REAL-TIME MODAL SYNTHESIS OF NONLINEARLY INTERCONNECTED NETWORKS

Michele Ducceschi

Department of Industrial Engineering
University of Bologna
Bologna, Italy
michele.ducceschi@unibo.it

Stefan Bilbao

Acoustics and Audio Group
University of Edinburgh
Edinburgh, United Kingdom
sbilbao@ed.ac.uk

Craig J. Webb

Physical Audio Ltd.
London, United Kingdom
craig@physicalaudio.co.uk

ABSTRACT

Modal methods are a long-established approach to physical modeling sound synthesis. Projecting the equation of motion of a linear, time-invariant system onto a basis of eigenfunctions yields a set of independent forced, lossy oscillators, which may be simulated efficiently and accurately by means of standard time-stepping methods. Extensions of modal techniques to nonlinear problems are possible, though often requiring the solution of densely coupled nonlinear time-dependent equations. Here, an application of recent results in numerical simulation design is employed, in which the nonlinear energy is first quadratised via a convenient auxiliary variable. The resulting equations may be updated in time explicitly, thus avoiding the need for expensive iterative solvers, dense linear system solutions, or matrix inversions. The case of a network of interconnected distributed elements is detailed, along with a real-time implementation as an audio plugin.

1. INTRODUCTION

Modal methods are a well-known approach to physical modeling synthesis. In this framework, akin to spectral-like techniques [1], a distributed linear, time-invariant system is described as a superposition of spatial eigenfunctions called the "modes" of the system [2]. These may be computed offline, along with a set of modal weights depending on static parameters describing the input and output (I/O) locations for the system. The resulting equations take the form of a set of independent forced, damped oscillators. Setting appropriate natural frequencies and frequency-dependent damping ratios is simple in this framework. This structure lends itself naturally to fast time-stepping implementations [3], which can be made virtually dispersion-free by employing exact integrators [4, 5]. These desirable numerical properties and the relative ease of implementation made modal methods attractive in early approaches to physical modeling sound synthesis, when frameworks such as Mosaic and Modalys emerged [6, 7]. Further simulation methods, such as the Functional Transformation Method, share some common features with modal synthesis [8, 9], though the equations are here updated in time via inverse Laplace transforms.

In spite of many desirable properties, modal methods possess some limitations. Most notably, the modes can be obtained analytically only in a few cases of interest in musical acoustics (though the eigenvalue problem may still be solved numerically, using e.g. finite differences or finite elements [10]). Furthermore, efficiency quickly deteriorates when the I/O's need to be rendered dynamically. Finally, modal methods yield a rather cumbersome structure of densely nonlinearly coupled equations when applied to nonlinear systems [11], the topic of this work. With regard to this latter issue, exceptions exist, such as the Kirchhoff-Carrier string model [4] (Chapter 8), or the case of strings colliding against a barrier with a linear restoring force [12]. Perceptually-motivated assumptions may reduce the computational burden of nonlinear modal systems, such as the simplified nonlinear model adopted in [13] for the simulation of piano strings. More efficient direct numerical methods, primarily finite differences, are often preferred in such cases[4].

Copyright: © 2023 Michele Ducceschi et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, adaptation, and reproduction in any medium, provided the original author and source are credited.

This work presents a fast implementation of modal techniques in the case of nonlinearly coupled distributed elements. While modularity in physical modelling has long roots (see e.g. [14, 15, 16, 17]), this work presents an efficient, energy-stable algorithm for large, nonlinearly coupled modal systems. To this end, a numerical method analogous to the Scalar Auxiliary Variable (SAV) approach is employed [18]. Recently, this method was applied to Hamiltonian systems with non-negative potential energy, yielding computation times on par with simple explicit designs such as the Störmer-Verlet method while conserving energy to machine accuracy and allowing the extraction of sufficient conditions for numerical stability [19]. This is enabled by an appropriate "quadratisation" of the nonlinear potential energy, via an auxiliary state function to be updated independently. The resulting equations of motion may be updated explicitly, through fast linear system solution techniques that exploit matrix structure. The numerical methods described here form the basis for an efficient real-time audio plugin.

The article is structured as follows: Section 2 introduces continuous models of the distributed canonical elements to be used to construct the network. Useful identities for modal projections are given alongside energy considerations. A model for the connections between elements is also detailed. Section 3 introduces the equations for the network, as well as a suitable "quadratised" form of the resulting modal equations, in state-space form. An energy-stable, efficient time discretisation of the same system is described in detail in Section 4, followed by several numerical examples. Finally, a discussion of the architecture of a real-time plugin, written in C++ and suitable for use in commercial digital audio workstations is detailed in Section 5.

2. MODELS

In this section, the basic components of a modular network are described: these are the resonators, including bars and plates, and connections.

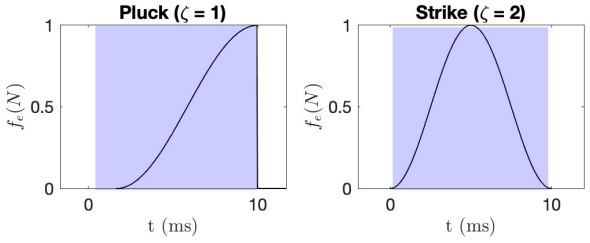

Figure 1: *Excitation functions of the form given in* (4)*. Here,* $T_e = 10$ *ms,* $f_{e,max} = 1$ *N.*

2.1. Resonators

The resonating elements considered here are α-dimensional distributed objects, with $\alpha \in \{1, 2\}$ (that is, $\alpha = 1$ for a string or bar and $\alpha = 2$ for a membrane or plate). A model for such distributed elements is given as [4]:

$$\mathcal{G}u = \delta_e f_e(t), \quad \text{where} \quad \mathcal{G} := \rho\,\partial_t^2 - \mathcal{L} + \rho\sigma\,\partial_t. \tag{1}$$

