Real-time simulation of the struck piano string with geometrically exact nonlinearity via a scalar quadratic energy method

Michele Ducceschi*, Stefan Bilbao† and Craig J. Webb‡
*Dipartimento di Ingegneria Industriale, University of Bologna, Italy
†Acoustics and Audio Group, University of Edinburgh, UK
‡ Physical Audio Ltd, London, UK

Summary. This work addresses the problem of the struck piano string. This system is highly nonlinear, and a sound representation of the energy balance is therefore necessary in any time stepping routine used in simulation, in order to preserve stability. Many algorithms have been developed in previous works. Among them, some present fully-implicit discretisations, which are only approachable using iterative root finders such as Newton-Raphson. Others are linearly-implicit, but not quite suited for real-time rendering. Here, a novel approach is presented, based on the idea of energy quadratisation. It will be shown that, when the nonlinearities are consolidated into a scalar auxiliary state function, the time stepping scheme presents a fast inversion formula. A C++ implementation of the proposed scheme yields indeed compute times below real-time, for typical strings.

Introduction

The dynamics of musical strings is often assumed to be linear. However, many perceptual phenomena cannot be explained by linear theory alone, and inclusion of nonlinearities results therefore necessary [1]. This is certainly true for the piano, where the appearance of "phantom partials" in the output spectra is often explained in terms of the nonlinear coupling between the transverse and the longitudinal waves. Furthermore, the intermittent hammer-string interaction cannot be modelled linearly [2]. At the simulation level, a sound representation of the energy balance of the hammer-string system is necessary to preserve stability of the underlying time stepping scheme, and this can be accomplished via fully-implicit discretisations [3, 4]. The finite element method, in particular, is preeminent here, see e.g. [5, 6], where the conservation of a non-negative numerical energy allows to derive a stability condition involving the mesh size and the time step. Since these methods are fully-implicit, they require the solution of a large nonlinear algebraic system at each time step, for which existence and uniqueness of the solution must be proven, and which are only approachable using iterative routines such as Newton-Raphson. Further complications arise in the choice of the tolerance thresholds and maximum number of iterations of the routine [7, 8].

In recent years, a new class of schemes has emerged, based on the idea of "quadratisation": when the nonlinear potential energy is non-negative, it can be written as the square of an auxiliary function. This leads to schemes for which the update is computable as the solution of a linear system, thus sidestepping the machinery of iterative root finders. Besides the obvious computational advantage, existence and uniqueness of the computed solution are proven by inspection of the update matrix alone. These methods have appeared in various guises, including applications in Port-Hamiltonian systems [9, 10, 4, 11], nonlinear parabolic phase-field models [12, 13, 14], and others. This latter class of methods is often referred to as "Invariant Energy Quadratisation" (IEQ) method, in that the auxiliary state function is treated as an independent grid function to be solved for. This method has been applied successfully to the geometrically exact nonlinear string [15, 16]. In this work, a different kind of quadratisation is proposed, where the auxiliary state function is not treated as a distributed state variable, but as a scalar one. This idea shares some similarities with the "Scalar Auxiliary Variable" (SAV) approach [17], but it also exploits the form of the update matrix to yield an extremely efficient algorithm. The application of this method to Hamiltonian system is the subject of a companion paper [18]; here, the example of the struck string is considered in detail. First, the problem is semi-discretised in space, resulting in a system of nonlinearly coupled ordinary differential equations in time. Then, all the nonlinear components (i.e. the geometric nonlinearity of the string, and the hammer-string nonlinear potential) are consolidated into a single, scalar state function. Then, it shown that the update matrix is in the form of a rank-1 perturbation of a diagonal matrix, and thus invertible efficiently using the Sherman-Morrison formula [19]. A computational test is then performed, showing that a sound implementation of this method in C++ yields speedups of the order of 10^2, compared to Matlab implementations of previously available methods. In particular, compute times for typical piano strings are below real-time. The introduction of a shift constant in the nonlinear potential energy is also illustrated, aiding convergence of the quadratised numerical scheme.

A Continuous Struck String Model

Piano strings are known to vibrate in a nonlinear regime as a consequence of the large strains originating during motion [1, 20]. The striking mechanism is also nonlinear, due to the intermittent contact between the string and the hammer

[2, 21]. A suitable mathematical model incorporating these effects may be given as follows:

$$(\rho A\partial_t^2 - \mathcal{L}_u)u(x,t) = \partial_x\left(\frac{\partial\phi_s}{\partial(\partial_x u)}\right) + \delta(x-x_c)\frac{d\phi_c}{d\eta}, \tag{1a}$$

$$(\rho A\partial_t^2 - \mathcal{L}_v)v(x,t) = \partial_x\left(\frac{\partial\phi_s}{\partial(\partial_x v)}\right), \tag{1b}$$

$$M_h\frac{d^2U(t)}{dt^2} = -\frac{d\phi_c}{d\eta}, \tag{1c}$$

$$\eta(t) = U(t) - \int_0^L \delta(x-x_c)u(x,t)\,\mathrm{d}x. \tag{1d}$$

Above, constants appear as: ρ, the string's volume density; A the area of the string's cross section; M_h, the mass of the hammer. The notation ∂_s^p denotes the p-th partial derivative with respect to s. $u = u(x,t) : [0,L]\times\mathbb{R}_0^+ \to \mathbb{R}$ is the transverse displacement of the string, and $v = v(x,t) : [0,L]\times\mathbb{R}_0^+ \to \mathbb{R}$ is the longitudinal displacement, where L is the string length; $\mathcal{L}_u$, $\mathcal{L}_v$ are linear differential operators (to be specified shortly) including the tension, stiffness and loss components for the two displacements; $\phi_s : \phi_s(\partial_x u, \partial_x v) : \mathbb{R}\times\mathbb{R} \to \mathbb{R}_0^+$ is the nonlinear potential density due to the string's large stretching; $\phi_c : \phi_c(\eta) : \mathbb{R} \to \mathbb{R}_0^+$ is the collision potential modelling the hammer-string interaction; $U = U(t) : \mathbb{R}_0^+ \to \mathbb{R}$ is the hammer's displacement; $\eta = \eta(t) : \mathbb{R}_0^+ \to \mathbb{R}$ defines the compression of the hammer felt, expressed as the difference between the hammer and string displacements at the contact point x_c.
The linear operators $\mathcal{L}_u$, $\mathcal{L}_v$ are given as:

