EFFICIENT SIMULATION OF THE BOWED STRING IN MODAL FORM

Riccardo Russo, Michele Ducceschi

Department of Industrial Engineering
University of Bologna
Viale Risorgimento 2, Bologna, Italy
riccardo.russo19@unibo.it
michele.ducceschi@unibo.it

Stefan Bilbao

Acoustics & Audio Group
University of Edinburgh
Edinburgh, UK
sbilbao@ed.ac.uk

ABSTRACT

The motion of a bowed string is a typical nonlinear phenomenon resulting from the interaction of the bow and the string via a friction force. The system can be described using suitable differential equations, where the friction force depends on the relative speed between the bow and the string. The bow supplies the energy that sets the string into motion, while dissipating part of it in a nonlinear feedback. Numerically, a sound implementation of the energy balance is necessary to ensure stability of the time stepping schemes. Implicit discretisations are known to yield energy-consistent algorithms, though they most often rely on the use of iterative nonlinear root finders such as Newton-Raphson. This carries several implementation issues, including variable operation cost, constraints on the time step to ensure existence and uniqueness, and the problem of choosing appropriate halt conditions. Recently, a novel method was developed, in the context of virtual-analogue simulation, allowing to solve nonlinear systems of ordinary differential equations non-iteratively. In this paper, it is proposed to extend these algorithms to bowed string simulation. Case studies of a mass-spring system and an ideal string coupled with a bow are investigated. Finally, the musical case of a bowed stiff string with loss is considered. When semi-discretisation is performed using a modal approach, a fast algorithm is available, yielding compute times below real-time for typical musical strings.

1. INTRODUCTION

Bowed strings simulation is a topic of longstanding interest in musical acoustics. Initial observations on bowed string motion were conducted by Helmholtz in the 19th century [1] and later extended by Raman [2], who provided the first mathematical theory. These studies laid the groundwork for many successive works [3]. A crucial part of the bowed string model lays in the law that governs the excitation force. It is common to consider the excitation to happen at a single point, and the force to be a frictional law dependent only on the relative velocity between the string and the bow. Under these assumptions, the force experienced by the string due to the bow is represented by a characteristic friction curve, as a function of the relative velocity. One common such curve is due to Smith and Woodhouse, obtained from experimental observations of the motion of a mass on a rosin-coated conveyor belt [4]. This curve is sometimes called the "classical" friction curve in this context [5], since analogous curves had in fact been used in several previous works, such as those by Friedlander [6], McIntyre and Woodhouse [7], and others. In the context of numerical simulation, curves with a "soft" characteristic have been proposed [8]: these are the curves considered in this work.

Numerically, bowed string simulation has been performed using digital waveguides [9, 10, 11, 12], as well as time domain methods [13]. Because of the strong nonlinear behaviour of the bow-string interaction, the numerical stability of the underlying time-stepping procedure must be ensured: one possible approach is to make use of energy methods [14, 8], encapsulating a notion of passivity. Extensive work on bowed string simulations using finite difference (FDTD) methods have been carried out by Desvages [13]. There, a complex model is developed, taking into account two directions of polarization of the string, the "classical" friction curve by Smith and Woodhouse [4], and the player's finger pressure on the string. Numerically, a passive energy balance ensuring stability is obtained using a fully implicit numerical scheme, relying on the use of iterative routines (Newton-Raphson). These are generally computationally expensive and are serial in nature. Other theoretical and computational issues emerge, such as variable cost at each time step and choices regarding appropriate tolerance thresholds. Furthermore, existence and uniqueness of the numerically-computed solutions is generally not ensured [15].

Recently, a numerical method was proposed [16, 17], in the context of virtual-analogue systems, employing a linearly-implicit time-stepping procedure for the solution of nonlinear ordinary differential equations (ODEs). This method is non-iterative, in that the update is expressed as one single linear system. Besides the obvious computational advantage [16], this method avoids entirely the issues typical of iterative procedures.

In this paper, an extension of the non-iterative method presented in [16, 17] is given. First, the bowed string model is expressed as a first-order-in-time system of partial differential equations (PDEs). Then, semi-discretisation in space is performed in order to reduce the problem to a system of nonlinearly coupled ODEs, in a way amenable to the structure of a Port-Hamiltonian system (PHS) [18, 19, 20]. Semi-discretisation is performed in two ways, using 1): a finite-difference form and 2): a modal form. The non-iterative time-stepping procedure from [16, 17], is then adopted to advance the equations in discrete time. It will be shown that the modal form presents an efficient update, expressed as a block-diagonal matrix plus a rank-1 perturbation, yielding a fast inversion.

The paper is structured as follows. Section 2 presents the case study of a bowed mass, including a comparison between various iterative and non-iterative discretisations. Section 3 considers a distributed resonator described by the simple wave equation coupled to a bow. Various numerical approaches are proposed, including a

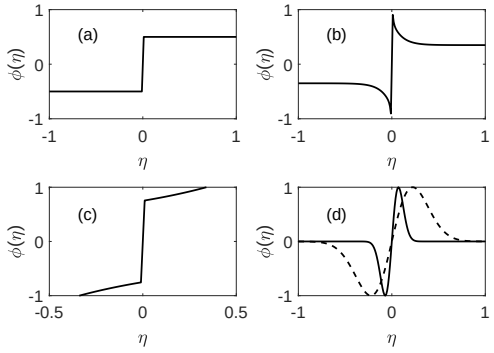

Figure 1: Various friction characteristics: (a) Coulomb dry friction, with equation $\phi(\eta) = \text{sgn}(\eta)\epsilon$, with $\epsilon = 1/2$; (b) the "classical" curve by Woodhouse and Smith [4], with the form $\phi(\eta) = \text{sgn}(\eta)(0.4e^{-|\eta|/0.01}+0.45e^{-|\eta|/0.1}+0.35)$; (c) the reconstructed curve by Galluzzo [5], defined by $\phi(\eta) = \text{sgn}(\eta)(0.4e^{-|\eta|/0.01} + 0.35)$; (d) the continuous approximation defined in equation 2, with $a = 10$ (dashed) and $a = 100$ (solid).

modal solution with a fast update. In section 4 the musical case of a stiff string with loss is presented.