Here, $u(\mathbf{x}, t) : \mathcal{V} \times \mathbb{R}_0^+ \to \mathbb{R}$ is the displacement of the element, in m, measured transversely from the rest position. u is a function of coordinates $\mathbf{x} = [x_1, \ldots, x_\alpha] \in \mathcal{V} \subset \mathbb{R}^\alpha$ as well as time $t \geq 0$. For simplicity, it is assumed that $\mathcal{V}$ represents an α-dimensional rectangular domain such that $0 \leq x_\kappa \leq L_\kappa$, with $1 \leq \kappa \leq \alpha$. The dynamics of the element are encapsulated by the differential operator $\mathcal{G}$, where ∂_t represents partial differentiation with respect to time t. ρ is the α-dimensional density, in kg·m$^{-\alpha}$, and $\sigma \geq 0$ is a loss parameter, with units of s^{-1}. Furthermore,

$$\mathcal{L} = T\Delta - D\Delta^2, \tag{2}$$

where Δ is the α-dimensional Laplace operator. Here, T, in N·m$^{1-\alpha}$ is a tension constant, and D, in N·m$^{3-\alpha}$ is a rigidity constant. Boundary conditions of simply-supported type are employed:

$$u = \partial_n^2 u = 0, \quad \text{for} \quad t \geq 0, \mathbf{x} \in \mathcal{B}. \tag{3}$$

Here, $\mathcal{B}$ is the boundary of $\mathcal{V}$, and ∂_n represents partial spatial differentiation in a direction perpendicular to the boundary. Finally, f_e represents the excitation function, given here as a raised (or half-raised) squared sine distribution, modeling either a strike or pluck:

$$f_e(t) = f_{e,max} \sin^2\left(\frac{\zeta\pi\,(t - t_e)}{2T_e}\right) \tag{4}$$

for $t_e \leq t \leq t_e + T_e$, and is 0 otherwise. Here t_e in s is the starting time of the excitation, T_e in s is the duration, and $f_{e,max}$ is the maximum force, in N, and where $\zeta = 1$ for a pluck and $\zeta = 2$ for a strike, see Figure 1. Other forms for f_e will be considered subsequently here, including pure sinusoids or sawtooth waves.

The force is assumed to act over a distribution δ_e, here idealized as an α-dimensional Dirac delta function acting at the excitation location $\mathbf{x} = \mathbf{x}_e$:

$$\delta_e := \delta(\mathbf{x} - \mathbf{x}_e). \tag{5}$$

2.1.1. Energy balance of the isolated resonator

Under this choice of boundary conditions, model (1) satisfies the following energy balance:

$$\frac{d(\int_\mathcal{V} \mathcal{H}\,\mathrm{d}\mathbf{x})}{dt} = \underbrace{-\rho\sigma \int_\mathcal{V} (\partial_t u)^2 \mathrm{d}\mathbf{x}}_{\mathcal{Q}} + \underbrace{\partial_t u(\mathbf{x}_e, t) f_e(t)}_{\mathcal{P}}, \tag{6}$$

where the energy density $\mathcal{H}$ is given as [3]:

$$\mathcal{H} = \frac{\rho}{2}(\partial_t u)^2 + \frac{T}{2}|\nabla u|^2 + \frac{D}{2}(\Delta u)^2. \tag{7}$$

Under unforced conditions, the injected power $\mathcal{P}$ is zero and the system is strictly dissipative (since $\mathcal{Q} > 0$), leading to boundedness of the solutions.

2.1.2. Modal expansion and identities

Consider now a modal expansion for the displacement, where each mode is represented by a product of one spatial and one time component:

$$u(\mathbf{x}, t) = \boldsymbol{\chi}^\intercal(\mathbf{x})\,\mathbf{q}(t). \tag{8}$$

Here, $\boldsymbol{\chi}$, $\mathbf{q}$ are column vectors of length M. This is a finite integer, to be specified in terms of the stability requirements of the associated time-stepping scheme, as will be shown below. Let the modal index be $m \in [1, ..., M]$. With the mth mode, there is associated a set of integers $\{\mu_1^m \ldots, \mu_\alpha^m\}$ (the modal numbers). Since simply-supported boundary conditions are assumed, one has

$$\chi_m(\mathbf{x}) := \prod\nolimits_{\kappa=1}^{\alpha} \sqrt{\frac{2}{L_\kappa}} \sin\frac{\mu_\kappa^m \pi x_\kappa}{L_\kappa}, \;\; \mu_\kappa^m \in \mathbb{N}. \tag{9}$$

From (9), it immediately follows that

$$\int_\mathcal{V} \boldsymbol{\chi}\boldsymbol{\chi}^\intercal\,\mathrm{d}\mathbf{x} = \mathbf{I}, \quad \int_\mathcal{V} \boldsymbol{\chi}\,\mathcal{L}(\boldsymbol{\chi}^\intercal)\,\mathrm{d}\mathbf{x} = -\rho\boldsymbol{\Omega}^2, \tag{10}$$

where $\mathbf{I}$ is the $M \times M$ identity matrix, and where $\boldsymbol{\Omega}$ is a diagonal $M \times M$ matrix whose diagonal elements $[\boldsymbol{\Omega}]_{mm}$ are the natural radian frequencies ω_m, defined as:

$$\omega_m = \sqrt{\frac{T}{\rho}\sum_{\kappa=1}^{\alpha}\left(\frac{\mu_\kappa^m \pi}{L_\kappa}\right)^2 + \frac{D}{\rho}\left(\sum_{\kappa=1}^{\alpha}\left(\frac{\mu_\kappa^m \pi}{L_\kappa}\right)^2\right)^2}. \tag{11}$$

In the above, the modal indices μ_κ^m are found by sorting the eigenfrequencies, such that $\omega_1 \leq \omega_2 \leq ... \leq \omega_M$. In the one-dimensional case, $\mu_1^m = m$. In the two-dimensional case, the modal indices μ_1^m, μ_2^m associated with each mode m depend on the aspect ratio, see Figure 2 for an example of this.