$$\mathcal{L}_u = T_0\partial_x^2 - EI\partial_x^4 - 2\rho A(\sigma_0 - \sigma_1\partial_x^2)\partial_t; \quad \mathcal{L}_v = T_0\partial_x^2 - 2\rho A\sigma_l\partial_t. \tag{2}$$

$\mathcal{L}_u$ includes tension and stiffness terms, the latter borrowed from the Euler-Bernoulli beam theory (this is largely sufficient for musical strings, see e.g. [22, 23]). T_0 is the applied tension, E is Young's modulus, I is the area moment of inertia. Losses in the transverse direction are modelled via the terms proportional to the coefficients $(\sigma_0,\, \sigma_1) \geq 0$, yielding a quadratic loss profile in the frequency domain. Inclusion of more refined loss profiles, such as the one proposed by Cuesta and Vallette [24], is possible, but this will be neglected here for simplicity. $\mathcal{L}_v$ includes a tension term, and a simple loss model proportional to $\sigma_l \geq 0$.
The forms of the potentials are given here as:

$$\phi_s(\partial_x u, \partial_x v) = \frac{EA - T_0}{2}\left(\sqrt{(1+\partial_x v)^2 + (\partial_x u)^2} - 1\right)^2, \quad \phi_c(\eta) = \frac{B}{\alpha+1}[\eta]_+^{\alpha+1}. \tag{3}$$

Here, B is the stiffness coefficient of the hammer felt, $\alpha \geq 1$ is an exponent: these values can be determined experimentally [25]. The notation $[\cdot]_+$ stands for "positive part", that is $[\eta]_+ \triangleq 0.5(\eta + |\eta|)$. Note that $\phi_c = 0$ whenever $\eta \leq 0$, that is, the collision model is one-sided, and note that this model is consistent only when the hammer strikes from below. Note as well that $\phi_s \geq 0$ when $EA \geq T_0$, a condition that is always verified for musical strings. In view of the numerical application shown below, it is convenient to lump the nonlinear potentials into one single scalar function ϕ, as

$$\phi(\partial_x u, \partial_x v, \eta) = \int_0^L \phi_s(\partial_x u, \partial_x v)\,\mathrm{d}x + \phi_c(\eta) + \frac{p_0}{2}, \tag{4}$$

where $p_0 \geq 0$ is a constant shifting the zero-point of the potential, but not affecting the equations of motion. This constant will prove useful in the numerical scheme presented below.
Equation (1c) must be completed by two initial conditions, setting the hammer in motion. These can be given as $U_0 \triangleq U(t=0)$ and $V_0 \triangleq dU/dt(t=0)$. Furthermore, the string's boundary conditions are given as: $u(0,t) = u(L,t) = \partial_x^2 u(0,t) = \partial_x^2 u(L,t) = v(0,t) = v(L,t) = 0$. These define an energy conserving set of boundary conditions; in particular, the string is simply-supported transversely and fixed longitudinally. Under such conditions, system (1) is energy-passive. A proof may be obtained by taking an l_2 inner product (i.e. an integral over the domain) of (1a) with $\partial_t u$, of (1b) with $\partial_t v$, and a product of (1c) with dU/dt, and summing the three equations. After suitable integrations by parts [16], one gets the following energy balance

$$\frac{dH}{dt} = -2\rho A\left(\int_0^L \left(\sigma_0(\partial_t u)^2 + \sigma_l(\partial_t v)^2 + \sigma_1(\partial_t\partial_x u)^2\right)\,\mathrm{d}x\right) \leq 0, \tag{5}$$

where the energy is

$$H(t) = \int_0^L \left(\frac{\rho A}{2}\left((\partial_t u)^2 + (\partial_t v)^2\right) + \frac{T_0}{2}\left((\partial_x u)^2 + (\partial_x v)^2\right) + \frac{EI}{2}(\partial_x^2 u)^2\right)\,\mathrm{d}x + \frac{M_h(dU/dt)^2}{2} + \phi \geq 0 \tag{6}$$

Equation (5) represents the energy balance of the hammer-string, encoding passivity (i.e. the total energy is positive, and decays over time).