2. CASE STUDY: THE BOWED MASS

Before examining the distributed string-bow system, it is convenient to start with the simpler case of a system with two degrees of freedom: a mass-spring excited by a bow. The system can be described by two coupled ODEs [8],

$$\ddot{u} = -\omega_0^2 u - F_B \phi(\eta)\,, \quad \eta = \dot{u} - v_B\,. \tag{1}$$

Here, $u = u(t) : \mathbb{R}_0^+ \to \mathbb{R}$ is the displacement of the mass in m, depending on time $t \geq 0$; ω_0 is the angular frequency of the oscillator, in rad/s; $\eta(t)$ is the relative velocity between the bow and the mass in m/s. The function $v_B(t)$ (assumed known) is the bow velocity, in m/s, while $F_B(t) > 0$ (also known) is the bow force normalised by the object's mass. The function $\phi = \phi(\eta) : \mathbb{R} \to \mathbb{R}$ is a dimensionless friction coefficient, expressed as a function of the relative velocity only. As mentioned in the introduction, various choices are available, some of which are shown in Figure 1. Here, the "soft" friction characteristic proposed in [8] (Chapter 4) is used (panel (d) of Figure 1). It may be expressed as:

$$\phi(\eta) = \sqrt{2a}\,\eta\, e^{-a\eta^2+1/2}, \tag{2}$$

where a is a free parameter. Note that ϕ satisfies

$$\eta\,\phi(\eta) \geq 0, \quad \lim_{|\eta|\to 0} \phi(\eta)/\eta = \sqrt{2ae} < \infty\,. \tag{3}$$

The first property in (3) is referred to as sector-boundedness, here to sector $[0, \infty]$. Both these properties will be used in the non-iterative numerical schemes presented below.

2.1. Energy Balance

An energy balance can be obtained by multiplying the first equation in (1) by $\dot{u}$. One gets

$$\dot{H} = -F_B(\eta + v_B)\phi\,(\eta) \tag{4}$$

where the energy H (scaled by mass) is

$$H\,(u, \dot{u}) = \frac{\dot{u}^2}{2} + \frac{\omega_0^2 u^2}{2}, \tag{5}$$

In the zero-velocity case ($v_B = 0$), owing to the first property in (3), one has: $\dot{H} \leq 0$, therefore the system dissipates.

2.2. The Bowed Mass as a First-Order System

As anticipated above, it is convenient to formulate the equation of motion (1) in first-order form. First, it is necessary to introduce the generalised coordinate q and momentum p:

$$q = \omega_0 u, \quad p = \dot{u}. \tag{6}$$

Note that p is momentum normalised by mass, with dimensions of velocity. The energy takes the form:

$$H(q, p) = \frac{p^2}{2} + \frac{q^2}{2}. \tag{7}$$

Therefore, equation (1) becomes

$$\dot{\mathbf{x}} = \mathbf{J}\nabla H - \mathbf{f} F_B \phi(\eta), \quad \eta = \mathbf{f}^\intercal \mathbf{x} - v_B. \tag{8}$$

Here:

$$\mathbf{x} = \begin{bmatrix} q \\ p \end{bmatrix}, \quad \mathbf{J} = \begin{bmatrix} 0 & \omega_0 \\ -\omega_0 & 0 \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{9}$$

and the gradient can be expressed as $\nabla = [\partial/\partial q, \partial/\partial p]^\intercal$. Thus:

$$\nabla H = \begin{bmatrix} q, p \end{bmatrix}^\intercal. \tag{10}$$

An energy balance can be obtained by multiplying (8) on the left by $(\nabla H)^\intercal$. Since $\mathbf{J}$ is skew-symmetric, and owing to the chain rule $\dot{H} = (\nabla H)^\intercal \dot{\mathbf{x}}$, one obtains in the zero-velocity case

$$\dot{H} = -F_B\, p\, \phi(p) \leq 0, \tag{11}$$

that is, the system dissipates, and is therefore passive. System (8) has the structure of a PHS, including energy storage elements, and dissipation induced by the bow [18, 19, 20]. In the continuous case, system (8) and equation (1) are entirely equivalent: one can be obtained from the other. Nevertheless, in the discrete case they will be discretised using distinct methods, as described below.

2.3. Time Difference Operators

Systems (1) and (8) will be now integrated using suitable energy-passive numerical schemes. Time is discretised with a time step k, yielding a sample rate $f_s = 1/k$. Note that k here denotes the time step, since h will be used to denote the grid spacing in the finite difference schemes for the bowed string, in Section 3.2. This is the notation used in various textbooks, see e.g. [8, 21]. Then, one defines the time series u^n, which represents an approximation to the continuous function $u(t)$ at time step $t = nk$, where $n \in \mathbb{N}$ is the time index. The basic operators in discrete time are the identity and shift operators, defined as:

$$1u^n = u^n, \quad e_+ u^n = u^{n+1}, \quad e_- u^n = u^{n-1}. \tag{12}$$

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

$$\delta_+ = \frac{e_+ - 1}{k},\ \delta_- = \frac{1 - e_-}{k},\ \delta_\cdot = \frac{e_+ - e_-}{2k}. \tag{13}$$

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

$$\delta_2 = \delta_+ \delta_- . \tag{14}$$

Finally, averaging operators can be written as:

$$\mu_+ = \frac{e_+ + 1}{2},\ \mu_- = \frac{1 + e_-}{2},\ \mu_\cdot = \frac{e_+ + e_-}{2}. \tag{15}$$

2.4. Time Domain Discretisation of the Bowed Mass

Three schemes are now presented, the first discretising the second-order system (1), and the other two discretising the first-order system (8).

2.4.1. Iterative Discretisation of the Second-Order System

A possible discretisation of equation (1) is:

$$\delta_2 u^n = -\omega_0^2 u^n - F_B^n \phi(\delta_\cdot u^n - v_B^n), \tag{16}$$

where F_B^n, v_B^n are time series representing the bow force and velocity, allowed to vary over time. A discrete energy balance, in the zero-velocity case ($v_B = 0$), is obtained by multiplying of the left by $\delta_\cdot u^n$:

$$\delta_+ \mathfrak{h}^{n-1/2} = -F_B^n \, \delta_\cdot u^n \phi(\delta_\cdot u^n) \leq 0, \tag{17}$$

that is, the discrete system dissipates. The discrete energy takes the form:

$$\mathfrak{h}^{n-1/2} = \frac{(\delta_- u^n)^2}{2} + \frac{\omega_0^2 u^n e_{t-} u^n}{2}, \tag{18}$$

and this is non-negative under the condition that

$$k < 2/\omega_0, \tag{19}$$

see e.g. [8] (Chapter 3) for a proof. Note that such condition arises solely as a consequence of the discretisation of the linear part, since the nonlinear dissipation is guaranteed in (16). Denoting $r \triangleq \delta_{t\cdot} u^n$, $b \triangleq 2u^{n-1} - 2u^n + \omega_0^2 k^2 u^n$, scheme (16) can be written as

$$2kr + b + k^2 F_B^n \phi(r - v_B^n) = 0, \tag{20}$$

which is a nonlinear algebraic equation in r, solvable with a nonlinear root finder such as Newton-Raphson. Notice that a condition on existence and uniqueness of the solution of (20) must be given, and this appears as a further constraint on the time step k, see [8] (Chapter 4).