2.2. Connections

Connections here take the form of nonlinear springs transferring vibrations across the network. These are energy-storing devices, for which the resulting force may be given as the gradient of a potential, as

$$f(\eta) = -\frac{d\phi}{d\eta}, \tag{12}$$

where η is the elongation of the spring. In this work, intermittent contact is permitted, such that

$$\phi(\eta) = \frac{K}{\gamma + 1}\,[|\eta| - \beta]_+^{\gamma+1} + \frac{\epsilon_0}{2} \geq 0. \tag{13}$$

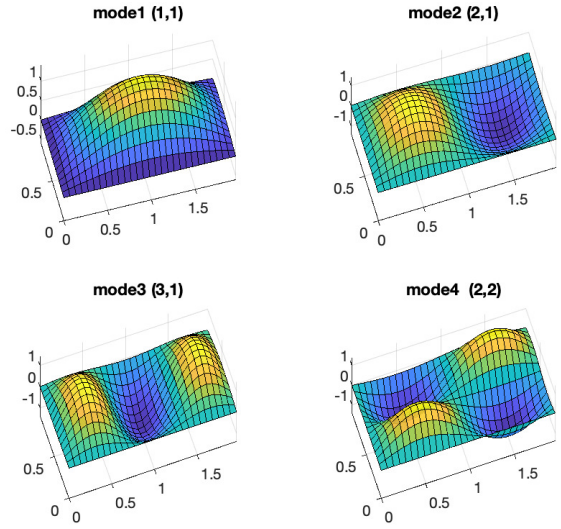

Figure 2: *First four modes of the membrane / plate, with aspect ratio $L_1/L_2 = 2$. The corresponding modal indices μ_κ^m are given in brackets.*

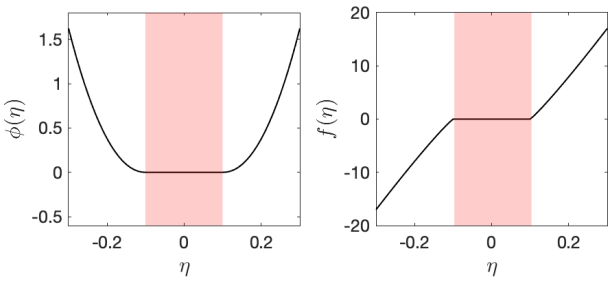

Figure 3: *Example of nonlinear potential* (13) *and corresponding force* (12). *Here, $K = 10^2$, $\gamma = 1.1$, $\beta = 0.1$, $\epsilon_0 = 0$. The shaded area, whose width is given by 2β, represents a dead zone (no force exerted).*

Here, $K \geq 0$ is a stiffness constant, $\gamma \geq 1$ is a nonlinear exponent and $\beta \geq 0$ is a gap. The gauge constant $\epsilon_0 \geq 0$ is introduced to shift the zero-point of the potential without affecting the force f in (12) (the role of such constant will be briefly discussed below). The potential and the resulting force are represented in Figure 3. Note that no force is exerted by the spring when $|\eta| < \beta$, resulting in a rattling-type force [12, 16]. When $\beta = 0$, linear and cubic springs are recovered by setting $\gamma = \{1, 3\}$, respectively.

3. MODULAR NETWORKS

The resonators described above can now be interconnected in a modular fashion, using groups of connections. Assume N_u such distributed elements, with displacement $u^{(j)}(\mathbf{x}^{(j)}, t)$ and described by the linear operator $\mathcal{G}^{(j)}$, defined over $\mathcal{V}^{(j)}$, $j \in [0, \dots, N_u]$. The excitation of the jth element, of the form (4) is here denoted as $f_e^{(j)}$, and is distributed according to $\delta_e^{(j)}$. Assume furthermore to have a number N_c of connections, indexed by $\nu = 1, ..., N_c$. The νth connection connects element j_ν to j'_ν, for $j_\nu, j'_\nu \in [0, \dots, N_u]$

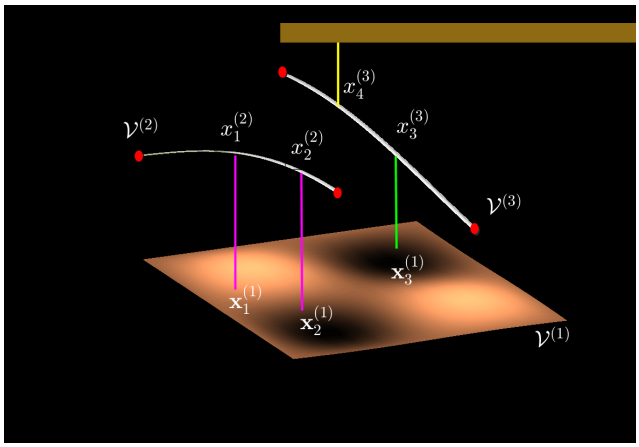

Figure 4: *Example of a network comprising $N_u = 3$ distributed elements (here, one plate and two strings), occupying the domains $\mathcal{V}^{(i)}, i = \{1, 2, 3\}$. The total number of connections is $N_c = 4$, including a connection to a fixed reference frame.*

(the special case in either one of j_ν, j'_ν is zero will be treated shortly). The connection locations are $\mathbf{x}_\nu \in \mathcal{V}^{(j_\nu)}$ and $\mathbf{x}'_\nu \in \mathcal{V}^{(j'_\nu)}$. The force experienced by the distributed elements j_ν, j'_ν due to connection ν is then of the form (12), so that:

$$f^{(\nu)}(\eta^{(\nu)}) = -\frac{d\phi^{(\nu)}}{d\eta^{(\nu)}}, \tag{14}$$

with

$$\eta^{(\nu)} := u^{(j_\nu)}(\mathbf{x}_\nu, t) - u^{(j'_\nu)}(\mathbf{x}'_\nu, t). \tag{15}$$

Furthermore, define the index set:

$$\mathbb{I}_c^{(j)} = \{\nu \in \{1, \dots, N_c\} \,|\, j_\nu = j\} \tag{16a}$$

and let

$$\delta^{(\nu)} := \delta(\mathbf{x}^{(j_\nu)} - \mathbf{x}_\nu), \ \nu \in \mathbb{I}_c^{(j)}. \tag{17}$$

and Given these, the equation of motion of the j^{th} element can be given as

$$\mathcal{G}^{(j)}(u^{(j)}) = \sum_{\nu \in \mathbb{I}_c^{(j)}} \delta^{(\nu)} f^{(\nu)} + \delta_e^{(j)} f_e^{(j)}, \tag{18}$$