Semi-discretisation

System (1) may be semi-discretised in a number of ways, yielding a system of coupled nonlinear ordinary differential equations in time. Here, a finite difference approach will be adopted, but the numerical method described below applies equally to any other suitable semi-discrete problem, obtained for instance via Galerkin-type methods (modal methods, finite elements, etc.). For the finite difference method, the continuous domain of length L is divided into segments of length h, the grid spacing. Let the total number of subintervals be M, that is, the grid is composed of $M+1$ grid points, including the boundary points. Then, the continuous displacements $u(x,t)$, $v(x,t)$ are approximated by grid functions $\mathbf{u}(t)$, $\mathbf{v}(t)$, which may be taken as column vectors of length $M-1$: here, the end points need not be stored or updated, since the boundary conditions are of fixed type. Difference matrices are now introduced, starting from the definition of the backward difference matrix acting on the column vector $\mathbf{u}$:

$$\mathbf{D}^- \mathbf{u} = \frac{1}{h}([\mathbf{u}^\intercal, 0] - [0, \mathbf{u}^\intercal]). \tag{7}$$

Note that this is a rectangular matrix of size $M \times M-1$. From this, one may define the forward difference matrix, the Laplace operator and the biharmonic operator, as

$$\mathbf{D}^+ = -(\mathbf{D}^-)^\intercal, \quad \mathbf{D}^2 = \mathbf{D}^+\mathbf{D}^-, \quad \mathbf{D}^4 = (\mathbf{D}^2)^2, \tag{8}$$

and note that these matrices are consistent with the prescribed boundary conditions in the transverse and longitudinal directions. Note as well that the Laplace and biharmonic operators are of size $M-1 \times M-1$. Owing to these definitions, semi-discretisation of the linear operators $\mathcal{L}_u$, $\mathcal{L}_v$ in (2) is obtained as

$$\mathfrak{l}_u = T_0\mathbf{D}^2 - EI\mathbf{D}^4 - 2\rho A(\sigma_0 - \sigma_1\mathbf{D}^2)\frac{d}{dt}; \quad \mathfrak{l}_v = T_0\mathbf{D}^2 - 2\rho A\sigma_l\frac{d}{dt}. \tag{9}$$

The Dirac delta function is here approximated using a linear Lagrange interpolant, that is, via the column vector $\mathbf{J}$ of length $M-1$. This is an all-zero vector, except at the grid points $m_c \triangleq \text{floor}(x_c/h)$ and m_c+1, for which

$$J_{m_c} = (1-\alpha)/h, \; J_{m_c+1} = \alpha/h, \quad \text{where } \alpha = x_c/h - m_c. \tag{10}$$

Using this definition, a semi-discrete version of (1d) is obtained as:

$$\eta(t) = U(t) - h\,\mathbf{J}^\intercal\,\mathbf{u}(t). \tag{11}$$

Note that the letter η is used here in a different fashion than in (1d) (in this section, the semi-discrete definition (11) is adopted). It is now convenient to define two interleaved grid functions, $\mathbf{q}$ and $\mathbf{r}$, which are column vectors of length M defined as:

$$\mathbf{q} = \mathbf{D}^-\mathbf{u}, \quad \mathbf{r} = \mathbf{D}^-\mathbf{v}. \tag{12}$$

Owing to these, a semi-discrete version of the nonlinear potential ϕ in (4) is obtained as:

$$\Phi = h\sum_{i=1}^{M}\Phi_s^i(q_i, r_i) + \Phi_c(\eta) + \frac{p_0}{2}; \text{ where } \Phi_s^i = \frac{EA - T_0}{2}\left(\sqrt{(1+r_i)^2 + q_i^2} - 1\right)^2, \; \Phi_c = \frac{B}{\alpha+1}[\eta]_+^{\alpha+1}. \tag{13}$$

Note that here $\Phi = \Phi(\mathbf{q}, \mathbf{r}, \eta) : \mathbb{R}^M \times \mathbb{R}^M \times \mathbb{R} \to \mathbb{R}_0^+$, and it will be necessary to compute partial derivatives according to $\mathbf{q}, \mathbf{r}, \eta$. These are denoted as

$$(\nabla_{\mathbf{q}}\Phi)_i \triangleq \frac{\partial\Phi_s^i}{\partial q_i}, \quad (\nabla_{\mathbf{r}}\Phi)_i \triangleq \frac{\partial\Phi_s^i}{\partial r_i}, \quad (\nabla_\eta\Phi) \triangleq \frac{d\Phi_c}{d\eta}, \quad i = 1, ..., M. \tag{14}$$

The chain rule gives

$$\frac{d\Phi}{dt} = (\nabla_{\mathbf{q}}\Phi)^\intercal\frac{d\mathbf{q}}{dt} + (\nabla_{\mathbf{r}}\Phi)^\intercal\frac{d\mathbf{r}}{dt} + (\nabla_\eta\Phi)\frac{d\eta}{dt} = \underbrace{[-\left(\mathbf{D}^+\nabla_{\mathbf{q}}\Phi\right)^\intercal - h\nabla_\eta\Phi\mathbf{J}^\intercal, -\left(\mathbf{D}^+\nabla_{\mathbf{r}}\Phi\right)^\intercal, \nabla_\eta\Phi]}_{\mathbf{g}_\Phi^\intercal}\frac{d\mathbf{w}}{dt}, \tag{15}$$

where $\mathbf{w} \triangleq [\mathbf{u}^\intercal, \mathbf{v}^\intercal, U]^\intercal$ is the state vector, lumping the longitudinal and transverse string grid functions, and the hammer's displacement. Note that the second equality in (15) is obtained using the transposition property of the forward and backward difference matrices in (8), and the definition of η in (11). Owing to these definitions, the semi-discrete equations can be written in compact form, as:

$$\mathbf{M}\frac{d^2\mathbf{w}(t)}{dt^2} + \mathbf{C}\frac{d\mathbf{w}(t)}{dt} + \mathbf{K}\mathbf{w}(t) = -\mathbf{S}\,\mathbf{g}_\Phi, \tag{16}$$

