.ac.uk

Michele Ducceschi *
Acoustics and Audio Group
University of Edinburgh, Edinburgh, UK
michele.ducceschi@ed.ac.uk

Craig J. Webb
Physical Audio Ltd.,
London, UK
craig@physicalaudio.co.uk

ABSTRACT

Due to recent increases in computational power, physical modeling synthesis is now possible in real time even for relatively complex models. We present here a modular physical modeling instrument design, intended as a construction framework for string- and bar-based instruments, alongside a mechanical network allowing for arbitrary nonlinear interconnection. When multiple nonlinearities are present in a feedback setting, there are two major concerns. One is ensuring numerical stability, which can be approached using an energy-based framework. The other is coping with the computational cost associated with nonlinear solvers—standard iterative methods, such as Newton-Raphson, quickly become a computational bottleneck. Here, such iterative methods are sidestepped using an alternative energy conserving method, allowing for great reduction in computational expense or, alternatively, to real-time performance for very large-scale nonlinear physical modeling synthesis. Simulation and benchmarking results are presented.

1. INTRODUCTION

One goal of physical modeling synthesis is the emulation of existing musical instruments, perhaps with extensions in terms of design and functionality not easily realisable in the real world. Another is the design of new instruments, without a real-world referent. The hope is that through adherence to the laws of physics, synthetic sound of a natural acoustic character can be produced. In this setting, modularity is an important concept—the idea is to give the user control over instrument design through the interconnection of semi-independent modules. In this contribution, a modular network, consisting of a set of string- or bar-like elements coupled through an auxiliary nonlinear connection network is presented.

Modularity in physical modeling synthesis is not new. The earliest attempt at a complete synthesis system, due to Cadoz [1], had such a modularity principle explicitly built in—the basic elements were masses and springs, through the interconnection of which more elaborate instruments could be constructed. The MOSAIC (later Modalys) system [2] uses the same notion, now employing modal synthesis [3]. Modular networks have also been proposed using wave-based methods [4] and time-stepping schemes—see, e.g., [5, 6]. The main difficulty is in simulating a network connected in a feedback configuration—a problem compounded when nonlinearities, essential for any musically-interesting sound output, are present. The complications are similar to those which occur in virtual analog modeling, but now in a mechanical setting, where individual elements have a distributed character.

Two main technical concerns emerge: one is ensuring numerical stability, as numerical methods for nonlinear systems are prone to explosive unstable solution growth. Another is computational efficiency; iterative methods (such as, e.g., Newton-Raphson [7]) commonly used in nonlinear ODE/PDE solvers, can lead to large increases in computational cost. For the problem of numerical stability, energy techniques are probably the only known method. Such methods, in different guises, have been popular in virtual analog applications (port-Hamiltonian approaches [8], wave digital filtering [9, 10], and used in PDE solvers [11]). For general nonlinearities, however, including those involving collision or intermittent contact, energy methods are available, but usually require the use of iterative methods. See, e.g., [12, 13, 14].

Recently, a new class of methods mas been proposed in the context of virtual analog modeling, relying on energy quadratisation [15], leading to numerical methods which are resolvable without recourse to iterative methods, and which maintain the notion of an energy balance leading to a numerical stability guarantee—see [16, 17], as well as [18] in the context of audio systems. Such an approach reduces computational costs by as much as an order of magnitude, and allows for the simulation of relatively complex nonlinear systems in real time on standard hardware.

In Section 2, a modular physical modeling synthesis system is presented, consisting of an interconnection of string- or bar-like primitives, and with nonlinear connections which allow intermittent contact. A numerical discretisation scheme is presented in Section 3, allowing for energy-stable non-iterative simulation. Simulation results, illustrating characteristic behaviour of such a modular synthesis system appear in Section 4. Performance results for a C implementation are presented in Section 5, demonstrating the possibility of real-time operation of relatively complex instrument designs. Some perspectives appear in Section 6.

2. A MODULAR INSTRUMENT MODEL

The canonical building block here is the linear bar or string (referred to henceforth here as a stiff string), of circular cross section, assumed to vibrate transversely in a single polarisation. The basic equation of motion (see, e.g., [11], as well as other closely-related forms [19, 20]), under unforced conditions, is of the form

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

where the linear partial differential operator $\mathcal{L}$ is defined as

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

Here, $u(x,t)$ is the transverse displacement of the stiff string, in m, as a function of time $t \geq 0$, in s, and spatial coordinate $x \in$

* M. Ducceschi was supported by an Early Career Fellowship from the Leverhulme Trust.

$\mathcal{D} = [0,\ L]$, for some string length L in m. ∂_t and ∂_x represent partial differentiation with respect to t and x, respectively.

The various parameters which define the stiff string are the material density ρ, in kg$\cdot$ m^{-3}, Young's modulus E in Pa, tension T in N, and string radius r in m, from which follow the cross-sectional area $A = \pi r^2$ and moment of inertia $I = \frac{1}{4}\pi r^4$. σ_0 and σ_1 are loss parameters, allowing for a simple frequency-dependent loss characteristic (though much more realistic loss models are available [21]). The symbol $\boldsymbol{\gamma}$ is used here to represent the set of parameters defining a particular string; its minimal form is:

$$\boldsymbol{\gamma} = \{\rho, E, r, T, \sigma_0, \sigma_1, L\} \tag{3}$$

Two boundary conditions must be supplied at each end of the string; many choices are possible (see [11]), but in this work, simply supported conditions will be enforced, so that

$$u = \partial_x^2 u = 0 \qquad \text{at} \qquad x = 0, L \tag{4}$$

The equation (1) requires two initial conditions: $u(x, 0)$ and $\partial_t u(x, 0)$. In a synthesis setting, initial conditions are usually set to zero, but will be maintained here in order to examine energetic behaviour under unforced conditions.

System (1) satisfies an energy balance. Multiplying (1) by $\partial_t u$, and integrating over $\mathcal{D}$ gives

$$\int_{\mathcal{D}} \partial_t u \mathcal{L} u \, dx = 0 \tag{5}$$

Defining the L_2 norm of a function $f(x)$ over the domain $\mathcal{D}$ as

$$\|f\|_{\mathcal{D}} = \sqrt{\int_{\mathcal{D}} f^2 dx} \tag{6}$$

and using integration by parts, as well as the boundary conditions (4) leads to the energy balance

$$d\mathcal{H}_s/dt + \mathcal{Q}_s = 0 \tag{7}$$

where

$$\mathcal{H}_s = \frac{\rho A}{2}\|\partial_t u\|_{\mathcal{D}}^2 + \frac{T}{2}\|\partial_x u\|_{\mathcal{D}}^2 + \frac{EI}{2}\|\partial_x^2 u\|_{\mathcal{D}}^2 \tag{8a}$$

$$\mathcal{Q}_s = 2\rho A \sigma_0 \|\partial_t u\|_{\mathcal{D}}^2 + 2\rho A \sigma_1 \|\partial_x \partial_t u\|_{\mathcal{D}}^2 \tag{8b}$$

$\mathcal{H}_s$ is the total stored energy for the string, and $\mathcal{Q}_s$ is the power loss. Given that $\mathcal{H}_s \geq 0$ and $\mathcal{Q}_s \geq 0$, under unforced conditions, this implies that $d\mathcal{H}_s/dt \leq 0$, and thus energy is monotonically non-increasing, and may be used to bound the growth of the state u itself—such a balance may be employed in discrete time to arrive at energy-based numerical stability conditions. See Section 3.