Note that, in this framework, the the special case $j'_\nu = 0$ refers to a connection to a fixed reference frame, as seen in Figure 4, for which

$$\eta^{(\nu, j'_\nu = 0)} := u^{(j_\nu)}(\mathbf{x}_\nu, t). \tag{19}$$

It is convenient to define a single scalar potential $\phi = \phi(\boldsymbol{\eta}) : \mathbb{R}^{N_c} \to \mathbb{R}_0^+$, incorporating the potentials from all the connections in the network. This is

$$\phi = \sum_{\nu=1}^{N_c} \phi^{(\nu)}. \tag{20}$$

An energy balance for the network is obtained after multiplying (18) by $\partial_t u^{(j)}$, integrating, and summing all the equations in the network. This yields:

$$\frac{d(\sum_{j=1}^{N_u} \int_{\mathcal{V}^{(j)}} \mathcal{H}^{(j)} \mathrm{d}\mathbf{x}^{(j)} + \phi)}{dt} = \sum_{j=1}^{N_u} \left(-\mathcal{Q}^{(j)} + \mathcal{P}^{(j)}\right), \tag{21}$$

where $\mathcal{Q}^{(j)}$, $\mathcal{P}^{(j)}$ have expressions analogous to those in (6) for the element in isolation. When $\mathcal{P}^{(i)} = 0\ \forall i$, the energy is non-increasing, leading to boundedness of the solutions.

3.1. Energy Quadratisation

Central to the time-stepping scheme presented below is the idea of "quadratisation" of the potential energy. This techinque is often referred to as the Scalar Auxiliary Variable (SAV) method, proposed originally for dissipative phase-field models [18], and later applied to Hamiltonian systems [19]. For that, define

$$\psi = \sqrt{2\phi}. \tag{22}$$

Under such definition, an application of the chain rule allows one to write the forces in terms of ψ, as

$$f^{(\nu)} = -\psi \frac{\partial \psi}{\partial \eta^{(\nu)}} := -\psi\, g^{(\nu)}, \tag{23}$$

and note that ψ, like ϕ is a *scalar* function of $\boldsymbol{\eta} := [\eta^{(1)}, ..., \eta^{(N_c)}]$. When $\phi = \psi^2/2$ is substituted in (21), the energy includes quadratic terms only. Note as well that the auxiliary variable ψ evolves in time according to

$$\dot{\psi} = \mathbf{g}^\intercal\, \dot{\boldsymbol{\eta}}. \tag{24}$$

3.2. Modal Equations

A set of time-dependent modal equations is obtained from (18), after left-multiplying by $\boldsymbol{\chi}^{(j)}$ and integrating over $\mathcal{V}^{(j)}$. By virtue of (10), one gets

$$\Gamma^{(j)}(\mathbf{q}^{(j)}) = \sum_{\nu \in \mathbb{I}_c^{(j)}} \mathbf{Y}^{(\nu)} g^{(\nu)} \psi + \boldsymbol{\chi}^{(j)}(\mathbf{x}_e^{(j)}) f_e^{(j)}, \tag{25}$$

where:

$$\Gamma^{(j)}(\mathbf{q}^{(j)}) = \rho^{(j)} \left(\ddot{\mathbf{q}}^{(j)} + \mathbf{C}^{(j)} \dot{\mathbf{q}}^{(j)} + (\boldsymbol{\Omega}^{(j)})^2 \mathbf{q}^{(j)} \right). \tag{26}$$

In the above, $\mathbf{q}^{(j)}$ is the vector of modal coordinates of the jth distributed element, of length $M^{(j)}$. Furthermore:

$$\mathbf{Y}^{(\nu)} := \int_{\mathcal{V}^{(j_\nu)}} \boldsymbol{\chi}^{(j_\nu)} \delta^{(\nu)}\, \mathrm{d}\mathbf{x}^{(j_\nu)},\ \nu \in \mathbb{I}_c^{(j)}. \tag{27}$$

Above, the matrix $\mathbf{C}$ is a positive diagonal matrix whose diagonal elements include the modal damping coefficients, thus generalising the simple (i.e. frequency independent) loss profile given in (1). The modal equations of all the N_u distributed elements can now be consolidated into a single system:

$$\mathbf{M} \left(\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\Omega}^2 \mathbf{q} \right) = -\mathbf{Y}\mathbf{g}\, \psi + \mathbf{f}_e(t), \tag{28}$$

where $\mathbf{M}$ is a fully diagonal mass matrix including the $\rho^{(j)}$'s; $\mathbf{C}$ and $\boldsymbol{\Omega}$ are fully diagonal matrices including the losses and radian frequencies, sorted accordingly. The vector $\mathbf{f}_e$ is obtained by concatenating $\boldsymbol{\chi}^{(j)}(\mathbf{x}_e^{(j)}) f_e^{(j)}$, $j = 1, ..., N_u$. Finally, $\mathbf{Y}$ is a (generally dense) $M \times N_c$ matrix, where $M = \sum_{j=1}^{N_u} M^{(j)}$. Note as well that, in modal form, (24) becomes

$$\dot{\psi} = (\mathbf{Y}\mathbf{g})^\intercal\, \dot{\mathbf{q}}. \tag{29}$$

3.2.1. State-space form

In view of the numerical application presented below, it is worth recasting (28) and (29) in state-space form. This is:

$$\dot{\mathbf{q}} = \mathbf{M}^{-1}\mathbf{p}, \tag{30a}$$

$$\dot{\mathbf{p}} = -\mathbf{C}\mathbf{p} - \mathbf{K}\mathbf{q} - \mathbf{Y}\mathbf{g}\, \psi + \mathbf{f}_e(t), \tag{30b}$$

$$\dot{\psi} = (\mathbf{Y}\mathbf{g})^\intercal \mathbf{M}^{-1}\mathbf{p}, \tag{30c}$$

where the stiffness matrix was conveniently defined as $\mathbf{K} := \mathbf{M}\,\boldsymbol{\Omega}^2$. In this configuration, to a vector of inputs $\mathbf{f}_e(t)$ corresponds a vector $\mathbf{y}^{(j)}(t)$ of $N_o^{(j)}$ outputs of the j^{th} element, defined as

$$y_i^{(j)}(t) = (\boldsymbol{\chi}^{(j)}(\mathbf{x}_i^{(j)}))^\intercal \mathbf{q}^{(j)}(t). \tag{31}$$

System (30) is linear in the state variables $\mathbf{q}, \mathbf{p}, \psi$. The nonlinearities are now expressed solely via the vector $\mathbf{g}$. Such linearity is the key feature of the underlying numerical scheme, in that if $\mathbf{g}$ is assumed known at all times, then the system may be solved by a simple matrix inversion. Since here $\mathbf{M}$ is diagonal, the scheme overall becomes explicit, as will be seen shortly.