where $\mathbf{M} \triangleq \text{diag}([\rho A\mathbf{I}, \rho A\mathbf{I}, M_h])$, $\mathbf{K} \triangleq \text{diag}([-T_0\mathbf{D}^2+EI\mathbf{D}^4, -T_0\mathbf{D}^2, 0])$, $\mathbf{C} \triangleq \text{diag}([2\rho A(\sigma_0\mathbf{I}-\sigma_1\mathbf{D}^2), 2\rho A\sigma_l\mathbf{I}, 0])$, $\mathbf{S} \triangleq \text{diag}([\mathbf{I}/h, \mathbf{I}/h, 1])$. Here, the operator "diag" produces a block-diagonal matrix, with blocks given as components separated by commas. The matrix $\mathbf{I}$ is here the $M-1 \times M-1$ identity matrix.
In (16), the left-hand side includes all the linear components (the mass, stiffness and loss matrices), and the nonlinearities are now lumped in the vector $\mathbf{g}_\Phi$ as defined in (15). From here, it is now easy to see that system (16) is energy-passive: it is sufficient to multiply both sides of the equation on the left by $\left(\mathbf{S}^{-1}\frac{d\mathbf{w}}{dt}\right)^\intercal$. This gives

$$\frac{d\mathfrak{h}}{dt} = -\frac{d\mathbf{w}^\intercal}{dt}\tilde{\mathbf{C}}\frac{d\mathbf{w}}{dt} \leq 0, \text{ with } \mathfrak{h} = \frac{1}{2}\frac{d\mathbf{w}^\intercal}{dt}\tilde{\mathbf{M}}\frac{d\mathbf{w}}{dt} + \frac{1}{2}\mathbf{w}^\intercal\tilde{\mathbf{K}}\mathbf{w} + \Phi \geq 0, \tag{17}$$

where the "tilde" notation indicates the rescaled matrices $\tilde{\mathbf{M}} = \mathbf{S}^{-1}\mathbf{M}$, $\tilde{\mathbf{K}} = \mathbf{S}^{-1}\mathbf{K}$, $\tilde{\mathbf{C}} = \mathbf{S}^{-1}\mathbf{C}$ (the scaling by the factor h is needed here to restore units of Joules in the transverse and longitudinal string components). Note that the "tilde" matrices are all non-negative, hence the inequalities in (17) ensue.

Quadratisation

The expression for the semi-discrete energy $\mathfrak{h}$ in (17) includes a linear part, given by the sum of quadratic forms corresponding to the kinetic and linear potential energy components, plus the nonlinear potential Φ. It results therefore convenient to "quadratise" the nonlinear potential via the transformation

$$\Psi = \sqrt{2\Phi}, \tag{18}$$

for the function $\Psi = \Psi(\mathbf{q}, \mathbf{r}, \eta) : \mathbb{R}^M \times \mathbb{R}^M \times \mathbb{R} \to \mathbb{R}_0^+$. This transformation is always well-defined since Φ is non-negative. This transformation yields a semi-discrete system entirely equivalent to (16), but expressed in terms of Ψ as:

$$\mathbf{M}\frac{d^2\mathbf{w}(t)}{dt^2} + \mathbf{C}\frac{d\mathbf{w}(t)}{dt} + \mathbf{K}\mathbf{w}(t) = -\Psi(t)\mathbf{S}\,\mathbf{g}_\Psi, \tag{19}$$

where

$$\mathbf{g}_\Psi = [-\left(\mathbf{D}^+\nabla_\mathbf{q}\Psi\right)^\intercal - h\nabla_\eta\Psi\mathbf{J}^\intercal, -\left(\mathbf{D}^+\nabla_\mathbf{r}\Psi\right)^\intercal, \nabla_\eta\Psi], \tag{20}$$

and note that the partial derivatives of Ψ are easily obtained from (14), since e.g. $\nabla_\mathbf{q}\Psi = \nabla_\mathbf{q}\Phi/\Psi$. Division by Ψ in the expression of the partial derivatives may be ill defined when the shift constant p_0 in (13) is set to zero, and it will therefore be necessary to shift the potential energy by a finite amount. Application of the chain rule gives

$$\frac{d\Psi}{dt} = \mathbf{g}_\Psi^\intercal\frac{d\mathbf{w}}{dt}. \tag{21}$$

The energy balance now reads:

$$\frac{d\mathfrak{h}}{dt} = -\frac{d\mathbf{w}^\intercal}{dt}\tilde{\mathbf{C}}\frac{d\mathbf{w}}{dt} \leq 0, \text{ where } \mathfrak{h} = \frac{1}{2}\frac{d\mathbf{w}^\intercal}{dt}\tilde{\mathbf{M}}\frac{d\mathbf{w}}{dt} + \frac{1}{2}\mathbf{w}^\intercal\tilde{\mathbf{K}}\mathbf{w} + \frac{\Psi^2}{2} \geq 0, \tag{22}$$

and the expression for the energy only contains quadratic terms. Note that the nonlinear energy components are now lumped in the single scalar function Ψ. The "quadratised" expression for the nonlinear energy is convenient since it can be exploited in a time-stepping integrator yielding a non-negative nonlinear numerical energy, as detailed below. Note that this type of quadratisation is fundamentally different from the one proposed in [15, 16], in that here the auxiliary state function is a scalar, and not a distributed quantity. Applications of this method to Hamiltonian systems are presented in the companion paper [18].