2.1. Excitation

For system (1), an excitation may be added as

$$\mathcal{L} u = \delta(x - x_e) f_e(t) \tag{9}$$

where here, $f_e(t)$ is an externally-supplied force signal (in N), and $\delta(x - x_e)$, is a spatial Dirac delta function selecting the excitation location $x_e \in [0,\ L]$. The pointwise character of the excitation can be easily extended to the case of distributed contact (as in, e.g., the case of a piano hammer [19] or bow [22]). Simple choices of the excitation function, if intended to model a strike or pluck, are raised cosine distributions of the form

$$f_e(t) = \begin{cases} \frac{1}{2} f_{e,max} \left(1 - \cos\left(\frac{q\pi(t - t_e)}{T_e}\right)\right), & t_e \leq t \leq t_e + T_e \\ 0, & \text{otherwise} \end{cases} . \tag{10}$$

where t_e is the starting time of the excitation, T_e is the duration, $f_{e,max}$ is the maximum force, in N, and where $q = 1$ for a pluck and $q = 2$ for a strike. See Figure 1.

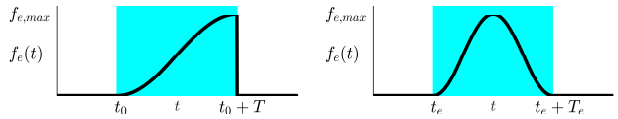

Figure 1: Excitation functions $f_e(t)$ for a pluck (left) and strike (right), as per the form given in (10).

From an energetic standpoint, one may again multiply (9) by $\partial_t u$, and then integrate over $\mathcal{D}$ to obtain the energy balance:

$$\frac{d}{dt}\mathcal{H}_s + \mathcal{Q}_s = \underbrace{\partial_t u(x_e) f_e}_{\triangleq \mathcal{P}} \tag{11}$$

where $\mathcal{H}_s$ and $\mathcal{Q}_s$ are as in (8), and where $\mathcal{P}$ is the instantaneous input power due to the excitation. Thus the rate of growth of energy in the system may be bounded in terms of supplied power.

2.2. Connection to a Lumped Object

Consider now a coupled system of a string under excitation, and in pointwise contact, at location $x = x_c$, with a lumped object of mass M which is constrained to travel parallel to the plane of polarisation of the string, and with displacement $w = w(t)$:

$$\mathcal{L} u = -\delta(x - x_c) f_c + \delta(x - x_e) f_e \qquad M\frac{d^2 w}{dt^2} = f_c \tag{12}$$

Here, $f_c = f_c(\eta)$ is the connection force, assumed dependent on

$$\eta = u(x_c, t) - w(t) \tag{13}$$

the relative displacement between the lumped object and the string at the connection point. See Figure 2.

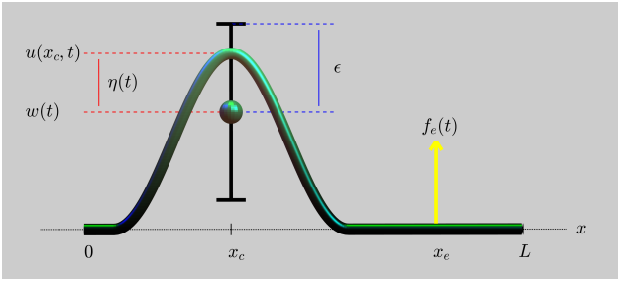

Figure 2: Stiff string, under an excitation force and coupled to a lumped object, as per (12).

Many choices for the interaction force f_c are possible. A basic linear/cubic restoring force served as a basis for the modular network presented in [5]. A more general choice, allowing for intermittent loss of contact (or collisions), and considerably widening

the range of possible sound output, is the following choice:

$$f_c(\eta) = K[|\eta| - \epsilon]_+^{\alpha} \mathrm{sgn}(\eta) \tag{14}$$

Such a force law is parameterised by $K \geq 0$, a stiffness constant, $\alpha \geq 1$, a nonlinearity exponent inspired by models in contact dynamics [23, 19], and $\epsilon \geq 0$, an effective length of the lumped object, which is now able to rattle, and which allows for a dead zone where no force is exerted by the lumped object on the string and vice versa. The notation $[\cdot]_+$ indicates the "positive part of." See Figure 3. Note that when $\epsilon = 0$, so that the lumped mass and string are always in contact, the force (14) reduces to a linear connection when $\alpha = 1$, and a cubic nonlinear force when $\alpha = 3$. It is also possible to go further here, and introduce a model of contact loss, as per similar models in contact dynamics (see, e.g., [23]), and employed in musical instrument modeling [13].

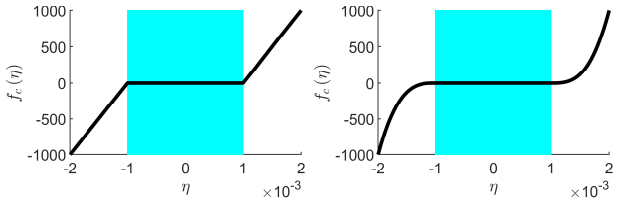

Figure 3: Nonlinear force characteristics of the form given in (14). Left: with $K = 10^6$, $\alpha = 1$ and $\epsilon = 0.001$, and right: with $K = 10^{12}$, $\alpha = 3$ and $\epsilon = 0.001$. The dead zone, over which there is no connection force, is shown as a blue shaded region.