An energy balance for the modal system may be obtained immediately, after left-multiplying (30b) by $\left(\mathbf{M}^{-1}\,\mathbf{p}\right)^\intercal$. Applying simple identities, one gets:

$$\frac{d}{dt}\left(\frac{1}{2}\mathbf{p}^\intercal \mathbf{M}^{-\intercal}\mathbf{p} + \frac{1}{2}\mathbf{q}^\intercal \mathbf{K}\mathbf{q} + \frac{\psi^2}{2} \right) = -\mathcal{Q} + \mathcal{P}, \tag{32}$$

where $\mathcal{Q} = \mathbf{p}^\intercal \mathbf{M}^{-\intercal}\mathbf{C}\mathbf{p}$, $\mathcal{P} = \mathbf{p}^\intercal \mathbf{M}^{-\intercal}\mathbf{f}_e$. One may of course obtain the same results by carrying out the integrals directly in (21), when the integration over all the $\mathcal{V}^{(i)}$'s is performed using the eigenfunctions (9), and after quadratisation of the nonlinear energy as per (22).

4. DISCRETE-TIME EQUATIONS

A discretisation of system (30) follows as a direct application of the time-stepping procedure detailed in [19]. This method allows us to update the discrete equations in time explicitly, while guaranteeing a passive form of the discrete energy balance. Time is now discretised according to a sample rate $f_s := 1/k$, where k is the sampling period, in s. Then, $n \in \mathbb{N}$ defines the time index, such that solutions are evaluated at the time $t_n := kn$. An interleaved time grid will also be employed, at half-integer steps, as $t_{n+1/2} := k(n + 1/2)$. Time difference operators are introduced as follows, for a time series $\mathbf{u}^n$ defined on the integer time grid:

$$\mathfrak{d}_+ \mathbf{u}^n = \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{k}. \tag{33}$$

An analogous definition holds for an interleaved time series $\mathbf{v}^{n-1/2}$, such that $\mathfrak{d}_+ \mathbf{v}^{n-1/2} = (\mathbf{v}^{n+1/2} - \mathbf{v}^{n-1/2})/k$. An averaging operator for an interleaved time series is also given as

$$\mathfrak{m}_+ \mathbf{v}^{n-\frac{1}{2}} = \frac{\mathbf{v}^{n+\frac{1}{2}} + \mathbf{v}^{n-\frac{1}{2}}}{2}. \tag{34}$$

Given these definitions, a discretisation of (30) is given as:

$$\mathfrak{d}_+ \mathbf{q}^n = \mathbf{M}^{-1}\mathbf{p}^{n+\frac{1}{2}}, \tag{35a}$$

$$\mathfrak{d}_+ \mathbf{p}^{n-\frac{1}{2}} = -\mathbf{C}\,\mathfrak{m}_+ \mathbf{p}^{n-\frac{1}{2}} - \mathbf{K}\mathbf{q}^n - \mathbf{Y}\mathbf{g}^n\,\mathfrak{m}_+ \psi^{n-\frac{1}{2}} + \mathbf{f}_e^n, \tag{35b}$$

$$\mathfrak{d}_+ \psi^{n-\frac{1}{2}} = (\mathbf{Y}\mathbf{g}^n)^\intercal\, \mathbf{M}^{-1} \mathfrak{m}_+ \mathbf{p}^{n-\frac{1}{2}}. \tag{35c}$$

This discretisation is energy-passive, in that left-multiplying (35c) by $(\mathbf{M}^{-1}\mathfrak{m}_+\mathbf{p}^{n-\frac{1}{2}})^\intercal$ leads to [19]:

$$\mathfrak{d}_+\left(\frac{1}{2}(\mathbf{p}^{n-\frac{1}{2}})^\intercal\mathbf{M}^{-\intercal}\mathbf{p}^{n-\frac{1}{2}}+\frac{1}{2}\mathbf{q}^{n-1}\mathbf{K}\mathbf{q}^{n}+\frac{(\psi^{n-\frac{1}{2}})^2}{2}\right)= -(\mathfrak{m}_+\mathbf{p}^{n-\frac{1}{2}})^\intercal\mathbf{M}^{-\intercal}\mathbf{C}\,\mathfrak{m}_+\mathbf{p}^{n-\frac{1}{2}}+(\mathfrak{m}_+\mathbf{p}^{n-\frac{1}{2}})^\intercal\mathbf{M}^{-\intercal}\mathbf{f}_e^n, \tag{36}$$

which discretises (32). While the discrete energy comprises quadratic terms only, numerical stability is guaranteed only under non-negativity of the energy overall. Using (35a) to express the momenta $\mathbf{p}$ in terms of $\mathbf{q}$ in the expression for the energy above, one may derive the following bound on the time step [19]:

$$k<\frac{2}{\lambda_{\max}(\mathbf{M}^{-\frac{1}{2}}\mathbf{K}\mathbf{M}^{-\frac{\intercal}{2}})}=\frac{2}{\omega_M}, \tag{37}$$

where $\lambda_{\max}$ denotes the largest eigenvalue, and ω_M is the largest eigenfrequency across all the N_u distributed elements, taking the form (11). This condition arises solely as the particular discretisation adopted here for the linear part, and it guarantees non-negativity of the discrete energy in (36).