2.4.2. Iterative Discretisation of the First-Order System

A possible discretisation of the first-order system (8) can be obtained as follows

$$\delta_+ \mathbf{x}^n = \mathbf{J} \nabla \mathfrak{h}^{n+1/2} - \mathbf{f} F_B \phi(\mu_+ \eta^n), \quad \eta^n = \mathbf{f}^\intercal \mathbf{x}^n - v_B^n. \tag{21}$$

Here, the discrete energy is given as:

$$\mathfrak{h}(q^n, p^n) \triangleq \mathfrak{h}^n = \frac{(p^n)^2}{2} + \frac{(q^n)^2}{2}. \tag{22}$$

In (21), the form of the discrete gradient $\nabla \mathfrak{h}^{n+1/2}$ can be obtained from a suitable energy-conserving discretisation, see e.g. [19, 22, 23]. In this respect, the partial derivative of $\mathfrak{h}$ with respect to q is given as:

$$\delta_{q+} \mathfrak{h}(q^n, p^n) = \frac{\mathfrak{h}(q^{n+1}, p^n) - \mathfrak{h}(q^n, p^n)}{q^{n+1} - q^n} = \mu_+ q^n. \tag{23}$$

The partial derivative with respect to p can be defined analogously. Therefore, one has:

$$\nabla \mathfrak{h}^{n+1/2} \triangleq [\delta_{q+} \mathfrak{h}^n, \delta_{p+} \mathfrak{h}^n]^\intercal = \mu_+ \left[q^n, p^n\right]^\intercal = \mu_+ \mathbf{x}^n. \tag{24}$$

Note that, under such choice for the discrete gradient, scheme (21) is equivalent to the midpoint method. The discrete balance is obtained by multiplying on the left (21) by $\nabla \mathfrak{h}^\intercal$, to get (in the zero-velocity case)

$$\delta_+ \mathfrak{h}^n = -F_B^n \, \mu_+ p^n \, \phi(\mu_+ p^n) \leq 0, \tag{25}$$

that is, the system dissipates. Note that, in this case, the scheme is uncoditionally stable, by virtue of the non-negativity of the discrete energy. The update equation of (21) is

$$\left(\frac{\mathbf{I}}{k} - \frac{\mathbf{J}}{2}\right) \mathbf{x}^{n+1} - \left(\frac{\mathbf{I}}{k} + \frac{\mathbf{J}}{2}\right) \mathbf{x}^n + \mathbf{f} F_B^n \phi(\mu_+ \eta^n) = 0, \tag{26}$$

which is a nonlinear algebraic system (following the implicit definition of $\mathbf{x}^{n+1}$ appearing in the argument of the nonlinear function ϕ). An iterative root finder is therefore necessary in this case.

2.4.3. Non Iterative Discretisation of the First-Order System

An alternative discretisation of the first-order system (8) is now given, and adapted from [16, 17]. There, a family of non-iterative schemes is presented in the context of stiff nonlinear ODEs modelling audio circuits. These schemes yield good numerical behaviour without the need for iterative techniques, as will be seen in the numerical examples below. These are:

$$\boldsymbol{\sigma}^{(P)}(\mathbf{x}^n) \delta_+ \mathbf{x}^n = \mathbf{J} \nabla \mathfrak{h}^{n+1/2} - \mathbf{f} F_B^n d^n \mu_+ \eta^n. \tag{27}$$

Here, $\boldsymbol{\sigma}^{(P)}(\mathbf{x}^n)$ is a factor taking the form of a perturbation expansion. This can be set so to yield a $(P+1)$-accurate truncation error. The first two terms are given explicitly as

$$\boldsymbol{\sigma}^{(0)} = \mathbf{I}, \quad \boldsymbol{\sigma}^{(1)} = \mathbf{I} + \frac{k F_B}{2} (\lambda^n - d^n) \mathbf{f}\mathbf{f}^\intercal, \tag{28}$$

where

$$\lambda^n \triangleq d\phi/d\eta\big|_{\eta=\eta^n}\ , \quad d^n \triangleq \phi/\eta\big|_{\eta=\eta^n}\ . \tag{29}$$

Note that both λ and d are well-defined, owing to properties (3). The definition of η is as in (21), and the discrete gradient $\nabla \mathfrak{h}^n$ is as in (24). Using $P = 1$, one obtains a second-order accurate scheme of the form

$$\mathbf{A}^n \mathbf{x}^{n+1} = \mathbf{B}^n \mathbf{x}^n + \mathbf{f} F_B^n d^n \mu_+ v_B^n, \tag{30}$$

where

$$\mathbf{A}^n = \frac{\mathbf{I}}{k} + \frac{F_B \lambda^n}{2} \mathbf{f}\mathbf{f}^\intercal - \frac{\mathbf{J}}{2},\ \mathbf{B}^n = \frac{\mathbf{I}}{k} + F_B \left(\frac{\lambda^n}{2} - d^n\right) \mathbf{f}\mathbf{f}^\intercal + \frac{\mathbf{J}}{2}.$$

Note that both $\mathbf{A}^n$, $\mathbf{B}^n$ are computed using values from previous time steps, hence the update $\mathbf{x}^{n+1}$ in (30) is expressed as the solution of a single linear system. Numerical passivity of schemes (27) is harder to get to, compared to the iterative discretisations shown above. Partial results are available in [17], but are not included here for brevity. However, the numerical tests presented below allow to gather some insight on the stability properties of these schemes, compared to the iterative discretisations shown above, and to a standard explicit integrator (Forward Euler).

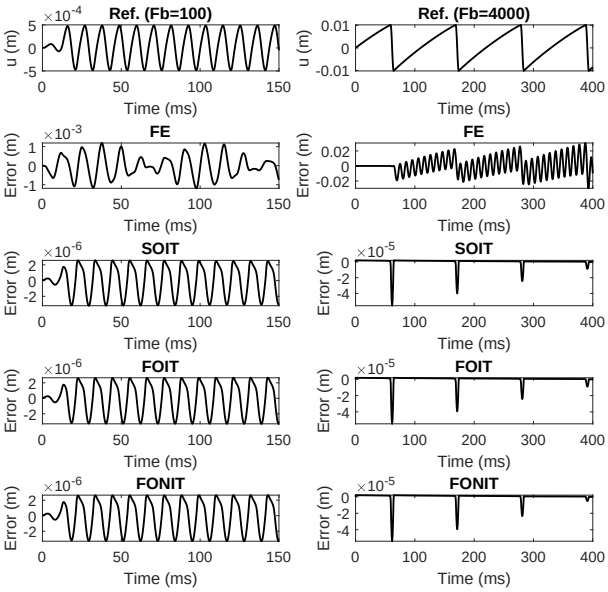

Figure 2: Comparison between the different bowed mass simulations under two choices of pressure F_B, using the friction function (2). In all cases it was set: $a = 100$, $f_0 = \omega_0/2\pi = 100$ Hz, $v_B = 0.2$ m/s. The acronyms indicate: the Forward Euler integrator (FE), the second-order iterative scheme (SOIT), the first-order iterative scheme (FOIT) and the first-order non-iterative scheme (FONIT). "Ref." indicates the reference solution.