Time Stepping Procedure

(19) defines a system of nonlinearly coupled ordinary differential equations in time. Integration is now performed using finite differences. To that end, the continuous state vector $\mathbf{w}(t)$ is approximated at the time $t_n = kn$ by a vector time series $\mathbf{w}^n$, where $n \in \mathbb{N}$ is the time index, and $k = 1/f_s$ is the time step, defined as the multiplicative inverse of the sample rate f_s. The basic operators in discrete time are the identity and shift operators, defined as:

$$1\mathbf{w}^n = \mathbf{w}^n, \quad e_+\mathbf{w}^n = \mathbf{w}^{n+1}, \quad e_-\mathbf{w}^n = \mathbf{w}^{n-1}. \tag{23}$$

From these, the time difference operators can be defined as:

$$\delta_+ = \frac{e_+ - 1}{k}, \ \delta_- = \frac{1 - e_-}{k}, \ \delta_\cdot = \frac{e_+ - e_-}{2k}. \tag{24}$$

These are the forward, backward and centred operators respectively. The second-difference operator is obtained by combining the the operators above:

$$\delta_2 = \delta_+\delta_- \,. \tag{25}$$

Finally, averaging operators can be written as:

$$\mu_+ = \frac{e_+ + 1}{2},\ \mu_- = \frac{1 + e_-}{2},\ \mu_\cdot = \frac{e_+ + e_-}{2}. \tag{26}$$

The auxiliary state function is discretised on an interleaved time grid, so that $\Psi(t) \to \Psi^{n-1/2}$. The definitions of the difference operators are formally unchanged when applied to an interleaved time series, so that e.g. $\delta_+ \Psi^{n-1/2} = (\Psi^{n+1/2} - \Psi^{n-1/2})/k$, etc. Note that $\Psi^{n-1/2}$ will be treated here as an *independent* scalar time series, to be updated at each time step. To that end, it is necessary to discretise (19) along with the time derivative of Ψ in (21). One possible such discretisation is given here as

$$\mathbf{M}\,\delta_2 \mathbf{w}^n + \mathbf{C}\,\delta_- \mathbf{w}^n + \mathbf{K}\,\mathbf{w}^n = -\mu_+ \Psi^{n-1/2}\,\mathbf{S}\,\mathbf{g}_\Psi^n, \quad \delta_+ \Psi^{n-1/2} = (\mathbf{g}_\Psi^n)^\intercal \delta_\cdot \mathbf{w}^n. \tag{27}$$

This defines a three-step scheme, in which the updates $\mathbf{w}^{n+1}$, $\Psi^{n+1/2}$ are obtained from the previous state values $\mathbf{w}^n, \mathbf{w}^{n-1}$, $\Psi^{n-1/2}$. Note as well that $\mathbf{g}_\Psi^n$ is known, and computed from (20) at the time step n, using the analytic expressions for the derivatives of the continuous function Ψ.
System (27) is completed by initial conditions on $\mathbf{w}$ and Ψ. At the initial time $n = 0$, it is assumed that the string is at rest, with the hammer moving according the initial conditions given above. The time series Ψ must be initialised accordingly, using definition (18) and the expression of Φ from (13). Thus

$$\mathbf{w}^0 = [\mathbf{0}^\intercal, \mathbf{0}^\intercal, U_0]^\intercal, \quad \mathbf{w}^1 = [\mathbf{0}^\intercal, \mathbf{0}^\intercal, kV_0 + U_0]^\intercal, \quad \Psi^{1/2} = \sqrt{p_0}. \tag{28}$$

Scheme (27) is energy-conserving. A proof is obtained by multiplying on left by $(\mathbf{S}^{-1}\delta_\cdot \mathbf{w}^n)^\intercal$, and using various identities (not shown here for brevity; see e.g. [20]). One gets

$$\delta_+ \mathfrak{h}^{n-1/2} = -(\delta_\cdot \mathbf{w}^n)^\intercal\, \tilde{\mathbf{C}}\, \delta_\cdot \mathbf{w}^n \leq 0, \tag{29}$$

where

$$\mathfrak{h}^{n-1/2} = \frac{1}{2}(\delta_+ \mathbf{w}^n)^\intercal \left(\tilde{\mathbf{M}} - \frac{k}{2}\tilde{\mathbf{C}}\right) \delta_+ \mathbf{w}^n + \frac{1}{2}(\mathbf{w}^n)^\intercal\, \tilde{\mathbf{K}}\, e_- \mathbf{w}^n + \frac{(\Psi^{n-1/2})^2}{2}. \tag{30}$$

This expresses the fully-discrete energy balance. It is worth noting that the expression for the fully discrete energy (30) is not positive semi-definite in all cases (as opposed to the semi-discrete energy in (22), which is always non-negative). However, conditions on non-negativity can be obtained by inspection of the linear part alone, since the nonlinear energy is here clearly non-negative, and expressed via the square of Ψ. It is possible to show (see e.g. [20, 16]) that, when the loss coefficients are small (as is the case for all musical strings), a sufficient condition for the non-negativity of the total discrete energy is

$$h \geq \sqrt{E/\rho}\, k. \tag{31}$$

When this condition is enforced, the discrete energy is non-negative, and hence stability of the time stepping scheme follows. Note that (31) is the CFL condition associated with the longitudinal motion, and it is expressed as a lower bound on h: this is much larger than the natural bound for the transverse waves (which is of the order of $\sqrt{T_0/\rho A}\, k$ for thin strings). Thus, enforcing condition (31) has consequences in the choice of the sample rate f_s, which has to be chosen much larger than typical audio rates (such as e.g. 44.1 kHz) to get enough resolution within the audio band.
Note as well that in (27) the operator δ_- was used to approximate the time derivative of the losses. This formally produces a first-order convergent scheme in time. However, it also allows for a fast update, as detailed below. Since losses are small, and since the time step k has to be chosen in the $10^{-5} - 10^{-6}$ range, this approximation does not introduce significant errors compared to a second-order accurate discretisation of the same derivative.