The key feature of scheme (35) is the explicit form of $\mathbf{g}$, defined here as:

$$\mathbf{g}^n:=\nabla_{\boldsymbol{\eta}}\psi\Big|_{t=t_n}=\frac{1}{\sqrt{2\phi}}\nabla_{\boldsymbol{\eta}}\phi\Big|_{t=t_n}, \tag{38}$$

where ϕ is as per (20). Note that the division by $\sqrt{2\phi}$ may be ill-defined when the gauge constant $\epsilon_0=0$ in (13). Thus, it may be useful to shift the potential upward. It is known that the gauge constant has an influence on the convergence properties of the quadratised schemes [20], though a thorough discussion on the role of such constant is out-of-scope here. Using (35a) and (35c) in (35b), one arrives at the following:

$$\mathbf{A}^n\mathbf{q}^{n+1}=\mathbf{d}\left(\mathbf{q}^n,\mathbf{q}^{n-1},\mathbf{g}^n,\psi^{n-\frac{1}{2}}\right),\quad\text{with} \tag{39}$$
$$\mathbf{A}^n:=\mathbf{I}+\mathbf{a}\mathbf{b}^\intercal,\ \mathbf{a}:=\frac{k}{2}\mathbf{M}^{-1}\mathbf{Y}\mathbf{g}^n,\ \mathbf{b}:=\frac{k}{2}\mathbf{Y}\mathbf{g}^n.$$

Above, the vector $\mathbf{d}$ contains values of the state vector from previous time steps. As noted above, the update is in the form of a linear system (a rank-1 perturbation of the identity matrix). However, the inversion of $\mathbf{A}$ may be performed in $\mathcal{O}(M)$ operations, using the Sherman-Morrison formula [21, 19], thus yielding a very efficient algorithm, as shown below.

4.1. Numerical Examples

Numerical experiments are given in Figures 5 and 6. There, scheme (35) is compared to the simple Störmer-Verlet method [22], applied directly to the second-order-in-time system (28). This is an explicit method which, in the linear case, has the same stability condition as (37). However, stability is not guaranteed under nonlinear conditions, and unpredictable numerical behaviour ensues for sufficiently large stiffness values, as seen in Figure 6.

In spite of this, the compute times for method (35) are very much on par with the Störmer-Verlet method, since these are both explicit schemes. Remarkably, the computational bottleneck here

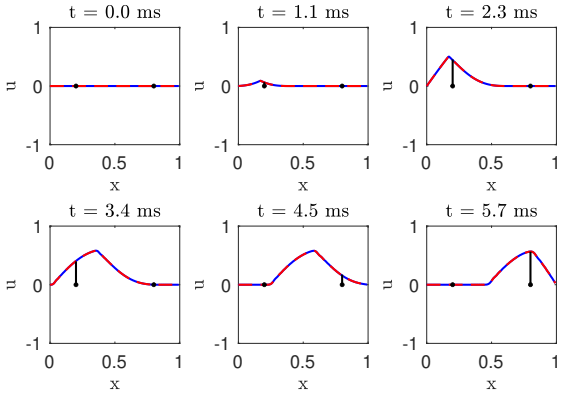

Figure 5: *Time evolution of a string under the influence of $N_s=2$ nonlinear springs (represented as black lines). Solid blue: scheme (35); dashed red: Störmer-Verlet. Here, the string has a fundamental frequency of 100 Hz, a mass of 1 g, and presents a total $M=128$ modes between the fundamental and the limit of stability. The string is activated using a strike with $T_e=5$ ms, $f_{e,max}=200$ N. The springs are located at $0.2L$, $0.8L$, and have $K=7.3\cdot10^{-4}$, $\gamma=1.1$, $\beta=0$. For this simulation, $\epsilon_0=10^9$.*

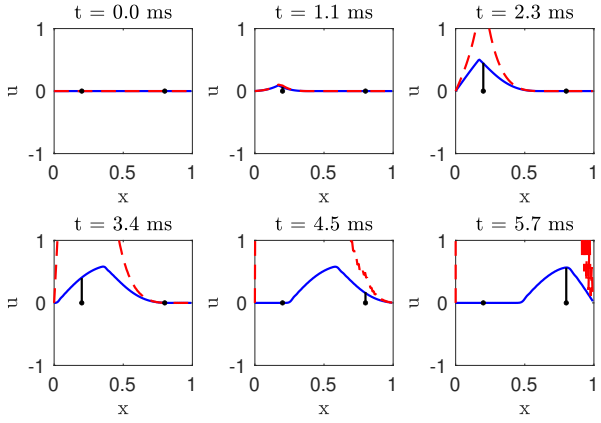

Figure 6: *Same as Figure 5, but here $K=730$.*

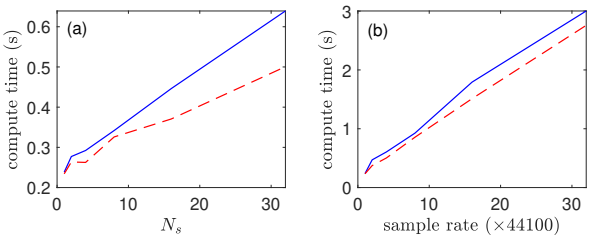

Figure 7: *Matlab compute times for the string of Figure 5. (a): $f_s = 44100$, N_s as indicated. (b): $N_s = 1$, f_s as indicated. Solid blue: scheme* (35)*; dashed red: Störmer-Verlet.*

is not in the the structure of the scheme itself, but rather in the algebraic operations involving the (dense) product $\mathbf{Y}\mathbf{g}^n$, as can be seen from Figure 7. In the figure, it is seen that the slope of the two schemes is identical (and the compute times are very similar) when the number of connections is fixed, and the sample rate is changed. However, when the sample rate is fixed, and the number of connections is varied, scheme (35) presents a somewhat steeper slope than Störmer-Verlet. In practice, though, only a handful of such nonlinear connections is needed for expressive sound synthesis, as will be shown in the examples below.

5. REAL-TIME IMPLEMENTATION

There are a number of considerations that have to be addressed in order to create a real-time instrument in the form of an audio plug-in. Computational expense, polyphonic behaviour and parameter control are all key aspects in developing a usable system. Three different configurations of models were tested: connected strings (SS), strings connected to a plate (SP), and two connected plates (PP), as shown in Figures 8 - 10. In the SS set-up the excitation pluck is applied to string 1 (that is $f_e^{(j)}$ in (18) is identically zero for $j > 1$), with the remaining strings acting as resonators. These can be tuned to different fundamental frequencies as required, with either semitone or fine tuning offsets (this is a trivial procedure in the modal framework). In the SP configuration both strings are plucked and the plate acts as a resonating device. In the final set-up PP an excitation signal is applied to the top plate with the second plate acting as the resonator. This excitation can be in the form of a raised cosine as described above, or indeed some arbitrary signal such as a sawtooth or other waveform.