2.5. Solvers Comparison

Figure 2 displays the results computed with different solvers under two choices of F_B. A reference solution was obtained running scheme (16) with a sample rate $f_s = 30{\cdot}44100$ Hz. Then, schemes (16), (21) and (27) were run using a sample rate $f_s = 2 \cdot 44100$ Hz, and the error was computed as the difference between the outputs of these schemes and the reference solution. Newton-Raphson was run with a threshold of 10^{-9}. A standard explicit integrator, Forward Euler (see e.g. [21]), was also included in the test. For the lower bowing pressure ($F_B = 100$), Forward Euler yields an error three orders of magnitude larger than the other schemes. More strikingly, for the larger bowing pressure ($F_B = 4000$) Forward Euler displays a clear instability. On the other hand, note that the non-iterative scheme (27) displays robust numerical behaviour, comparable to that of the iterative schemes.

3. CASE STUDY: THE IDEAL STRING

In this section, the case of a bowed string is considered, employing a simple model for the string in the form of the ideal 1-D wave equation. As for the case of the bowed mass, this section is intended as a useful test case, here including a distributed resonator. Important features for realistic sound synthesis, such as the string's stiffness and losses, are neglected here, but will be included below in the case study of Section 4. The string-bow system is described by the following:

$$\partial_t^2 u = c^2 \partial_x^2 u - F_B \delta(x - x_B)\phi(\eta), \tag{31a}$$

$$\eta = \int_0^L \delta(x - x_B)\partial_t u \,\mathrm{d}x - v_B. \tag{31b}$$

Here, $u = u(x,t) : [0,L]\times\mathbb{R}^+ \to \mathbb{R}$ is the function describing the displacement of a string of length L. The value c is the wave speed on the string and $\eta = \eta(t) \in \mathbb{R}$ is the relative velocity of the bow and the string at the bowing location x_B. Notice that the notation ∂_t^j, ∂_x^j indicates the j^{th} partial derivative of u with respect to t and x, respectively. An energy balance can be obtained by taking the $\mathcal{L}^2$ inner product of (31a) with $\partial_t u$ over the domain $[0, L]$. If energy-passive boundary conditions are considered, in the zero input-case ($v_B = 0$) dissipation is guaranteed. Here, Dirichlet boundary conditions are used; therefore: $u(x,t) = 0$ at $x = 0$ and $x = L$. After integration by parts (see e.g. [8] (Chapter 6)), one obtains in the zero-velocity case:

$$\frac{d\int \mathcal{H}\,\mathrm{d}x}{dt} = -F_B\,\partial_t u(x_B,t)\,\phi(\partial_t u(x_B,t)) \leq 0, \tag{32}$$

where the energy density is:

$$\mathcal{H}(\partial_x u, \partial_t u) = \frac{1}{2}(\partial_t u)^2 + \frac{c^2}{2}(\partial_x u)^2. \tag{33}$$

3.1. The Bowed String as a First-Order System

As in the case of the mass, it is convenient to express the system in first-order form. First, one defines the generalised coordinate and momentum:

$$q = c\,\partial_x u, \quad p = \partial_t u. \tag{34}$$

The energy density now takes the form:

$$\mathcal{H} = \frac{p^2}{2} + \frac{q^2}{2}. \tag{35}$$

The equations of motion in first-order form then are:

$$\frac{\partial \mathbf{x}}{\partial t} = \mathbf{J}\nabla\mathcal{H} - \boldsymbol{\zeta} F_B \phi(\eta), \tag{36a}$$

$$\eta = \int_0^L \boldsymbol{\zeta}^\intercal \mathbf{x}\,\mathrm{d}x - v_B, \tag{36b}$$

where:

$$\mathbf{x} = \left[q, p\right]^\intercal, \quad \boldsymbol{\zeta} = \left[0, \delta(x - x_B)\right]^\intercal, \quad \mathbf{J} = c\begin{bmatrix} 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial x} & 0 \end{bmatrix}.$$

The gradient is $\nabla = [\partial/\partial q, \partial/\partial p]^\intercal$. The energy balance is obtained by multiplying (36a) on the left by $\nabla\mathcal{H}^\intercal$, and integrating. Considering Dirichlet boundary conditions, and using the identity: $\int_0^L \nabla\mathcal{H}^\intercal \frac{\partial \mathbf{x}}{\partial t}\mathrm{d}x = \frac{d\int \mathcal{H}\,\mathrm{d}x}{dt}$, one obtains again the energy balance (32).

3.2. Spatial Difference Operators

Spatial difference operators are now introduced. First, the domain of the string is divided into M subintervals of length h, the grid spacing. This yields $M + 1$ discretisation points, including the end points. All throughout, boundary conditions of fixed (i.e. Dirichlet) type will be considered. therefore, the end

points of the domain need not be stored or updated. The continuous functions $u(x,t), q(x,t), p(x,t)$ appearing in (31) and (36) are all approximated by grid functions $\mathbf{u}(t)$, $\mathbf{q}(t)$ and $\mathbf{p}(t)$, which are $(M-1)\times 1$, $M\times 1$ and $(M-1)\times 1$ column vectors, respectively. Note that $\mathbf{q}$ is spatially interleaved with respect to $\mathbf{u}$ and $\mathbf{p}$. The difference matrix $\mathbf{D}^-$ acting on the state vector $\mathbf{u}$ can be defined as

$$\mathbf{D}^-\mathbf{u} = \frac{1}{h}([\mathbf{u}^\intercal, 0] - [0, \mathbf{u}^\intercal]). \tag{37}$$

Note that, since fixed boundaries were considered, this matrix is rectangular, with dimensions $M\times(M-1)$. The matrix $\mathbf{D}^+$ can be simply defined as $\mathbf{D}^+ = -(\mathbf{D}^-)^\intercal$, and is a rectangular matrix of dimensions $(M-1)\times M$. The 1-D Laplace operator is then obtained by composing the difference matrices: $\mathbf{D}^2 = \mathbf{D}^+\mathbf{D}^-$, and is a square matrix of dimensions $(M-1)\times(M-1)$, satisfying the Dirichlet conditions.

3.3. Semi-Discrete Formulations of the Bowed String

3.3.1. Second-Order System

Given the definitions above, a semi-discrete version of (31a) is obtained immediately as

$$\ddot{\mathbf{u}} = c^2\mathbf{D}^2\mathbf{u} - \boldsymbol{\xi}F_B\phi(\eta) \tag{38a}$$
$$\eta = h\boldsymbol{\xi}^\intercal\dot{\mathbf{u}} - v_B \tag{38b}$$

where $\boldsymbol{\xi}$ is any suitable discrete version of the Dirac delta function, obtained for instance via Lagrange interpolants, and represented as an $(M-1)\times 1$ column vector.