At this point, we introduce the potential $\phi(\eta)$, such that

$$\phi(\eta) \geq 0 \qquad f_c = d\phi/d\eta \tag{15}$$

and note that, through the chain rule,

$$d\phi/dt = f_c d\eta/dt \tag{16}$$

Given the non-negativity of ϕ, it is also possible to define ψ as

$$\psi = \sqrt{2\phi} \qquad f_c = \psi d\psi/d\eta \tag{17}$$

This quadratisation of the potential energy has been employed in the context of virtual analog modeling and musical acoustics in [17, 24, 16], and also, recently, in much more general numerical settings, where it is referred to as *invariant energy quadratisation*—see, e.g., [15]. In the case of the force characteristic given in (14), the corresponding forms of ϕ and ψ are

$$\phi(\eta) = \frac{K}{\alpha+1}[|\eta| - \epsilon]_+^{\alpha+1} \quad \psi(\eta) = \sqrt{\frac{2K}{\alpha+1}}[|\eta| - \epsilon]_+^{\frac{\alpha+1}{2}} \tag{18}$$

Though the forms in (15) and (17) are equivalent, in the numerical context, they lead to distinct energy-conserving methods—in particular, the form in (15) leads to a form requiring iterative solution, and that in (17) to one which may be resolved explicitly. For a direct comparison of this distinction in a simplified case, see the companion paper [18].

From an energetic standpoint, multiplying the first of (12) by $\partial_t u$ and integrating over $\mathcal{D}$ yields

$$d\mathcal{H}_s/dt + \mathcal{Q}_s = \mathcal{P} - f_c \partial_t u(x_c) \tag{19}$$

Multiplying the second by dw/dt gives

$$\frac{d}{dt}\left(\frac{M}{2}\left(\frac{dw}{dt}\right)^2\right) = \frac{dw}{dt} f_c \tag{20}$$

Adding (19) and (20) gives, then

$$\frac{d}{dt}\mathcal{H}_s + \mathcal{Q}_s + \frac{d}{dt}\left(\frac{M}{2}\left(\frac{dw}{dt}\right)^2\right) + f_c\left(\partial_t u(x_c) - \frac{dw}{dt}\right) = \mathcal{P} \tag{21}$$

But, using

$$d\eta/dt = \partial_t u(x_c) - dw/dt \tag{22}$$

as well as (16) leads to the energy balance

$$d\mathcal{H}/dt + \mathcal{Q}_s = \mathcal{P} \tag{23}$$

where

$$\mathcal{H} = \mathcal{H}_s + \mathcal{H}_c \geq 0 \qquad \mathcal{H}_c = \frac{M}{2}\left(\frac{dw}{dt}\right)^2 + \phi \geq 0 \tag{24}$$

Thus the total energy $\mathcal{H}$ of the system may be decomposed into the energy of the stiff string $\mathcal{H}_s$ and that of the connection mechanism $\mathcal{H}_c$, and as before is non-negative—its rate of growth may again be bounded in terms of the input power $\mathcal{P}$. Such a connection mechanism is passive (and furthermore lossless).

2.3. A Complete Modular Network

The step to constructing an arbitrary network is relatively straightforward, using the model above as a starting point.

Suppose a network is defined with N_s stiff string elements, each characterised by a parameter set $\boldsymbol{\gamma}^{(q)}$, $q = 1, \ldots, N_s$, of the form given in (3). For each element, the associated transverse displacement is $u^{(q)} = u^{(q)}\left(x^{(q)}, t\right)$, for $x^{(q)} \in \mathcal{D}^{(q)} = [0,\ L^{(q)}]$, and the associated partial differential operator is $\mathcal{L}^{(q)}$. Suppose also that there are N_m lumped objects, of mass $M^{(j)}$ kg, and of displacement $w^{(j)}(t)$, $j = 1, \ldots, N_m$. See Figure 4.

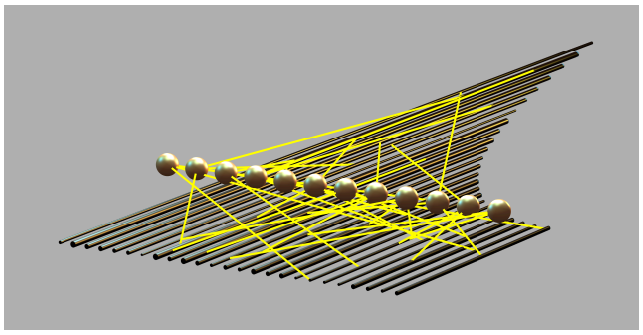

Figure 4: Modular network of stiff strings and lumped masses, with connections indicated as yellow lines.

Now suppose that there are N_c individual connections, each associated with a force $f_c^{(\nu)}$, $\nu = 1, \ldots, N_c$. The νth connection links the q_νth stiff string, at location $x^{(q_\nu)} = x_c^{(\nu)}$ with the j_νth lumped object, where $q_\nu \in \{1, \ldots, N_s\}$ and $j_\nu \in \{1, \ldots, N_m\}$. The force $f_c^{(\nu)}$ will be of the form

$$f_c^{(\nu)} = f_c^{(\nu)}\left(\eta^{(\nu)}\right) \quad \eta^{(\nu)} = u^{(q_\nu)}\left(x_c^{(\nu)}\right) - w^{(j_\nu)} \tag{25}$$

The force $f_c^{(\nu)}\left(\eta^{(\nu)}\right)$ is assumed to be related to an underlying potential energy $\phi^{(\nu)}\left(\eta^{(\nu)}\right) \geq 0$ as

$$f_c^{(\nu)} = d\phi^{(\nu)}/d\eta^{(\nu)} \tag{26}$$

In this case, $\phi^{(\nu)}$ will be chosen to be of the form (18), but any non-negative form is possible. Furthermore, define the index sets

$$\mathbb{I}_c^{(q)} = \{\nu \in \{1,\ldots,N_c\} \,|\, q_\nu = q\} \tag{27a}$$

$$\mathbb{I}_c^{(j)} = \{\nu \in \{1,\ldots,N_c\} \,|\, j_\nu = j\} \tag{27b}$$

Finally, assume N_e excitation functions $f_e^{(\xi)}$, $\xi = 1,\ldots,N_e$, acting on the stiff strings with index q_ξ where $q_\xi \in \{1,\ldots,N_s\}$ at locations $x_e^{(\xi)}$. Define also the index set

$$\mathbb{I}_e^{(q)} = \{\xi \in \{1,\ldots,N_e\} \,|\, q_\xi = q\} \tag{28}$$

The excitation functions $f_e^{(\xi)}$ could be chosen as in (10).