Scheme Update

Scheme (27) yields an efficient update equation. To show this, first use the identity $\mu_+ = (k/2)\delta_+ + 1$ in order to express the right-hand side of the first equation in (27) in terms of $\delta_\cdot \mathbf{w}^n$ from the second equation. Then, one rearranges the terms to get

$$\left(\frac{\mathbf{M}}{k^2} + \frac{1}{4}\mathbf{S}\,\mathbf{g}_\Psi^n\,(\mathbf{g}_\Psi^n)^\intercal\right)\mathbf{w}^{n+1} = \left(\frac{2\mathbf{M}}{k^2} - \frac{\mathbf{C}}{k} - \mathbf{K}\right)\mathbf{w}^n + \left(-\frac{\mathbf{M}}{k^2} + \frac{\mathbf{C}}{k} + \frac{1}{4}\mathbf{S}\,\mathbf{g}_\Psi^n\,(\mathbf{g}_\Psi^n)^\intercal\right)\mathbf{w}^{n-1} - \Psi^{n-1/2}\mathbf{S}\,\mathbf{g}_\Psi^n, \tag{32}$$

and note that the update matrix is now composed of the diagonal matrix $\mathbf{M}/k^2$ plus the rank-1 perturbation $\frac{1}{4}\mathbf{S}\,\mathbf{g}_\Psi^n\,(\mathbf{g}_\Psi^n)^\intercal$. This is invertible in $O(M)$ operations using the Sherman-Morrison formula [19, 18]. Once $\mathbf{w}^{n+1}$ is known, one may compute $\Psi^{n+1/2}$ using the second equation in (27).

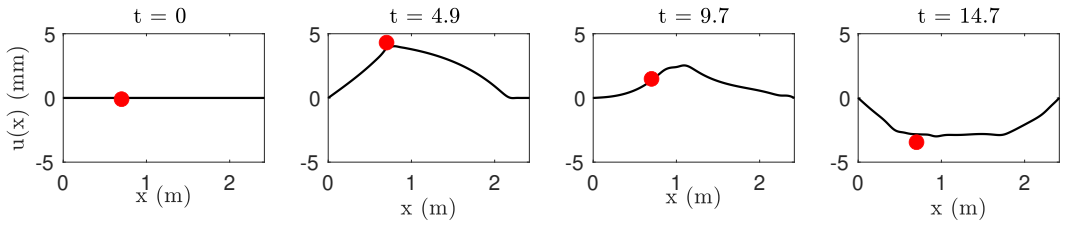

Figure 1: Snapshots of the hammer-string interaction. Here, the hammer and the string parameters are taken from [25], for the C2 string. The hammer initial velocity is 2 m/s. The sample rate used is $12 \cdot 48000$ Hz, resulting in a state size $2M - 1 = 547$ components. The times indicated are in ms.

Numerical Experiments

Scheme (27) is now tested in a number of numerical experiments. Figure 1 shows snapshots of the hammer-string dynamics, for the C2 piano string (the string and hammer parameters are taken from [25]). As expected, the hammer strikes the string intermittently. Here, the string's largest displacement is of the order of 4 mm, large enough to entail nonlinear effects.

The energetic behaviour of scheme (27) is checked in Figure 2, where the same C2 string is simulated in the absence of losses (that is, the matrix $\mathbf{C}$ is the zero matrix here). The discrete energy (30) is indeed conserved, with the error in the range of 10^{-14}. Note that here the kinetic energy includes that of the string and the hammer; the linear potential energy includes the string's alone, and the nonlinear energy includes the contributions of both the string's large displacement and the hammer-string collision. Note that the shift constant p_0 was set to a finite amount, to avoid division by zero in the computation of the derivatives of Ψ.

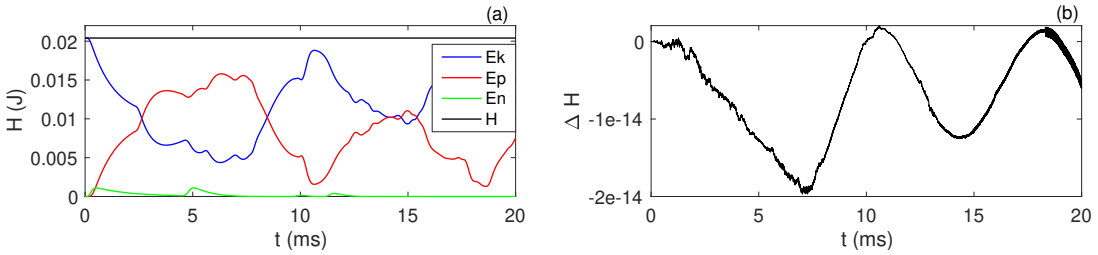

Figure 2: (a) Energy components and (b) energy error for the same case as Figure 1. In (a), E_k is the kinetic energy (comprising the string and the hammer), E_p is the linear potential energy of the string, E_n is the nonlinear potential energy including the string's nonlinear energy and the hammer-string collision potential, H is the total energy. In (b), ΔH is defined as $1 - \mathfrak{h}^{n-1/2}/\mathfrak{h}^{1/2}$, where the expression for the discrete energy $\mathfrak{h}$ is as per (30). In this simulation, the shift constant p_0 is set to 10^{-15}.

A quick convergence test is presented in Figure 3. Here, the time domain solutions at one output point are plotted, for a number of sample rate values. As mentioned above, significant oversampling factors are needed, compared to audio rate, since the number of grid points is now set according to bound (31), yielding a large grid spacing compared to typical wavelengths in the transverse direction. For high enough sample rates (OF $\gtrsim 12$), the computed solutions are very close, but discrepancies are evident for low-sampled waveforms. Various tests were run (not shown here), indicating that an oversampling factor of at least 12 is usually required for good resolution within the audio band. Some audio files are available on the companion webpage[1].