3.3.2. First-Order System

The generalised coordinates and momenta can be written as:

$$\mathbf{q} = c\,\mathbf{D}^-\mathbf{u}, \quad \mathbf{p} = \dot{\mathbf{u}}. \tag{39}$$

The energy is now:

$$H(\mathbf{q},\mathbf{p}) = \frac{\mathbf{p}^\intercal\,\mathbf{p}}{2} + \frac{\mathbf{q}^\intercal\,\mathbf{q}}{2}. \tag{40}$$

The equations of motion in semi-discrete form take the form:

$$\dot{\mathbf{x}} = \mathbf{J}\nabla H - \boldsymbol{\zeta}F_B\phi(\eta), \tag{41a}$$
$$\eta = h\,\boldsymbol{\zeta}^\intercal\,\mathbf{x} - v_B. \tag{41b}$$

where

$$\mathbf{x} = \left[\mathbf{q},\mathbf{p}\right]^\intercal, \quad \boldsymbol{\zeta} = \left[\mathbf{0},\boldsymbol{\xi}\right]^\intercal, \quad \mathbf{J} = c\begin{bmatrix}\mathbf{0} & \mathbf{D}^- \\ \mathbf{D}^+ & \mathbf{0}\end{bmatrix}, \tag{42}$$

and note that $\mathbf{J}$ a square, skew-symmetric matrix, yielding the form typical of PHS [19].

3.3.3. Modal Form

As an alternative approach, the continuous equations (31) are now semi-discretised using a modal expansion. To that end, the system solution u can be rewritten as a superposition of modal displacements:

$$u(x,t) = \sum_{i=1}^{N} X_i(x)s_i(t), \tag{43}$$

where N, the number of modes, is in theory infinite, but for practical purposes is truncated to a finite integer. Thus, equation (43) can be rewritten in vector form: $u(x,t) = \mathbf{X}^\intercal(x)\mathbf{s}(t)$. For an ideal string of length L, with Dirichlet boundary conditions, the mode shapes (eigenfunctions) take the form [8]: $X_i(x) = \sqrt{2/L}\cdot$ $\cdot\sin(i\pi x/L)$. Substituting equation (43) into (31a) and (31b), multiplying (31a) on the left by $\mathbf{X}$ and taking an $\mathcal{L}^2$ inner product over the string length yields the equations that describe the modal system:

$$\ddot{\mathbf{s}} = -\boldsymbol{\Omega}_0^2\mathbf{s} - F_B\mathbf{X}(x_B)\phi(\eta) \tag{44a}$$
$$\eta = \mathbf{X}^\intercal(x_B)\dot{\mathbf{s}} - v_B. \tag{44b}$$

Here, $\boldsymbol{\Omega}_0$ is a diagonal matrix containing the eigenfrequencies of the system, which are: $\omega_i = i\pi c/L$, $i = 1,\ldots,N$. In view of a non-iterative discretisation, it is useful to express the equation of motion (44a) in first-order form. First, one can define $\tilde{\mathbf{q}} \triangleq \boldsymbol{\Omega}_0\mathbf{s}$, $\tilde{\mathbf{p}} \triangleq \dot{\mathbf{s}}$ and $\tilde{\mathbf{x}} \triangleq [\tilde{\mathbf{q}},\tilde{\mathbf{p}}]^\intercal$. The energy here takes the form:

$$\tilde{H}(\tilde{\mathbf{q}},\tilde{\mathbf{p}}) = \frac{\tilde{\mathbf{p}}^\intercal\,\tilde{\mathbf{p}}}{2} + \frac{\tilde{\mathbf{q}}^\intercal\tilde{\mathbf{q}}}{2}. \tag{45}$$

Therefore, equations (44a) and (44b) can be written as:

$$\dot{\tilde{\mathbf{x}}} = \tilde{\mathbf{J}}\nabla\tilde{H} - \tilde{\boldsymbol{\zeta}}\,F_B\phi(\eta) \tag{46a}$$
$$\eta = \tilde{\boldsymbol{\zeta}}^\intercal\tilde{\mathbf{x}} - v_B, \tag{46b}$$

where:

$$\tilde{\mathbf{J}} = \begin{bmatrix}\mathbf{0} & \boldsymbol{\Omega}_0 \\ -\boldsymbol{\Omega}_0 & \mathbf{0}\end{bmatrix}, \quad \tilde{\boldsymbol{\zeta}} = \begin{bmatrix}\mathbf{0} \\ \mathbf{X}(x_B)\end{bmatrix}, \quad \nabla\tilde{H} = \begin{bmatrix}\tilde{\mathbf{q}} \\ \tilde{\mathbf{p}}\end{bmatrix}. \tag{47}$$

The string displacement at the desired output position x_{out} can be obtained by projecting the state vector $\mathbf{s}(t)$ onto the corresponding eigenfunctions; therefore: $u(x_{out},t) = \mathbf{X}^\intercal(x_{out})\mathbf{s}(t)$.

3.4. Fully-Discrete Formulations of the Bowed String

The semi-discrete formulations (38), (41), (46) are in the form of nonlinearly coupled systems of ODEs. These can now be discretised in time using the time difference operators defined in Section 2.

3.4.1. Iterative Discretisation of the Second-Order System

Integration of (38) may be performed simply as:

$$\delta_2\mathbf{u}^n = c^2\mathbf{D}^2\mathbf{u}^n - \boldsymbol{\xi}F_B\phi(\eta^n) \tag{48a}$$
$$\eta^n = h\,\boldsymbol{\xi}^\intercal\delta_\cdot\mathbf{u}^n - v_B \tag{48b}$$

A discrete energy balance can be obtained by multiplying (48a) on the left by $h\delta_\cdot(\mathbf{u}^n)^\intercal$ [8]. In the zero-velocity case one obtains:

$$\delta_+\mathfrak{h}^{n-1/2} = -F_B^n\,(h\boldsymbol{\xi}^\intercal\delta_\cdot\mathbf{u}^n)\phi(h\boldsymbol{\xi}^\intercal\delta_\cdot\mathbf{u}^n) \leq 0, \tag{49}$$

which is a discrete counterpart of the continuous energy balance (32). Here, the discrete energy has the form:

$$\mathfrak{h}^{n-1/2} = \frac{h}{2}\left((\delta_-\mathbf{u}^n)^\intercal(\delta_-\mathbf{u}^n) + c^2(\mathbf{D}^-\mathbf{u}^n)^\intercal\mathbf{D}^-e_{t-}\mathbf{u}^n\right).$$

This is a distributed version of the discrete energy obtained for the mass-spring, in (18). Like before, a stability condition arises as a

consequence of the explicit discretisation of the linear part. The energy is non-negative overall if and only if

$$h \geq ck, \tag{50}$$

that is, the CFL condition, see e.g. [8] (Chapter 6). Since the update $\mathbf{u}^{n+1}$ appears implicitly as the argument of ϕ in (48a), the scheme is fully-implicit, and a solution may be found using a suitable iterative routine (e.g. Newton-Raphson) in vector form (not shown here for brevity).