The complete system is:

$$\mathcal{L}^{(q)}u^{(q)} = -\sum_{\nu\in\mathbb{I}_c^{(q)}} f_c^{(\nu)}\delta\left(x^{(q)} - x_c^{(\nu)}\right) + \sum_{\xi\in\mathbb{I}_e^{(q)}} f_e^{(\xi)}\delta\left(x^{(q)} - x_e^{(\xi)}\right) \tag{29a}$$

$$M^{(j)}\frac{d^2w^{(j)}}{dt^2} = \sum_{\nu\in\mathbb{I}_c^{(j)}} f_c^{(\nu)} \tag{29b}$$

Energy analysis is an extension of the case of the single string. Multiplying (29a) by $\partial_t u^{(q)}$, and integrating over $\mathcal{D}^{(q)}$ gives

$$\frac{d}{dt}\mathcal{H}_s^{(q)} + \mathcal{Q}_s^{(q)} + \sum_{\nu\in\mathbb{I}_c^{(q)}} f_c^{(\nu)}\partial_t u^{(q)}\left(x_c^{(\nu)}\right) = \mathcal{P}^{(q)} \tag{30}$$

where $\mathcal{H}_s^{(q)}$ and $\mathcal{Q}_s^{(q)}$ are stored energy and power loss for stiff string q, as defined in (8), with parameter set $\boldsymbol{\gamma}^{(q)}$, and where $\mathcal{P}^{(q)}$ is of the form given in (21). Summing over all strings gives

$$\frac{d}{dt}\mathcal{H}_s + \mathcal{Q}_s + \sum_{q=1}^{N_s}\sum_{\nu\in\mathbb{I}_c^{(q)}} f_c^{(\nu)}\partial_t u^{(q)}\left(x_c^{(\nu)}\right) = \mathcal{P} \tag{31}$$

where

$$\mathcal{H}_s = \sum_{q=1}^{N_s}\mathcal{H}_s^{(q)} \quad \mathcal{Q}_s = \sum_{q=1}^{N_s}\mathcal{Q}_s^{(q)} \quad \mathcal{P} = \sum_{q=1}^{N_s}\mathcal{P}^{(q)} \tag{32}$$

are the total stored energy, dissipated power and supplied power for the set of stiff strings. Recognising that the sets $\mathbb{I}_c^{(q)}$ form a partition of the set $\{1,\ldots,N_c\}$, one has, simply,

$$\frac{d}{dt}\mathcal{H}_s + \mathcal{Q}_s + \sum_{\nu=1}^{N_c} f_c^{(\nu)}\partial_t u^{(q_\nu)}\left(x_c^{(\nu)}\right) = \mathcal{P} \tag{33}$$

Similarly, multiplying (29b) by $dw^{(j)}/dt$, and summing over $j \in \{1,\ldots,N_m\}$, again using the partitioning property of $\mathbb{I}_c^{(j)}$, gives

$$\frac{d}{dt}\sum_{j=1}^{N_m}\left(\frac{M^{(j)}}{2}\left(\frac{dw^{(j)}}{dt}\right)^2\right) - \sum_{\nu=1}^{N_c} f_c^{(\nu)}\frac{d}{dt}w^{(j_\nu)} = 0 \tag{34}$$

Adding (33) and (34) gives, using (25) and (26),

$$d\mathcal{H}/dt + \mathcal{Q}_s = \mathcal{P} \tag{35}$$

where

$$\mathcal{H} = \mathcal{H}_s + \mathcal{H}_c \geq 0 \qquad \mathcal{H}_c = \sum_{j=1}^{N_m}\frac{M}{2}\left(\frac{dw^{(j)}}{dt}\right)^2 + \sum_{\nu=1}^{N_c}\phi^{(\nu)} \geq 0 \tag{36}$$

3. DISCRETE-TIME SIMULATION

Finite difference schemes for the stiff string are covered in detail in [11], and will be reviewed only briefly here.

Assume first a sample rate f_s (and associated time step $k = 1/f_s$), to be employed uniformly across all components in the network. Consider now a single stiff string as defined by (1). The grid function u_l^n represents an approximation to $u(x,t)$ at $t = nk$ and $x = lh$, for integer n and l, and where h is the grid spacing. In particular, $0 \leq l \leq N$, where $h = L/N$, for integer N. The discrete domains d, $\bar{d}$ and $\underline{d}$ are defined as

$$d = \{0,\ldots,N\} \quad \bar{d} = \{0,\ldots,N-1\} \quad \underline{d} = \{1,\ldots,N-1\} \tag{37}$$

A discrete inner product and norm may be defined over a discrete domain $b \subset \mathbb{Z}$ with grid spacing h (such as d, $\bar{d}$ or $\underline{d}$ in (37) above), for grid functions f_l^n, g_l^n, as

$$\langle f^n, g^n\rangle_b = \sum_{l\in b} h f_l^n g_l^n \qquad \|f^n\|_b = \sqrt{\langle f^n, f^n\rangle_b} \tag{38}$$

3.1. Shift and Difference Operators

The forward and backward time-shift operators e_{t+} and e_{t-} may be defined, with regard to the grid function u_l^n, as

$$e_{t+}u_l^n = u_l^{n+1} \qquad e_{t-}u_l^n = u_l^{n-1} \tag{39}$$

Forwards, backwards and centered approximations to a first time derivative may be defined in terms of shifts as

$$\delta_{t+} = \frac{e_{t+}-1}{k} \quad \delta_{t-} = \frac{1-e_{t-}}{k} \quad \delta_{t\circ} = \frac{e_{t+}-e_{t-}}{2k} \tag{40}$$

and averaging operators μ_{t+} and μ_{t-} and an approximation δ_{tt} to a second derivative may be written as

$$\mu_{t+} = \frac{e_{t+}+1}{2} \qquad \mu_{t-} = \frac{1+e_{t-}}{2} \qquad \delta_{tt} = \delta_{t+}\delta_{t-} \tag{41}$$

Similarly, spatial shifts e_{x+} and e_{x-} may be defined as

$$e_{x+}u_l^n = u_{l+1}^n \qquad e_{x-}u_l^n = u_{l-1}^n \tag{42}$$

and approximations to the first and second spatial deriavtaives as

$$\delta_{x+} = \frac{e_{x+}-1}{h} \quad \delta_{x-} = \frac{1-e_{x-}}{h} \quad \delta_{xx} = \delta_{x+}\delta_{x-} \tag{43}$$

3.2. Basic Scheme

A basic explicit finite difference scheme for an uncoupled linear stiff string, as in (1) may be written, for $u = u_l^n$, $l \in d$ as

$$\underbrace{\left(\rho A \delta_{tt} - T\delta_{xx} + EI\delta_{xx}^2 + 2\rho A\sigma_0 \delta_{t\circ} - 2\rho A\sigma_1 \delta_{t-}\delta_{xx}\right)}_{\ell} u = 0 \tag{44}$$

Numerical boundary conditions corresponding to (4) are

$$u_l^n = \delta_{xx} u_l^n = 0 \qquad l = 0, N \tag{45}$$

and allow (44) to be well-defined even when applied at or adjacent to the domain end points.

