

REAL-TIME IMPLEMENTATION OF A PHYSICAL MODEL OF THE TROMBA MARINA

Silvin Willemsen and Stefania Serafin
Multisensory Experience Lab, CREATE
Aalborg University Copenhagen
{sil, sts}@create.aau.dk

Stefan Bilbao and Michele Ducceschi
Acoustics and Audio Group
University of Edinburgh
{s.bilbao, michele.ducceschi}@ed.ac.uk

ABSTRACT

You can place comments in colour if you want :) Like [this](#) ([Stefan](#)), [this](#) ([Michele](#)) or [this](#) ([Stefania](#)).

1. INTRODUCTION

The tromba marina (see Figure ??) is a medieval monochord bowed instrument with a long quasi-trapezoidal body and a uniquely fashioned bridge (often called a shoe, because of its shape see Figure ??). The name of the instrument derives from the fact that *tromba* means *trumpet* in Italian. As a matter of fact, a foot of the bridge is free to rattle against the soundboard of the instrument in sympathy with the vibrating string. This unusual bridge creates a trumpet-like sound. The frequency produced by the instrument is varied by placing the side of the knuckle of the non-dominant hand, lightly, at specific nodal points on the string, in order to select various harmonics of the open string. The dominant hand controls the bow, which is drawn across the string above the non-dominant hand [?].

The tromba marina is therefore a rather unusual instrument with an interesting acoustics which is worth investigating using physical models. One of the ultimate goals is the emulation of an instrument that, due to its rarity is not accessible to a large audience.

In this paper, we present a real-time implementation of a physical model of the tromba marina.

Physical modelling for sound synthesis has a long history. Various techniques have been developed to simulate real-world instruments, including mass-spring systems [?, ?, ?], digital waveguides [?] and modal synthesis [?]. [can probably use just one of the Cadoz refs](#)

Finite-difference time-domain (FDTD) methods were first used for sound synthesis by Hiller and Ruiz in [?, ?, ?], and later by Chaigne et al. in [?, ?] and elaborated upon by Bilbao and colleagues in [?, ?]. Compared with other techniques, FDTD methods are more computationally expensive, but easily generalisable and flexible—no assumptions of linearity of travelling wave solutions are employed. Our goal is to implement these accurate techniques in real-time and thereby make the simulations playable for the users.



Figure 1. The tromba marina from the Danish Music Museum in Copenhagen.

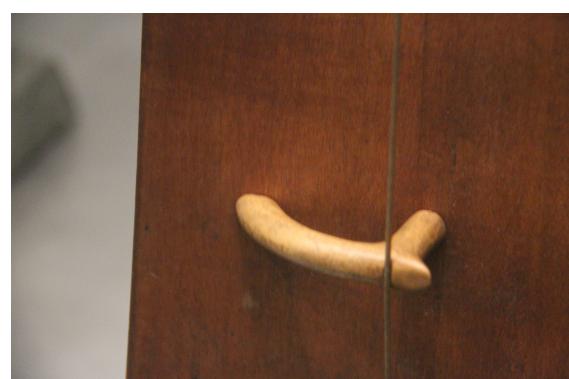


Figure 2. The bridge of the tromba marina from the Danish Music Museum in Copenhagen.

The emulation of nonlinear collision interactions in musical instruments normally requires the use of iterative solvers (such as, e.g. Newton-Raphson) [?]. For the non-linear collisions present in the instrument, a method recently proposed in the field of audio by Ducceschi and Bilbao in [?] allows such iterative methods to be sidestepped. It is thus suited to creating a real-time implementation of the tromba marina.

This paper is structured as follows: Section ?? presents the models used, Section ?? shows

2. MODELS

The tromba marina can be subdivided into three main components: the string, the bridge and the body. In this section, the PDEs of the different components in isolation will be given in the form

$$\mathcal{L}u = 0, \quad (1)$$

with linear (partial) differential operator \mathcal{L} and $u = u(\mathbf{x}, t)$ describes the component state over time t and space $\mathbf{x} \in \mathcal{D}$, where the dimensions of domain \mathcal{D} depend on the component at hand. As different models share variable names, subscripts ‘s’, ‘m’ and ‘p’ are added to denote that, among others, the variable u , domain \mathcal{D} or operator \mathcal{L} applies to the string, bridge (mass) or body (plate), respectively.