3.4.2. Iterative Discretisation of the First-Order System

Integration of (41) can be performed using the same discretisation employed in the case of the bowed mass in Section 2.4.2. Hence

$$\delta_+ \mathbf{x}^n = \mathbf{J} \nabla \mathfrak{h}^{n+1/2} - \boldsymbol{\zeta} F_B \phi(\mu_+ \eta^n), \tag{51a}$$
$$\mu_+ \eta^n = h \boldsymbol{\zeta}^\intercal \mu_+ \mathbf{x}^n - \mu_+ v_B^n \tag{51b}$$

Here, the discrete energy is

$$\mathfrak{h}^n = \frac{(\mathbf{p}^n)^\intercal \mathbf{p}^n}{2} + \frac{(\mathbf{q}^n)^\intercal \mathbf{q}^n}{2}, \tag{52}$$

and thus the discrete gradient is $\nabla \mathfrak{h}^{n+1/2} = [\mu_+ \mathbf{q}^n, \mu_+ \mathbf{p}^n]^\intercal$. Note that this scheme is unconditionally stable, since the energy is non-negative $\forall\ \mathbf{p}^n, \mathbf{q}^n$. Scheme (51) is in the form of a nonlinear algebraic system in the update $\mathbf{x}^{n+1}$, with a form analogous to (26), and hence an iterative root finder is required for its solution.

3.4.3. Non Iterative Discretisation of the First-Order System

A non-iterative integrator for (41) can be obtained using a scheme analogous (27), that is:

$$\boldsymbol{\sigma}^{(P)}(\mathbf{x}^n) \delta_+ \mathbf{x}^n = \mathbf{J} \nabla \mathfrak{h}^{n+1/2} - \boldsymbol{\zeta} F_B^n d^n \mu_+ \eta^n. \tag{53}$$

Choosing $P = 1$, one has

$$\boldsymbol{\sigma}^{(1)} = \mathbf{I} + \frac{kh}{2} F_B (\lambda^n - d^n) \boldsymbol{\zeta} \boldsymbol{\zeta}^\intercal \tag{54}$$

and a second-order accurate discretisation arises [16, 17]. Here, the definitions of λ and d are as in (29). The discrete gradient and η are the same as in (51). Isolating the state vector $\mathbf{x}^{n+1}$ in (53) results in an update equation analogous to (30), that is, a linear system.

3.4.4. Non Iterative Discretisation of the Modal System

The non-iterative solver can be applied for the integration of the the modal system 46. As previously, the scheme can be written in a form analogous to (30), that is:

$$\tilde{\mathbf{A}}^n \tilde{\mathbf{x}}^{n+1} = \tilde{\mathbf{B}}^n \tilde{\mathbf{x}}^n + F_B^n d^n \tilde{\boldsymbol{\zeta}} \mu_+ v_B^n, \tag{55}$$

with

$$\tilde{\mathbf{A}}^n = \frac{\mathbf{I}}{k} + \frac{F_B^n \lambda^n}{2} \tilde{\boldsymbol{\zeta}} \tilde{\boldsymbol{\zeta}}^\intercal - \frac{\tilde{\mathbf{J}}}{2},\ \tilde{\mathbf{B}}^n = \frac{\mathbf{I}}{k} + F_B^n \left(\frac{\lambda^n}{2} - d^n \right) \tilde{\boldsymbol{\zeta}} \tilde{\boldsymbol{\zeta}}^\intercal + \frac{\tilde{\mathbf{J}}}{2}.$$

It is now shown that the form of the matrix $\tilde{\mathbf{A}}^n$ lends itself naturally to a fast inversion. To that end, define

$$\mathbf{T} \triangleq \frac{\mathbf{I}}{k} - \frac{\tilde{\mathbf{J}}}{2} = \begin{bmatrix} \mathbf{I}/k & -\boldsymbol{\Omega}_0/2 \\ \boldsymbol{\Omega}_0/2 & \mathbf{I}/k \end{bmatrix}. \tag{56}$$

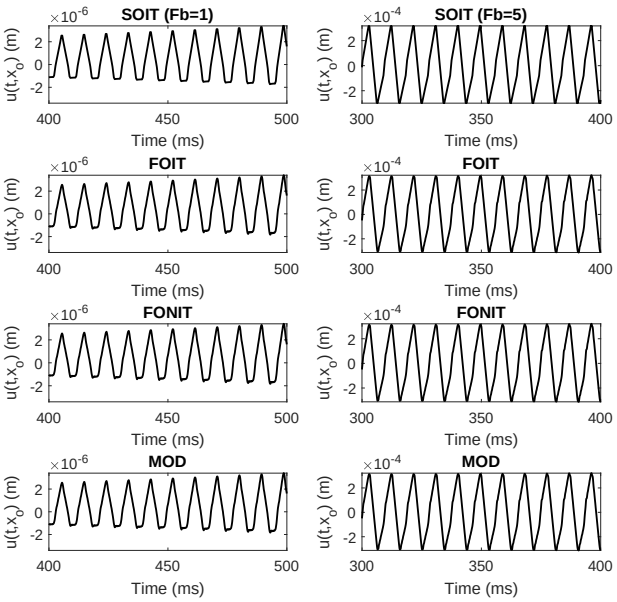

Figure 3: Ideal string simulation under two choices of F_B. In both cases it was set: $c = 150$ m/s, $L = 0.7$ m, $v_B = 0.2$ m/s, $a = 100$. The input and output positions were, respectively: $0.633 \cdot L$ and $0.33 \cdot L$. The acronyms indicate: the second-order iterative scheme (SOIT), the first-order iterative scheme (FOIT), the first-order non-iterative scheme (FONIT) and the modal scheme (MOD).

The matrix $\tilde{\mathbf{A}}^n$ is then expressed as the sum of $\mathbf{T}$ plus a rank-1 perturbation, invertible using the Sherman-Morrison formula [24]:

$$(\tilde{\mathbf{A}}^n)^{-1} = \mathbf{T}^{-1} - \frac{F_B^n \lambda^n}{2} \frac{\mathbf{T}^{-1} \tilde{\boldsymbol{\zeta}} \tilde{\boldsymbol{\zeta}}^\intercal \mathbf{T}^{-1}}{1 + \frac{F_B^n \lambda^n}{2} \tilde{\boldsymbol{\zeta}}^\intercal \mathbf{T}^{-1} \tilde{\boldsymbol{\zeta}}}. \tag{57}$$

Notice that $\mathbf{T}$ is block-diagonal; thus, it allows for a fast decomposition in the form of an block LUD factorisation [25]. One can write:

$$\begin{aligned} \mathbf{T} &= \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{21} & \mathbf{T}_{22} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{T}_{21} \mathbf{T}_{11}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{T}_{11} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\chi} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{T}_{11}^{-1} \mathbf{T}_{12} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \end{aligned} \tag{58}$$

where $\boldsymbol{\chi} \triangleq \mathbf{T}_{22} - \mathbf{T}_{21} \mathbf{T}_{11}^{-1} \mathbf{T}_{12}$ is the Schur complement, which in this case is diagonal, as it results from algebraic operations between diagonal matrices. This decomposition allows to invert the matrix $\mathbf{T}$ using a few serial operations involving diagonal matrices. Since $\mathbf{T}$ only contains constant values, it is possible to compute the factorisation components offline. All these operations allow to dramatically reduce the compute time, as shown below.