The scheme (44) satisfies an energy balance of the form

$$\delta_{t+} \mathfrak{h}_s^{n-1/2} + \mathfrak{q}_s^n = 0 \tag{46}$$

where $\mathfrak{h}_s^{n-1/2}$ and $\mathfrak{q}_s^n$ are the discrete-time energy storage function and power loss, respectively, defined as

$$\begin{aligned} \mathfrak{h}_s^{n-\frac{1}{2}} &= \frac{\rho A}{2}\|\delta_{t-} u^n\|_d^2 + \frac{T}{2}\langle \delta_{x+} u^n, e_{t-}\delta_{x+} u^n\rangle_{\bar{d}} \qquad (47a) \\ &\quad + \frac{EI}{2}\langle \delta_{xx} u^n, e_{t-}\delta_{xx} u^n\rangle_{\underline{\bar{d}}} - \rho A\sigma_1 k\|\delta_{t-}\delta_{x+} u\|_{\bar{d}}^2 \\ \mathfrak{q}_s^n &= 2\rho A\sigma_0\|\delta_{t\circ} u\|_d^2 + 2\rho A\sigma_1\|\delta_{t\circ}\delta_{x+} u\|_{\bar{d}}^2 \qquad (47b) \end{aligned}$$

The power loss $\mathfrak{q}_s^n$ is non-negative, but the stored energy $\mathfrak{h}_s$ is only non-negative under the condition

$$h^2 \geq \frac{1}{2}\left(\frac{Tk^2}{\rho A} + 4\sigma_1 k + \sqrt{\left(\frac{Tk^2}{\rho A} + 4\sigma_1 k\right)^2 + 16\frac{EIk^2}{\rho A}}\right) \tag{48}$$

which is a numerical stability condition for scheme (44) in isolation. Under this condition, the state size may be bounded in terms of supplied energy.

In implementation, scheme (44) is explicit—if the state u_l^n is written as a column vector $\mathbf{u}^n = [u_0^n, \ldots, u_N^n]^T$, (44) becomes

$$\mathbf{u}^{n+1} = \mathbf{B}\mathbf{u}^n + \mathbf{C}\mathbf{u}^{n-1} \tag{49}$$

where $\mathbf{B}$ and $\mathbf{C}$ are sparse $(N+1) \times (N+1)$ matrices. See [11] for the construction of these matrices in terms of the various spatial difference operators.

3.3. String with Nonlinear Connection and Excitation

Consider now the simple case of a stiff string under an excitation force and a nonlinear connection, as per (12). A discrete form is

$$\ell u = -f_c \frac{1}{h}\mathcal{I}(x_c) + f_e \frac{1}{h}\mathcal{I}(x_e) \qquad M\delta_{tt} w = f_c \tag{50}$$

Here, $f_c = f_c^n$ is a time series representing an approximation to $f_c(t)$, of a nature to be described shortly, and $f_e = f_e^n$ is an approximation to the externally-supplied function $f_e(t)$, perhaps obtained through sampling. The grid functions $\mathcal{I}_l(x_0)$, selects a given location x_0. Many choices are possible, but the simplest, and that which will be employed subsequently here, is certainly

$$\mathcal{I}_l(x_0) = \begin{cases} 1, & l = l_0 = \text{round}(x_0/h) \\ 0, & \text{otherwise} \end{cases} \tag{51}$$

When multiplied by $1/h$, as in (50) above, approximations to the Dirac delta functions from (12) result, where $x_0 = x_c, x_e$.

The force f_c^n will be dependent on the relative displacement η of the string and lumped mass at the connection grid index $l = l_c$:

$$\eta^n = u_{l_c}^n - w^n \tag{52}$$

Consider now the inner product of the first of (50) with $\delta_{t\circ} u$ over d—this leads to the energy balance

$$\delta_{t+}\mathfrak{h}_s^{n-1/2} + \mathfrak{q}_s^n + f_c^n \delta_{t\circ} u_{l_c}^n = \underbrace{f_e^n \delta_{t\circ} u_{l_e}^n}_{\mathfrak{p}^n} \tag{53}$$

where $\mathfrak{h}_s^{n-1/2}$ and $\mathfrak{q}_s^n$ are as defined in (47), and where $\mathfrak{p}^n$ is the discrete input power due to the excitation.

Similarly, multiplying the second of (50) by $\delta_{t\circ} w^n$ leads to

$$\delta_{t+}\left(\frac{M}{2}(\delta_{t-} w^n)^2\right) - f_c \delta_{t\circ} w^n = 0 \tag{54}$$

Adding (53) and (54), and using (52) leads to the balance

$$\delta_{t+}\left(\mathfrak{h}_s^{n-1/2} + \frac{M}{2}(\delta_{t-} w^n)^2\right) + \mathfrak{q}_s^n + f_c^n \delta_{t\circ}\eta^n = \mathfrak{p}^n \tag{55}$$

What is lacking is a definition of f_c^n. Here are two:

$$f_c^n = \frac{\delta_{t+}\phi^{n-1/2}}{\delta_{t\circ}\eta} \qquad f_c^n = \mu_{t+}\psi^{n-1/2}\frac{\delta_{t+}\psi^{n-1/2}}{\delta_{t\circ}\eta^n} \tag{56}$$

Here, $\phi^{n-1/2}$ is an approximation to ϕ at time $t = (n-1/2)k$, such as $\phi^{n-1/2} = \mu_{t-}\phi(\eta^n) \geq 0$, and $\psi^{n-1/2}$ is an approximation to ψ at time $t = (n-1/2)k$. When inserted in (55) above, either form in (56) leads to the energy balance

$$\delta_{t+}\mathfrak{h}^{n-1/2} + \mathfrak{q}_s^n = \mathfrak{p}^n \qquad \mathfrak{h}^{n-1/2} = \mathfrak{h}_s^{n-1/2} + \mathfrak{h}_c^{n-1/2} \tag{57}$$

where, for the two choices of the force f_c^n from (56),

$$\mathfrak{h}_c^{n-1/2} = \frac{M}{2}(\delta_{t-} w^n)^2 + \begin{cases} \phi^{n-1/2} \\ \frac{1}{2}\left(\psi^{n-1/2}\right)^2 \end{cases} \geq 0 \tag{58}$$

Under the further condition (48), then $\mathfrak{h}_s \geq 0$ implying that $\mathfrak{h} \geq 0$, and as before, the system as a whole will be numerically stable.

3.4. Implementation: Iterative vs. Non-iterative Methods

When written in vector/matrix form, the scheme (50) has the form

$$\begin{aligned} \mathbf{u}^{n+1} &= \mathbf{B}\mathbf{u}^n + \mathbf{C}\mathbf{u}^{n+1} - \theta \mathbf{j}_c f_c^n + \theta \mathbf{j}_e f_c^n \qquad (59a) \\ w^{n+1} &= 2w^n - w^{n-1} + \frac{k^2}{M} f_c^n \qquad (59b) \end{aligned}$$

where $\theta = k^2/(\rho A h(1+\sigma_0 k))$, and where $\mathbf{j}_e$ and $\mathbf{j}_c$ are $N+1$ element column vectors, all zero except for a 1 in locations l_e and l_c respectively.