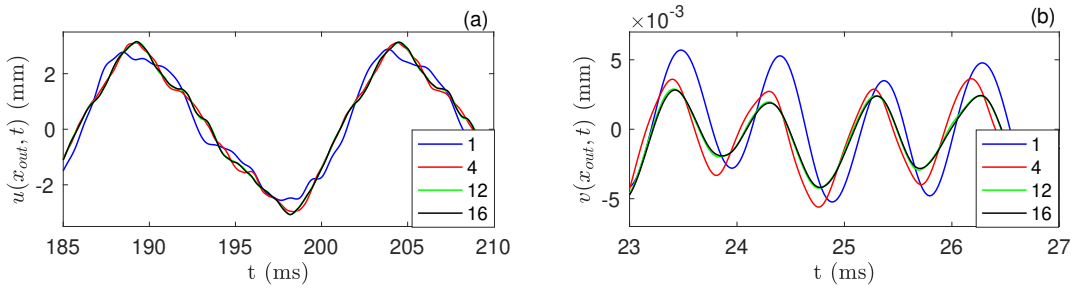

Figure 3: Convergence test under various oversampling factors for a base sample rate $f_{s0} = 48$ kHz, indicated in the legend. (a): transverse displacement. (b): longitudinal displacement. Here $x_{out} = 0.32L$, and the string's parameters are the same as in Figure 1.

Figure 4 presents the output spectra of the longitudinal displacement under three different hammer initial velocities. Note the change in brightness as the velocity increases, as well as the appearance of phantom partials.

[1] https://mdphys.org/ENOC_2022.html

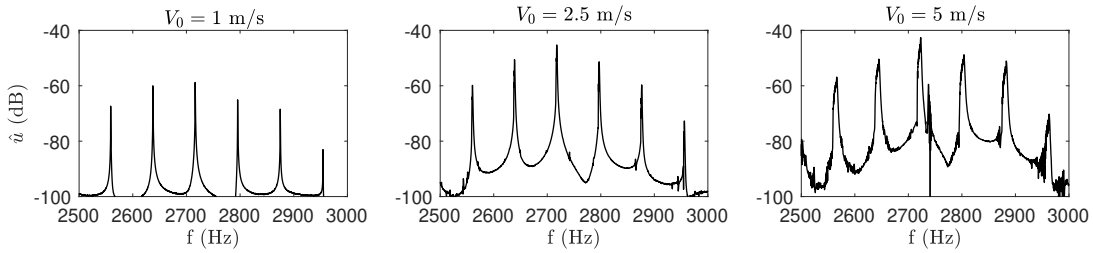

Figure 4: Output spectra for the D2 string, with initial hammer velocities as indicated. Output is extracted at $x_{out} = 0.32L$

Table 1 presents a computational testing comparing various schemes, and implementations. Two schemes, called `ICA_IT` and `ICA_NIT`, are taken from [15]. `ICA_IT` is a finite difference iterative scheme, obtained via a fully-implicit conservative discretisation of the gradient in a fashion similar to that used in many works, see e.g. [3, 4, 5, 26]; `ICA_NIT` is a non-iterative finite difference scheme obtained via quadratisation, in a fashion similar to the scheme presented here, but where the nonlinear auxiliary variable Ψ is approximated via a *grid function*, and not as a single scalar variable as here (that is, it is not possible to update `ICA_NIT` using the fast Sherman-Morrison formula, though the scheme's update is in a form of a linear system). `LU_NIT`, presented in [16], is also a non-iterative scheme, where the transverse waves are simulated using finite differences, and the longitudinal waves are approximated using a modal approach. The system is again in "quadratised" form, and a block LUD decomposition is adopted to compute the update. All these schemes were coded in Matlab. Finally, `SM_MTL` and `SM_C++` present the compute times for the fast Sherman-Morrison formula

OF	M	M_{long}	`ICA_IT`	`ICA_NIT`	`LU_NIT`	`SM_MTL`	`SM_C++`
1	10	2	14.6	8.73	7.59	14.1	0.008
2	21	4	24.7	19.8	16.1	16.4	0.029
4	42	7	53.4	41.4	22.5	19.1	0.108
8	85	14	145	102	87.0	26.6	0.362
12	128	21	267	191	173	37.8	0.756
16	170	28	453	324	279	58.3	1.15

Table 1: Compute time / real time ratios, using various schemes and implementations, for the D3 piano string from [25]. Here, `ICA_IT` and `ICA_NIT` are, respectively, the finite difference iterative and non-iterative schemes presented in [15]; `LU_NIT` is the non-iterative mixed finite difference-modal method described in [16] (where the transverse displacement is approximated using finite differences, and the longitudinal displacement using a modal approach); `SM_MTL` is the Sherman-Morrison implementation presented here, using Matlab; `SM_C++` is the same scheme implemented in C++. In the table, OF indicates the oversampling factor for a base sample rate $f_{s0} = 48$ kHz. M is the number of grid points; M_{long} is the number of longitudinal modes used in `LU_NIT`. In this test, for simplicity, the hammer was not considered, and the string was initialised in its first mode of vibration in the transverse direction, i.e. $u_0(x) = 0.01 \sin(\pi x/L)$. The size of the state vector, for all schemes, is then $2M - 2$, with the exception of `LU_NIT` for which the size is $M - 1 + M_{long}$. Below-real-time simulations are highlighted. All simulations were run on a 2016 Macbook Pro, equipped with a 2.9 GHz Intel i7 processor. Matlab simulations were run in MatlabR2020.

given here, using both Matlab and C++ implementations. A few aspects are worth commenting. First, note that the rate of growth of the compute times for the two `SM` schemes is much lower than for the other schemes, since, as remarked previously, the Sherman-Morrison formula is solvable in $O(M)$ operations. Second, note how the C++ implementation yields considerable speedups compared to Matlab. Here, all the matrix operations were "unrolled", avoiding the need for sparse matrix libraries. Update (32) is remarkably fast, yielding under-real-time compute times for systems with over 250 nonlinearly coupled degrees-of-freedom. Compared to typical fully-implicit discretisations, such as `ICA_IT` coded in Matlab, `SM_C++` is about 350 times faster at OF = 12.