3.5. Solvers Comparison

Figure 3 shows the string's output displacement $u(x_{out}, t)$, computed using (48) (SOIT), (51) (FOIT), (53) (FONIT) and (55) (MOD), under two different values of F_B. The schemes were run at a sample rate $f_s = 2 \cdot 44100$ Hz. As explained in the previous section, both SOIT and FOIT require the Newton-Raphson algorithm in vector form, and thus the inversion of the Jacobian at each iteration: this operation was achieved using Matlab's own

backslash function. The tolerance was set to 10^{-9}. Backslash was also employed for the non-iterative scheme FONIT, whereas the modal system was solved with the efficient technique described above. The values of F_B yield slightly different types motion. When $F_B = 1$, the motion is steady, but not Helmholtz-like. Using a larger value for the bowing pressure, $F_B = 5$, results in a fully developed Helmholtz motion. The change of motion regime with bowing pressure is typical of musical strings, and is often summarised in diagrams such as Shelleng's [26]. It can be observed how all the different solvers yield solutions of comparable behaviour.

OS	F_B	SOIT	FOIT	FONIT	MOD
1	1	3.91 (3.00)	33.46 (3.00)	16.97	0.69
	5	4.89 (3.61)	47.39 (3.83)	17.05	0.88
	30	6.45 (4.58)	54.26 (4.28)	16.86	0.87
2	1	9.95 (3.00)	106.66 (3.00)	54.48	2.51
	5	12.21 (3.48)	144.99 (3.72)	54.72	2.47
	30	25.14 (6.34)	167.38 (4.15)	54.90	2.58

Table 1: Run-time/real-time ratio for the different schemes with two sample rates, under different values of F_B. OS indicates the oversampling factor, so that $f_s = \text{OS} \cdot 44100$. The string's parameters are the same as in Figure 3. The average number of iterations per time-step requested by Newton-Raphson to converge is reported in brackets for the iterative schemes.

Table 1 displays the run-time/real-time ratio for the various solvers, and, for the iterative schemes, the average number of iterations needed per time-step, with two sample rates, under different values of F_B. Run-times were measured with the Matlab `tic toc` function. As expected, the number of iterations increases with F_B, as the system becomes "stiffer". Run-time of the non-iterative schemes is clearly unaffected by F_B. It can be observed that the efficient modal algorithm is the fastest among all the solvers. On the other hand, the two non-modal, first-order systems are the slowest implementations. This is because both schemes require to explicitly invert a non-diagonal matrix (the Jacobian in the FOIT case, and $\mathbf{A}$ in the FONIT case), which is twice the size of the Jacobian of the SOIT scheme. Nevertheless, it has to be pointed out that no further optimisation was considered for SOIT, FOIT and FONIT: various speedups are available, but were not considered here for lack of space. The results in Table 1 highlight the importance of a sound implementation of linear system solvers, in particular in the presence of structured matrices, since such structures are often overlooked by compilers.

4. A MUSICAL EXAMPLE: THE DAMPED STIFF STRING IN MODAL FORM

As a musical application, the case of a string with stiffness and losses is now considered. The model reads

$$\partial_t^2 u = c^2 \partial_x^2 u - \kappa^2 \partial_x^4 u - 2\sigma_0 \partial_t u - F_B \delta(x - x_B)\phi(\eta). \quad (59)$$

Here, $c = \sqrt{T_0/\rho A}$, $\kappa = \sqrt{EI/\rho A}$, where ρ is the string material density, A is the cross-sectional area, T_0 is the applied tension, E is the Young's modulus, I is the moment of inertia and σ_0 is the damping coefficient.

In (59), the 1-D wave equation is augmented by a stiffness term, proportional to κ^2, coming from the Euler-Bernoulli theory of vibrating beams. This theory is largely satisfactory for musical purposes, see e.g. [27, 28]. Linear damping is here modelled by a term dependent on the string velocity times σ_0, the damping coefficient. In this configuration, each partial decays at the same rate, set by σ_0, yielding a rather crude model for dissipation. There are several ways to model frequency-dependent damping in the time domain, typically employing mixed space-time derivatives [8].

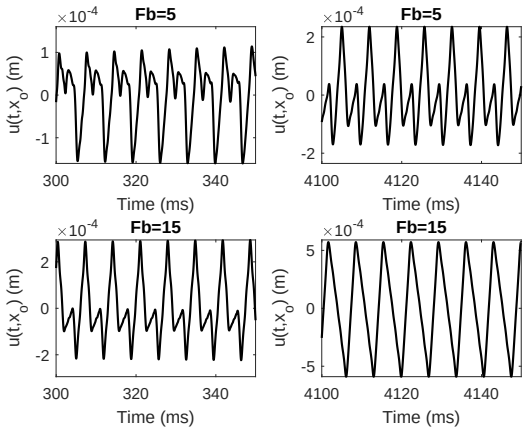

Figure 4: Vibration regimes of the stiff string under two values of F_B. Plots on the same row are snapshots of the same waveform, taken in different time instants. String parameters were the ones of a D3 cello string, taken from [13]. Bowing parameters, input and output positions were as in Figure 3.

OS = 1	OS = 2	OS = 5
0.19	0.39	0.91

Table 2: Run-time/real-time ratio for the stiff string, run with different oversampling factors. As before: $f_s = \text{OS} \cdot 44100$

As mentioned in Section 3.3, however, one of the advantages of the modal projection is the possibility of setting a decay constant for each one of the modes, without any additional computational effort. Implementation of refined physical loss profiles, such as that described by Valette and Cuesta [29], is immediate in this framework. After modal projection, (59) returns an augmented form of (44a), that is

$$\ddot{\mathbf{s}} = -\mathbf{\Omega}_s^2 \mathbf{s} - \mathbf{C}\dot{\mathbf{s}} - F_B \mathbf{X}(x_B)\phi(\eta), \quad (60)$$

where here $\mathbf{\Omega}_s$ is a diagonal matrix containing the natural frequencies of the stiff string, i.e. $\omega_i = \sqrt{(ci\pi/L)^2 + (\sqrt{\kappa}\, i\pi/L)^4}$, and where $\mathbf{C}$ is a diagonal matrix containing the appropriate damping constants matching the frequency-dependent loss profile of the Valette and Cuesta model. The resulting system can be simulated with an algorithm analogous to the one employed for the ideal string in Section 3.4.4. The only difference is that now the matrix $\mathbf{T}$ is augmented by the matrix $\mathbf{C}$. Since this matrix is diagonal, the efficiency of the algorithm is unaffected. Figure 4 shows different vibration regimes of the stiff string, under two values of F_B. The plots on the same row are snapshots of the same waveform, at different times: this allows to observe the string motion at the be-

ginning of the excitation, and after it reaches steady state. When $F_B = 5$, the string displays a kind of multi-slip motion, while in the case $F_B = 15$ it reaches Helmholtz motion after few seconds. Table 2 shows the run-time/real-time ratio for the stiff string: it is possible to see that the algorithm runs in real-time at high sample rates even in Matlab. Note that, compared to the simple wave equation, stiffness reduces the number of degrees of freedom, i.e. the number of harmonics within the audible range, thus lowering compute times. Sound samples for this model can be found at the following Github link [1].