Now, using the fact that

$$\delta_{t\circ}\eta^n = \frac{1}{2k}\left(\mathbf{j}_c^T\left(\mathbf{u}^{n+1} - \mathbf{u}^{n-1}\right) - w^{n+1} + w^{n-1}\right) \tag{60}$$

and the updates (59) leads to the affine relationship between $\delta_{t\circ}\eta^n$ and f_c^n:

$$\delta_{t\circ}\eta^n + \zeta f_c^n + \Xi^n = 0 \tag{61}$$

where Ξ^n consists of previously computed (known) values of $\mathbf{u}$ and samples of the excitation f_e^n, and ζ is a constant.

Consider now the first energy-conserving definition of f_c^n, from (56). In combination with (61), this leads to the nonlinear equation:

$$\underbrace{r + 2k\zeta \frac{\phi\left(r + \eta^{n-1}\right) - \phi\left(\eta^{n-1}\right)}{r} + 2k\Xi^n}_{F(r)} = 0 \tag{62}$$

where $r = \eta^{n+1} - \eta^{n-1}$ is the unknown. Such a form has appeared in earlier works regarding energy-conserving collision simulation [12, 13], and requires, in general, an iterative solution through, e.g., Newton-Raphson; the number of iterations depends strongly on the particular parameters defining the nonlinearity.

The second form of f_c given in (56) leads to a distinct update. Define

$$g^n = \frac{\delta_{t+}\psi^{n-1/2}}{\delta_{t\circ}\eta^n} \triangleq \frac{d\psi}{d\eta}\,|_{t=nk} \tag{63}$$

The time series g^n can be computed directly from known values of η^n and the functional form of ψ. Given g^n, the following pair of updates results from (61) and (63):

$$\delta_{t\circ}\eta^n + \zeta g^n \mu_{t+}\psi^{n-1/2} + \Xi^n = 0 \qquad \delta_{t+}\psi^{n-1/2} = g^n \delta_{t\circ}\eta^n \tag{64}$$

These can be consolidated into a single update yielding η^{n+1} from previously computed values of η and ψ:

$$\eta^{n+1} = \Theta\left(\eta^n, \eta^{n-1}, \psi^{n-1/2}\right) \tag{65}$$

and, given η^{n+1}, the second of (64) may be updated to yield $\psi^{n+1/2}$, at which point the simulation advances to the next time step. For more details, see [18].

3.5. A Complete Discrete-time Modular Network

Returning to the complete modular system presented in Section 2.3, the numerical scheme corresponding to (29) is

$$\begin{aligned} \ell^{(q)} u^{(q)} &= -\sum_{\nu \in \mathbb{I}_c^{(q)}} f_c^{(\nu)} \frac{1}{h^{(q)}} \mathcal{I}\left(x_c^{(\nu)}\right) + \sum_{\xi \in \mathbb{I}_e^{(q)}} f_e^{(\xi)} \frac{1}{h^{(q)}} \mathcal{I}\left(x_e^{(\xi)}\right) \\ \delta_{tt} w^{(j)} &= \frac{1}{M^{(j)}} \sum_{\nu \in \mathbb{I}_c^{(j)}} f_c^{(\nu)} \end{aligned} \tag{66}$$

Here, $u^{(q)} = u_l^{(q),n}$ is the grid function corresponding to the qth string, with grid spacing $h^{(q)}$, and defined over $l \in \{0, \ldots, N^{(q)}\}$ and $\ell^{(q)}$ is the difference operator derived from $\mathcal{L}^{(q)}$. $w^{(j)} = w^{(j),n}$ is the time series corresponding to the displacement of the jth lumped object, $j = 1, \ldots, N_m$ and $f_c^{(\nu)}$, $\nu = 1, \ldots, N_c$ to the associated connection forces. In (3.5) above, the time step n accompanying all dependent variables is suppressed for brevity.

Each of the N_c connection forces $f_c^{(\nu)} = f_c^{(\nu),n}, \nu = 1 \ldots, N_c$ is assumed dependent on a displacement

$$\eta^{(\nu),n} = u_{l_c^{(\nu)}}^{(q_\nu),n} - w^{(j_\nu),n} \tag{67}$$

There is assumed an underlying discrete potential $\phi^{(\nu),n-1/2}$ (and accompanying function $\psi^{(\nu),n-1/2}$), such that, for the non-iterative algorithm,

$$f_c^{(\nu),n} = \mu_{t+}\psi^{(\nu),n-1/2} \frac{\delta_{t+}\psi^{(\nu),n-1/2}}{\delta_{t\circ}\eta^{(\nu),n}} \tag{68}$$

Full details of the implementation and energy analysis will not be presented here, because of space limitations, but follow in almost all respects from the analysis of the single string/connection described in the previous section. A demonstration of energy conservation appears in Section 4.1, and of the benefit of a non-iterative algorithm, in terms of computation time, in Section 5.

4. SIMULATION RESULTS

In this section, we present some numerical results illustrating numerical and musical aspects of the modular system described here. All simulations are run at 44.1 kHz.

4.1. Energy Conservation

Consider first the energy conservation property, for a full network of 25 strings (ranging in pitch from C2 to C4), and using 24 two-sided connections, where the non-iterative scheme is employed. A worst case is assumed here, where the strings are lossless, and the initial conditions of the strings and lumped elements are randomised. In Figure 5, the energy partition as a function of time between strings and connections is shown, alongside the normalised energy variation, illustrating energy conservation to machine accuracy. The availability of such a measure allows for an excellent approach to debugging modular synthesis codes.

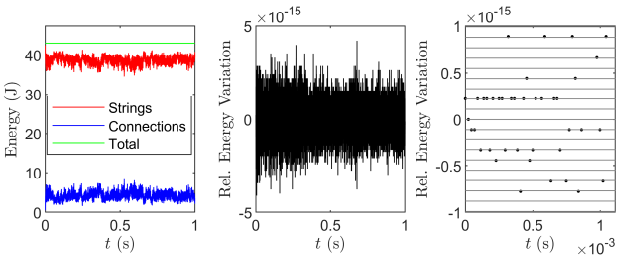

Figure 5: Left: energy partition, showing energy in strings, connections, and the total. Middle: normalised sample to sample energy variation, and right: a detail showing energy conservation to machine accuracy.

4.2. Single String/Lumped Object

An illustration of the interaction between a single string and a lumped object in intermittent contact is shown in Figure 6. Even in this extremely simple case, a wide variety of timbres is possible. In Figure 7, spectrograms of sound output are shown where string parameters are chosen according to a high E string on a guitar.

4.3. Large network

Consider now a network of 25 strings with 24 double-sided connections, representing approximately the largest possible in real time at 44.1 kHz. Snapshots of the time evolution of this network are shown in Figure 8.