Concluding Remarks

The piano string was modelled using a nonlinear, geometrically exact model, and including the hammer-string interaction. The model was first discretised in space using the finite difference method, and then "quadratised" via a scalar auxiliary state function. The resulting system was then integrated in time. It was shown that the update matrix lends itself to a fast inversion via the Sherman-Morrison formula. A C++ implementation of the proposed scheme was given, yielding speedups of a few orders of magnitude compared to Matlab realisations of previously available schemes. In particular, real-time simulations of typical piano strings are now available. The proposed schemes are part of a larger class of schemes of recent development, based on the idea of quadratisation.

Acknowledgments

This work was supported by the European Research Council (ERC), under grant 2020-StG-950084-NEMUS.

References

[1] P. Morse and U. Ingard. *Theoretical Acoustics*. Princeton University Press, Princeton, NJ, USA, 1968.

[2] S. Bilbao, A. Torin, and V. Chatziioannou. Numerical modeling of collisions in musical instruments. *Acta Acust United with Acust*, 101:155–173, 2015.

[3] T. Itoh and K. Abe. Hamiltonian-conserving discrete canonical equations based on variational difference quotients. *J. Comput. Phys.*, 76(1):85–102, 1988.

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

[5] J. Chabassier and P. Joly. Energy preserving schemes for nonlinear Hamiltonian systems of wave equations: Application to the vibrating piano string. *Comput Method Appl Mech*, 199(45):2779–2795, 2010.

[6] J. Chabassier. *Modélisation et simulation numérique d'un piano par modèles physiques. (Modeling and simulation of a piano by physical modelling.)*. PhD thesis, Ecole Polytechnique X, Paris, 2012.

[7] F. Fontana and E. Bozzo. Newton–Raphson solution of nonlinear delay-free loop filter networks. *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, 27(10):1590–1600, 2019.

[8] V. Chatziioannou, S. Schmutzhard, and S. Bilbao. On iterative solutions for numerical collision models. In *Proc Int Conf On Digital Audio Effects (DAFx-17)*, Edinburgh, UK, September 2017.

[9] 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.

[10] A. Falaize. *Modélisation, simulation, génération de code et correction de systèmes multi-physiques audios: Approche par réseau de composants et formulation Hamiltonienne À Ports. (Modeling, Simulation, code generation and correction of multiphysics audio systems: Component network approach and Port-Hamiltonian formulation)*. PhD thesis, Université Pierre et Marie Curie, Paris, July 2016.

[11] N. Lopes. *Approche passive pour la modélisation, la simulation et l'étude d'un banc de test robotisé pour les instruments de type cuivre. (Passive approach for modelling, simulation and study of a robotic test bench for brass instruments.)*. PhD thesis, Université Pierre et Marie Curie, Paris, July 2016.

[12] X. Yang. Linear, first and second-order, unconditionally energy stable numerical schemes for the phase field model of homopolymer blends. *J Comput Phys*, 327:294–316, 2016.

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

[14] J. Zhao, Q. Wang, and X. Yang. Numerical approximations for a phase field dendritic crystal growth model based on the invariant energy quadratization approach. *Int J Numer Meth Eng*, 110(3):279–300, 2017.

[15] M. Ducceschi and S. Bilbao. Non-iterative, conservative schemes for geometrically exact nonlinear string vibration. In *Proc Int Conf Acoust (ICA 2019)*, Aachen, Germany, September 2019.

[16] M. Ducceschi and S. Bilbao. Simulation of the geometrically exact nonlinear string via energy quadratisation. *J Sound Vib*, 2022 (in press).

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

[18] S. Bilbao and M. Ducceschi. Fast explicit algorithms for hamiltonian numerical integration. In *Proceedings of the European Nonlinear Dynamics Conference*, Lyon, France, July 2022.

[19] 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*, 21:124–127, 1950.

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

[21] M. Ducceschi, S. Bilbao, S. Willemsen, and S. Serafin. Linearly-implicit schemes for collisions in musical acoustics based on energy quadratisation. *J. Acoust. Soc. Am.*, 149(5):3502–3516, 2021.

[22] M. Ducceschi and S. Bilbao. Linear stiff string vibrations in musical acoustics: Assessment and comparison of models. *J Acoust Soc Am*, 140(4):2445–2454, 2016.

[23] M. Ducceschi and S. Bilbao. Conservative finite difference time domain schemes for the prestressed Timoshenko, shear and Euler-Bernoulli beam equations. *Wave Motion*, 89:142 – 165, 2019.

[24] C. Valette and C. Cuesta. *Mécanique de la corde vibrante*. Hermès, Paris, 1993.

[25] J. Chabassier and M. Duruflé. Physical parameters for piano modeling. Technical report, 2012. Available at `https://hal.inria.fr/hal-00688679v1/document`.

[26] V. Chatziioannou and M. van Walstijn. Energy conserving schemes for the simulation of musical instrument contact dynamics. *J Sound Vib*, 339:262–279, 2015.