5. CONCLUSIONS

This work presented the application of non-iterative solvers in the context of vibrating systems excited by the bow: the mass-spring, the ideal string, and the lossy, stiff string. The non-iterative schemes were developed in parallel to fully-implicit discretisations, obtained using previously available discrete-gradient methods. In order to implement the proposed method, the continuous equations were first expressed as first-order system of ODEs, yielding a form typical of Port-Hamiltonian systems. It was seen that the proposed schemes yield results comparable to the fully-implicit methods, while avoiding entirely the need for iterative routines such as Newton-Raphson. Yet, the proposed schemes are numerically robust, and outperform classic explicit integrators such as Forward-Euler in terms of stability. The string-bow was simulated using a number of iterative and non-iterative discretisation. In particular, after the string was semi-discretised using a modal approach, a fast algorithm was given, exploiting the structure of the update matrix. This yielded computations times below real-time, while allowing for an accurate representation of losses with the addition of a finely-tuned modal loss matrix. Fast implementations of the finite difference schemes in the case of the string-bow are also possible, but left as future work for lack of space. Future directions will also involve the implementation of the non-iterative schemes in a faster language, such as C++, and the development of software applications.

6. ACKNOWLEDGMENTS

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

7. REFERENCES

[1] H. L. F. Helmholtz, *On the Sensations of Tone as a Physiological Basis for the Theory of Music*, Cambridge University Press, third edition, 2009.

[2] C. V. Raman, "On the mechanical theory of the vibrations of bowed strings and of musical instruments of the violin family, with experimental verification of the results," *Indian Assoc. Cult. Sci. Bull.*, vol. 15, pp. 1–158, 1918.

[3] J. Woodhouse and P. Galluzzo, "The bowed string as we know it today," *Acta Acust. united Acust.*, vol. 90, pp. 579–589, 07 2004.

[4] J.H. Smith and J. Woodhouse, "The tribology of rosin," *J. Mech. Phys. Solids*, vol. 48, pp. 1633–1681, 08 2000.

[5] P. Galluzzo, J. Woodhouse, and H. Mansour, "Assessing friction laws for simulating bowed-string motion," *Acta Acust. united Acust.*, vol. 103, pp. 1080–1099, 11 2017.

[6] F. Friedlander, "On the oscillations of the bowed string," *Proc. Cambridge Phil. Soc.*, vol. 49, pp. 516–530, 1953.

[1] https://github.com/Rickr922/EfficientBowedString

[7] M. McIntyre and J. Woodhouse, "On the fundamentals of bowed string dynamics," *Acta Acust. united Acust.*, vol. 43, pp. 93–108, 09 1979.

[8] S. Bilbao, *Numerical Sound Synthesis*, John Wiley & Sons, Ltd, Chichester, UK, 2009.

[9] J. O. Smith, *Techniques for digital filter design and system identification with application to the violin*, Ph.D. thesis, Stanford University, 1983.

[10] J. O. Smith, "Physical modeling using digital waveguides," *Comput Music J.*, vol. 16, pp. 74, 1992.

[11] S. Serafin, *The sound of friction: Real-time models, playability and musical applications*, Ph.D. thesis, Stanford University, 2004.

[12] E. Maestre, C. Spa, and J. O. Smith, "A bowed string physical model including finite-width thermal friction and hair dynamics," in *Proc. Int. Computer Music Conf./Sound Music Comput. Conf. (SMC/ICMC-2014)*, Athens, Greece, 09 2014, pp. 1305–1311.

[13] C. Desvages, *Physical modelling of the bowed string and applications to sound synthesis*, Ph.D. thesis, The University of Edinburgh, 2018.

[14] B. Wendroff, *Difference methods for initial-value problems (R. D. Richtmyer and K. W. Morton)*, John Wiley & Sons, Ltd, Chichester, UK, 1968.

[15] F. Fontana and E. Bozzo, "Newton–Raphson solution of nonlinear delay-free loop filter networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 10, pp. 1590–1600, 2019.

[16] M. Ducceschi, S. Bilbao, and C. J. Webb, "Non-iterative schemes for the simulation of nonlinear audio circuits," in *Proc. Digital Audio Effects (DAFx-20in21)*, Vienna, Austria, 09 2021, pp. 25–32.

[17] M. Ducceschi and S. Bilbao, "Non-iterative simulation methods for virtual analog modelling," *IEEE Trans. Audio Speech Lang. Process.*, Under Review.

[18] A. van der Schaft and D. Jeltsema, "Port-hamiltonian systems theory: an introductory overview," *Found. and Trends in Syst. and Control*, vol. 1, no. 2,3, 2014.

[19] A. Falaize and T. Hélie, "Passive guaranteed simulation of analog audio circuits: a Port-Hamiltonian approach," *Appl. Sc.*, vol. 6, pp. 273 – 273, 2016.

[20] R. Müller, , and T. Hélie, "Power-balanced modelling of circuits as skew gradient systems," in *Proc. Digital Audio Effects (DAFx-19)*, Aveiro, Portugal, 09 2018.

[21] R.J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations. Steady State and Time Dependent Problems*, SIAM, Philadelphia, USA, 2007.

[22] T. Itoh and K. Abe, "Hamiltonian-conserving discrete canonical equations based on variational difference quotients," *J. Comput. Phys.*, vol. 76, no. 1, pp. 85–102, 1988.

[23] V. Chatziioannou and M. van Walstijn, "Energy conserving schemes for the simulation of musical instrument contact dynamics," *J. Sound Vib.*, vol. 339, pp. 262–279, 03 2015.

[24] 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.

[25] J. R. Bunch and J. E. Hopcroft, "Triangular factorization and inversion by fast matrix multiplication," *Math. Comput.*, vol. 28, no. 125, pp. 231–236, 1974.

[26] J. C. Schelleng, "The physics of the bowed string," *Sci. Am.*, vol. 230, no. 1, pp. 87 – 95, 1974.

[27] Michele Ducceschi and Stefan Bilbao, "Linear stiff string vibrations in musical acoustics: Assessment and comparison of models," *J. Acoust. Soc. Am.*, vol. 140, no. 4, pp. 2445–2454, 2016.

[28] M. Ducceschi and S. Bilbao, "Conservative finite difference time domain schemes for the prestressed timoshenko, shear and euler-bernoulli beam equations," *Wave Motion*, vol. 89, pp. 142–165, 2019.

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