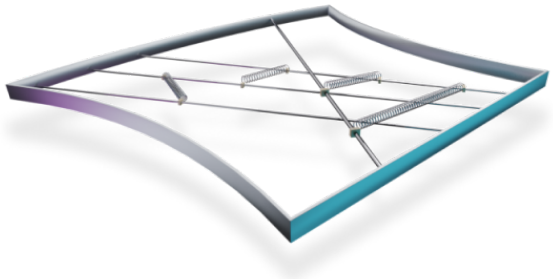

Figure 8: *System of four connected strings.*

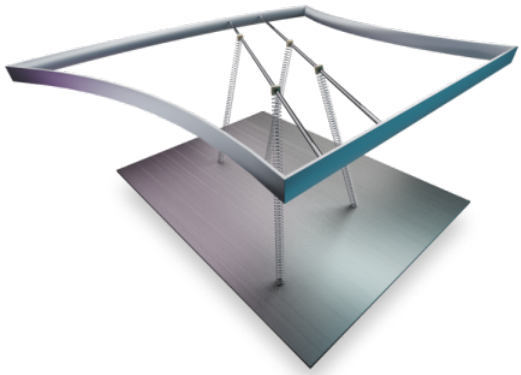

Figure 9: *System of two strings connected to a plate.*

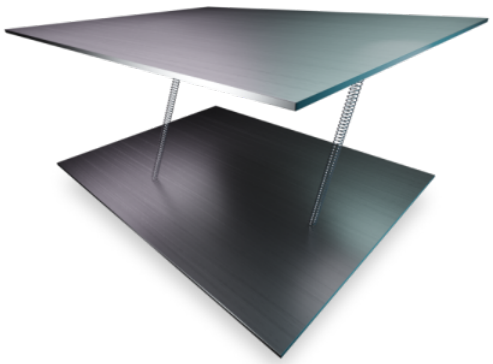

Figure 10: *System of two connected plates.*

Combining all 3 configurations in a single unified instrument provides a suitably wide palette of timbres, from harmonic string tones and pads, percussive strikes and evolving soundscapes. An example of the plugin capabilities is given in the companion page[1]. Each of the configurations requires a different approach to polyphonic behaviour, as described in Section 5.2 below.

5.1. Computational Performance

The core elements of the system which have to be computed at each time-step are mainly simple operations over one-dimensional arrays such as cumulative sum reductions, and also dense matrix-by-vector multiplication ($\mathbf{Y}\mathbf{g}^n$ in (35)). The compute performance varies linearly according to the number of modes used in the system and the number of connecting elements. Ensuring peak performance in C++ requires all aspects of the computation to be fully vectorized. Whilst the majority of the operations can be vectorized by the compiler with a suitable optimisation level (i.e. -Ofast in Clang), some elements do require manual application of vector intrinsics. These are either SSE/AVX intrinsics for Intel builds or NEON intrinsics for native use on Apple Silicon machines. Note

[1] `https://mdphys.org/DAFx23.html`

that double-precision was used throughout due to the high level of nonlinearity in the system.

Initial testing was performed off-line to gauge general CPU performance. The strings model was tested with all 4 strings tuned to either C0 (which used 800 modes) or C3 (which used 184 modes), and with 4 connecting elements. The string-plate model was tested using 1,100 modes and 4 connections, and finally the dual-plate model using 1,300 modes and 2 connections. Test machines were a Mac Pro with a Xeon E5 processor, a MacBook Pro with a Core i7, and Mac mini with Apple Silicon M1 processor. Table 1 shows the resulting computation times.

Table 1: *Computation times for optimised C++ simulations over 44100 time-steps.*

Configuration	Xeon E5	Core i7	Apple M1
Strings C0	0.20s	0.18s	0.11s
Strings C3	0.06s	0.05s	0.04s
Strings-Plate	0.27s	0.26s	0.15s
Plates	0.29s	0.26s	0.17s

These timings are all well within the bounds required for use in a real-time environment, even considering the lower overall clock frequencies which are obtained when running a plug-in inside a Digital Audio Workstation.

5.2. Polyphonic Behaviour

Having tested the models in their basic off-line states, the next stage is to decide how to use them to create a playable instrument. The CPU-usage of the SS model decreases significantly as the fundamental frequency increases. Only the bottom 2 octaves use the full 200 modes, and higher registers require much less. Therefore using a standard voice-based approach is viable in this case, and allows 5 to 6 note polyphony (holding one voice as being ready for reset). At a Note-On event a damped voice is recalculated to the given fundamental and tuning offsets of the connected strings, along with the other parameters that define the state of the string. A re-triggering system is also employed, so that a string may be plucked multiple times without having to set up a new voice.

The SP model, however, requires a very different mechanism. Due to the number of modes used in the plate the system will always use over 1,000 modes. Applying a voice-based system would very quickly consume an entire CPU core, even with just 2 or 3 note polyphony. Instead, a hybrid mechanism was used consisting of a single instance of the SP. In order to achieve multiple octaves of playable notes the strings are disconnected from the plate and retuned on-the-fly. So at a Note-On event the required state of a string is calculated whilst it is disconnected from the system, and it is then reconnected whilst applying the excitation pluck. At the same time the opposing string is damped and disconnected ready for the next Note-On, in a monophonic mode, or left to continue sounding in duo-phonic mode.

Finally the behaviour of the PP configuration depends on the type of excitation. Here again a single instance of the 2 plates is used. Note-On events can trigger strikes, or multiple strikes, for percussive sounds. Separately, a choice of sawtooth or sine wave signals can be used as the forcing excitation, thus allowing multiple octaves of polyphonic signals at very little computational cost using a voice-based setup. Traditional elements such as VCF and distortion units can be employed at this point in the signal chain to give a more flexible soundscape.

5.3. Parameter Control and Modulation

Parameter control of physical modelling systems is always a complicated process. In an optimal case all parameters would continuously affect the sound as the engine is being computed. However, this is often made difficult due to parameters that change the setup of the system state, and for example the mass of a plate or the stiffness of a string. During prototyping 3 different types of parameter control were used; real-time continuous, Note-On, and system-reset.