5. COMPUTATIONAL ANALYSIS

This section details the computational costs of the element types (stiff string and connection) from which the system is built. We

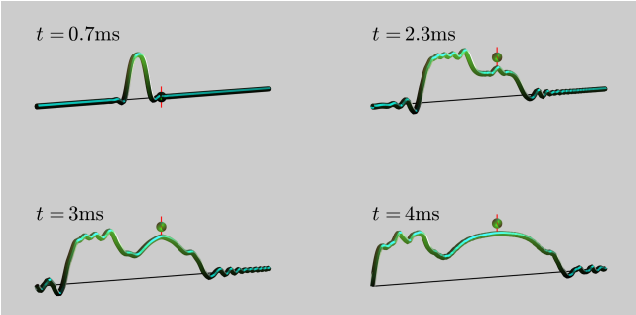

Figure 6: Interaction of a single stiff string with a lumped colliding object, at times as indicated.

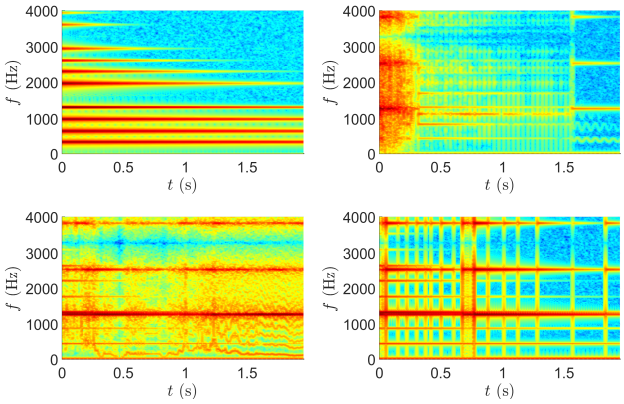

Figure 7: Spectrograms of sound output for a high E guitar string connected to a single lumped element. Top left: free vibration. Top right, under a single connection, illustrating abrupt changes in timbre over a long time scale. Bottom left: illustrating low-frequency "warbling" effects. Bottom right: for a high mass lumped object, leading to distinct bounces or temporal events.

start with a set of strings, and then compare an individual nonlinear connection solver in both iterative and non-iterative form. Finally we assess a full system of strings and connecting elements, as could be used in a real-time audio plug-in. All simulations are written in C++, and timings are produced by running at a sample rate of 44.1 kHz (in double precision) over 441,000 time steps and then dividing by 10 to give an average performance for 44,100 steps. The test CPU is a 6-core Intel Xeon E5-1650 v2, and the LLVM compiler was used. Whilst we will not consider multithreading here (as this presents problems for real-time usage) we do make use of vectorization, namely with Intel's SSE and AVX intrinsics [25].

The stiff string is simulated using the scheme (44), which is explicit, and amenable to spatial parallelization. The state size and computational cost can become large, but not unmanageably so: for the case of 25 low-pitched strings tuned from C2 at 65Hz to middle C4, the total state size required is 2,550 values. Table 1 shows results using various compiler optimisation levels, and also using manually written vector intrinsics which lead ultimately to a speedup by a factor of four. Performance is faster than real time.

Consider now a single nonlinear connection. Two forms of code were tested: first a form which uses an iterative Newton-

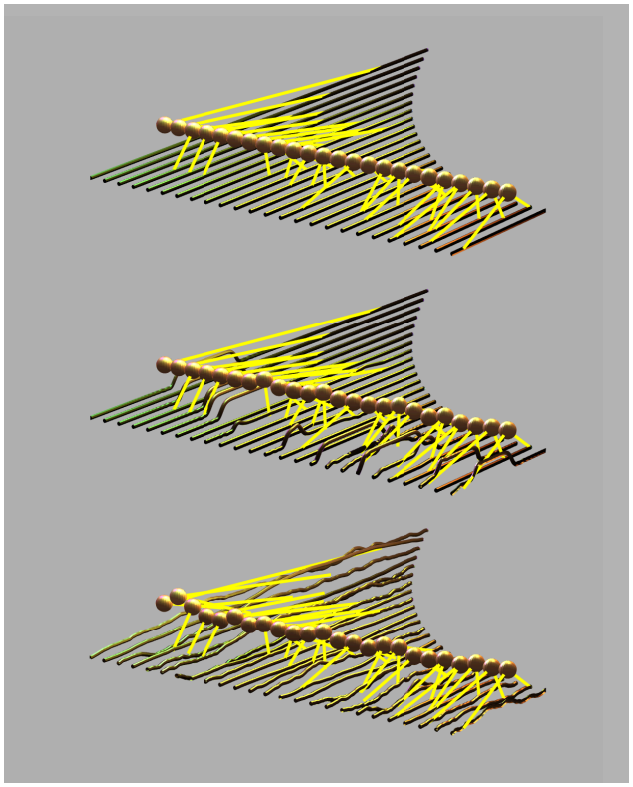

Figure 8: Large modular network, at rest (top), under initial excitations (middle) and later (bottom).

Table 1: *Computation times for 25 strings, tuned from C2 to C4, for 1 s output. Total state size: 2,550 grid points.*

Optimisation Level	Time (s)
-O0	0.59
-O3 SSE only enabled	0.18
Manual AVX	0.13

Raphson solver, and then a form which uses the new non-iterative solver. For the purposes of this testing the Newton-Raphson solver was run for 10 iterations at each time step, starting from a cached solution. This is a usable "average" figure for the simulation, although in practice could be larger, and also possibly reduced by testing for a residual.

The algorithm for solving for each connection force consists of some initial arithmetic operations and two calls to the `power()` function with non-integer exponents (using `std::pow()` from the cmath library). Then the Newton-Raphson solver is launched. This requires a further two calls of the `power()` function at each iteration. Finally there are some further arithmetic calculations to compute the forces at each end of the connection. The non-iterative form is considerably simpler, requiring just two `power()` functions in total. Table 2 shows the resulting computation times. The benefit of the non-iterative form is clear, giving a $11\times$ speed up over the iterative form. This is mostly due to the reduction in the number of calls to the `power()` function, from 22 in the iterative form to just two in the latter. Further optimisation may be

Table 2: *Compute times for an individual connection for 1 s.*

Optimisation	Iterative (ms)	Non-Iterative (ms)	Speedup
-O0	79.2	7.1	11.2×
-O3 SSE	73.5	6.3	11.6×

achieved by implementing a manually vectorized power function across groups of connection calculations.

The final test simulation involved two octaves of strings with 24 two-sided rattle connections, such that each string was connected to another through a lumped element—a single strike to any string will excite the entire network, as in Figure 8. At each time step the strings are updated first, followed by a loop over the connections which then adds back the calculated forces into the strings at the connection points. Table 3 shows the results for the full system.

Table 3: *Computation times: 25 strings/24 connections for 1 s.*

Optimisation	Iterative (s)	Non-Iterative (s)	Speedup
-O0	2.53	0.81	3.1×
-O3 SSE	1.93	0.35	5.5×
Manual AVX	1.84	0.28	6.6×

The version computed using an iterative solver results in a best time of 1.84 s for 1 s sound output, which is clearly well short of the real time threshold. The non-iterative solver results in a 6.6× speedup, and the resulting time of 0.28 seconds is just within the scope of usability for a real-time application. Note that performance levels may vary according to other factors such as the CPU-specific cache, and other compiler details.

6. CONCLUDING REMARKS

A form of this system available as a real-time plug-in is under development through Physical Audio [26], called *net2*. It is rewritten from a previous version (called *net1*), which ran offline in multicore using an iterative solver. The ability to sidestep such iterative methods, and maintain a numerically-stable synthesis algorithm has allowed this move to real time through the approximately 10× speedup for the nonlinear part of the simulation. Though only stiff strings have been shown here, the extension to systems including multiple vibrating plates is immediate—see, e.g., [5]. Also, only one type of nonlinearity has been presented here, but the non-iterative algorithm presented here is fully general.

A major consideration, at the level of the user experience, is in the UI design, particularly when there are potentially many stiff string/connection elements to manage—this has been partly dealt with during the design of the simpler Derailer system, also from Physical Audio. A deeper issue is that of exploring the design parameter space—for the system illustrated here with 25 strings and 24 connections, for example, there are 319 parameters to set. Some heuristics can be employed to make this more manageable (by, say, restricting string tunings), but clearly some more general strategy for finding interesting regions of the parameter space (perhaps invoking methods from machine learning) is necessary.

7. REFERENCES

[1] C. Cadoz, A. Luciani, and J.-L. Florens, "Responsive input devices and sound synthesis by simulation of instrumental mechanisms," *Comp. Music J.*, vol. 8, no. 3, pp. 60–73, 1983.

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

[3] J.-M. Adrien, "The missing link: Modal synthesis," in *Representations of Musical Signals*, G. DePoli, A. Picialli, and C. Roads, Eds., pp. 269–297. MIT Press, Cambridge, Massachusetts, 1991.

[4] R. Rabenstein, S. Petrausch, A. Sarti, G. De Sanctis, C. Erkut, and M. Karjalainen, "Block-based physical modeling for digital sound synthesis," *IEEE Sig. Proces. Mag.*, vol. 24, no. 2, pp. 42–54, 2007.

[5] S. Bilbao, "A modular percussion synthesis environment," in *Proc. Int. Digital Audio Effects Conf.*, Como, Italy, 2009, pp. 321–328.

[6] S. Bilbao, A. Torin, P. Graham, J. Perry, and G. Delap, "Modular physical modeling synthesis on gpu," in *Proc. Int. Comp. Music Conf.*, Athens, Greece, 2014.

[7] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 1992.

[8] R. Müller and T. Hélie, "Power-balanced modelling of circuits as skew gradient systems," in *Proc. Int. Conf. Digital Audio Effects*, Aveiro, Portugal, 2018, pp. 1–8.

[9] K. Werner, V. Nangia, J. O. Smith, and J. Abel, "A general and explicit formulation for wave digital filtres with multiple/multiport nonlinearities and complicated topologies," in *Proc. Workshop Appl. Signal Proces. Audio Acoust.*, Mohonk, NY, 2015, pp. 1–5.

[10] A. Bernardini, K. Werner, A. Sarti, and J. O. Smith, "Modeling nonlinear wave digital elements using the lambert function," *IEEE Trans. Circ. Syst. I*, vol. 63, no. 8, pp. 1231–1242, 2016.

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

[12] V. Chatziioannou and M. van Walstijn, "An energy conserving finite difference scheme for simulation of collisions.," in *Proc. Stockholm Musical Acoust. Conf.*, Stockholm, Sweden, 2013.

[13] S. Bilbao, A. Torin, and V. Chatziioannou, "Numerical modeling of collisions in musical instruments," *Acta Acustica u. with Acustica*, vol. 101, no. 1, pp. 155–173, 2015.

[14] M. Ducceschi and S. Bilbao, "Modelling collisions of nonlinear strings against rigid barriers: Conservative finite difference schemes with application to sound synthesis," in *Proc. Int. Conf. On Acoust. (ICA 2016)*, Buenos Aires, Argentina, September 2016.

[15] X. Yang, "Linear and unconditionally energy stable schemes for the binary fluid-surfactant phase field model," *Comp. Methods Appl. Mech. Eng.*, vol. 318, pp. 1005–1029, 2017.

[16] N. Lopes, T. Hélie, and A. Falaize, "Explicit second-order accurate method for the passive guaranteed simulation of port-hamiltonian systems," in *Proc. 5th IFAC 2015*, Lyon, France, July 2015.

[17] A. Falaize, *Modélisation, simulation, génération de code et correction de systèmes multi-physiques audios: Approche par réseau de composants et formulation Hamiltonienne À Ports*, Ph.D. thesis, Université Pierre et Marie Curie, Paris, July 2016.

[18] M. Ducceschi and S. Bilbao, "Non-iterative solvers for nonlinear problems: The case of collisions," 2019, Under review, 22nd Int. Conf. Digital Audio Effects.

[19] A. Chaigne and A. Askenfelt, "Numerical simulations of struck strings. I. A physical model for a struck string using finite difference methods," *J. Acoust. Soc. Am.*, vol. 95, no. 2, pp. 1112–1118, 1994.

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

[21] C. Vallette, "The mechanics of vibrating strings," in *Mechanics of Musical Instruments*, A. Hirschberg, J. Kergomard, and G. Weinreich, Eds., pp. 116–183. Springer, New York, New York, 1995.

[22] C. Desvages and S. Bilbao, "Two-polarisation physical model of bowed strings with nonlinear contact and friction forces, and application to gesture-based sound synthesis," *Appl. Sci.*, vol. 6, pp. 135, 2016.

[23] K. Hunt and F. Crossley, "Coefficient of restitution interpreted as damping in vibroimpact," *ASME J. Appl. Mech.*, pp. 440–5, June 1975.

[24] A. Falaize and T. Hélie, "Passive Guaranteed Simulation of Analog Audio Circuits: A Port-Hamiltonian Approach," *Appl. Sci.*, vol. 6, pp. 273 – 273, 2016.

[25] N Firasta, M. Buxton, P. Jinbo, K. Nasri, and S. Kuo, "Intel AVX: New frontiers in performance improvements and energy efficiency," *Intel White Paper*, 2008.

[26] "Physical audio," `www.physicalaudio.co.uk`, Accessed: 2019-03-26.