The core principle of these models is the ability to connect strings and plates together to form systems of sympathetic resonance. The parameters for the connecting elements, such as their strength and rattle gap, can be directly manipulated in real-time. The only requirement is some level of smoothing of the parameter movement to avoid unwanted noise.

Parameters that define the state of a string, such as stiffness (i.e., harmonicity), are defined as Note-On controls. Their value is picked-up from the control at a new Note-On event and the string state is calculated appropriately. This system works well for both the connected strings configuration where it fits naturally into the voice-based mechanism, and for the string-plate where we are disconnecting and reconnecting strings on-the-fly.

A further aspect of parameter control is modulation. One essential example is being able to perform pitchbend and vibrato on the strings. In a modal system this is straightforward as one has access to the modal frequencies and their weights directly in the sound engine. By computing a vector of frequencies at pitchbend up, and another at pitchbend down, a simple linear interpolation can be used to obtain the correct frequency during run-time. Further modulation effects can be obtained by varying the modal weights over time.

Figure 11: *Prototype of a connected strings model running as a VST3 plug-in.*

Parameters that control the state of a plate are more complicated as the objects are acting as continuous resonators in both configurations. One possible approach would be to store multiple tables of modal frequencies and weights for various control settings and interpolate between them. This approach has been used in our previous work on plate reverberation [3] to allow the plate

size to be dynamically adjusted without resetting the plate and disrupting the audio signal. A simpler method is to perform a fast damp and reset of the system when these parameters change, and then re-exciting the model with the new settings. This was used in prototyping due the number of plate controls which are available.

6. CONCLUDING REMARKS

This work has illustrated an application of newly devised schemes for the fast simulation of mechanical systems with non-negative potential energy. The schemes were applied to the modal equations of a nonlinearly coupled network of distributed elements, forming the basis of an advanced physical modelling synthesizer. The proposed schemes yield compute times close to those of simpler, though numerically unstable designs such as Störmer-Verlet. A working real-time plugin, suitable for use in most current digital audio workstations, has also been illustrated. It has been shown that this fast mathematical model allows the simulation of thousands of nonlinearly coupled modes, while keeping the CPU usage low enough for real-time performance. The wide sonic palette of the synthesizer, in conjunction with the flexibility of the modal approach, makes the plugin an attractive choice for musicians and sound designers alike.

7. ACKNOWLEDGMENTS

Michele Ducceschi received funding from the European Research Council (ERC), under the European Union's Horizon 2020 research and innovation programme, grant agreement No. 950084-NEMUS.

8. REFERENCES

[1] John P Boyd, *Chebyshev and Fourier spectral methods*, Dover, Mineola, New York, USA, 2001.

[2] L. Meirovitch, *Fundamentals of vibrations*, Waveland Press, Long Grove, Illinois, USA, 2010.

[3] M. Ducceschi and C.J. Webb, "Plate reverberation: Towards the development of a real-time plug-in for the working musician," in *Proc of Int Conf Acoust (ICA 2016)*, Buenos Aires, Argentina, September 2016.

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

[5] J. Cieslinski, "On the exact discretization of the classical harmonic oscillator equation," *J Differ Equ Appl*, vol. 17, no. 11, pp. 1673–1694, 2009.

[6] J.M. Adrien, "The missing link: Modal synthesis," in *Representations of Musical Signals*, p. 269–298. MIT Press, Cambridge, MA, USA, 1991.

[7] G. Eckel, "Sound synthesis by physical modelling with modalys," *Proc. ISMA'95*, pp. 478–482, 1995.

[8] R. Rabenstein and L. Trautmann, "Digital sound synthesis of string instruments with the functional transformation method," *Signal Proc*, vol. 83, no. 8, pp. 1673–1688, 2003.

[9] S. Schlecht, J. Parker, M. Schäfer, and R. Rabenstein, "Physical modeling using recurrent neural networks with fast convolutional layers," in *Proc Digital Audio Effects (DAFx-22)*, 2022, pp. 138–145.

[10] J. McQuillan and M. van Walstijn, "Modal spring reverb based on discretisation of the thin helical spring model," in *Proc Digital Audio Effects (DAFx-21)*, 2021, pp. 191–198.

[11] M. Ducceschi and C. Touzé, "Modal approach for nonlinear vibrations of damped impacted plates: application to sound synthesis of gongs and cymbals," *J Sound Vib*, vol. 334, pp. 313–331, 2015.

[12] M. van Walstijn, J. Bridges, and S. Mehes, "A real-time synthesis oriented tanpura model," in *Proc Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, September 2016.

[13] B. Bank and L. Sujbert, "Generation of longitudinal vibrations in piano strings: From physics to sound synthesis," *J Acoust Soc Am*, vol. 117, no. 4, pp. 2268–2278, 2005.

[14] 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.

[15] M. Van Walstijn and M. Sandor, "An explorative string-bridge-plate model with tunable parameters," in *Proc Digital Audio Effects (DAFx-17)*, Edinburgh, UK, September 2017.

[16] S. Bilbao, M. Ducceschi, and C. Webb, "Large-scale real-time modular physical modeling sound synthesis," in *Proc Digital Audio Effects (DAFx-19)*, Birmingham, UK, September 2019.

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

[18] J. Shen, J. Xu, and J. Yang, "The scalar auxiliary variable (sav) approach for gradient flows," *J Comput Phys*, vol. 353, pp. 407–416, 2018.

[19] S. Bilbao, M. Ducceschi, and F. Zama, "Explicit exactly energy-conserving methods for hamiltonian systems," *J Comput Phys*, vol. 472, pp. 111697, 2023.

[20] Z. Liu and X. Li, "The exponential scalar auxiliary variable (e-sav) approach for phase field models and its explicit computing," *SIAM J Sci Comput*, vol. 42, no. 3, pp. B630–B655, 2020.

[21] J. Sherman and W. J. Morrison, "Adjustment of an inverse matrix corresponding to a change in one element of a given matrix," *Ann Math Stat*, vol. 21, pp. 124–127, 1950.

[22] E. Hairer, C. Lubich, and G. Wanner, "Geometric numerical integration illustrated by the Störmer–Verlet method," *Acta Numerica*, vol. 12, pp. 399–450, 2003.

View publication stats