2.1 Bowed Stiff string

Consider a damped stiff string of length L (m), with domain $\mathcal{D} = \mathcal{D}_s = [0, L]$ and state variable $u = u_s(x, t)$. With reference to (??), we define the operator $\mathcal{L} = \mathcal{L}_s$ as

$$\mathcal{L}_s = \rho_s A \partial_t^2 - T \partial_x^2 + E_s I \partial_x^4 + 2\rho_s A \sigma_{0,s} \partial_t - 2\rho_s A \sigma_{1,s} \partial_t \partial_x^2, \quad (2)$$

Here, ∂_t and ∂_x indicate partial differentiation with respect to t and x . The various parameters appear as: material density ρ_s ($\text{kg}\cdot\text{m}^{-3}$), cross-sectional area $A = \pi r^2$ (m^2), radius r (m), tension $T = (2f_{0,s}L)^2 \rho_s A$ (N),¹ fundamental frequency $f_{0,s}$ (s^{-1}), Young’s modulus E_s (Pa), moment of inertia $I = \pi r^4/4$ (m^4), and loss coefficients $\sigma_{0,s}$ (s^{-1}) and $\sigma_{1,s}$ (m^2/s). We set the boundary conditions to be simply supported so that

$$u_s = \partial_x^2 u_s = 0, \quad \text{where } x = 0, L. \quad (3)$$

In the DAFx paper from last year we wrote $x = 0, L$ instead of $x = [0, L]$, as – in this case – we don’t want to specify a domain but single locations instead. Which is correct? As the string is excited using a bow, Equation (??) may be augmented as

$$\mathcal{L}_s u_s = -\delta(x - x_b) F_b \Phi(v_{\text{rel}}), \quad (4)$$

with externally supplied downward bow force $F_b = F_b(t)$ (N), spatial Dirac delta function $\delta(x - x_b)$ (m) selecting the bow position $x_b = x_b(t) \in \mathcal{D}_s$ (m), dimensionless friction characteristic

$$\Phi(v_{\text{rel}}) = \sqrt{2a} v_{\text{rel}} e^{-a v_{\text{rel}}^2 + 1/2}, \quad (5)$$

¹ Even though this definition for T from the fundamental frequency $f_{0,s}$ is only valid for a simply supported string without stiffness, the effect of the stiffness we eventually choose on $f_{0,s}$ is negligible.

with free parameter a . The relative velocity between the string at bow location x_b and the externally supplied bow velocity $v_b = v_b(t)$ (m/s) is defined as

$$v_{\text{rel}} = \partial_t u_s(x_b, t) - v_b. \quad (6)$$

2.2 Bridge

The bridge is modelled as a simple mass with a linear restoring force. As this system is point-like, the state variable $u = u_m(t)$ and the definition of domain \mathcal{D} is unnecessary. The operator $\mathcal{L} = \mathcal{L}_m$ is defined as

$$\mathcal{L}_m = M \frac{d^2}{dt^2} + M \omega_0^2 + M R \frac{d}{dt}, \quad (7)$$

with mass M (kg), linear angular frequency of oscillation $\omega_0 = 2\pi f_{0,m}$ (s^{-1}), fundamental frequency $f_{0,m}$ (s^{-1}) and damping coefficient R (s^{-1}).

2.3 Body

The body is simplified to a two-dimensional plate with side-lengths L_x and L_y , domain $\mathcal{D} = \mathcal{D}_p = [0, L_x] \times [0, L_y]$ and state variable $u = u_p(x, y, t)$. Using the 2D Laplacian

$$\Delta \triangleq \partial_x^2 + \partial_y^2, \quad (8)$$

the operator $\mathcal{L} = \mathcal{L}_p$ can be defined as

$$\mathcal{L}_p = \rho_p H \partial_t^2 + D \Delta \Delta + 2\rho_p H \sigma_{0,p} \partial_t - 2\rho_p H \sigma_{1,p} \partial_t \Delta, \quad (9)$$

with material density ρ_p ($\text{kg}\cdot\text{m}^{-3}$), plate thickness H (m), stiffness coefficient $D = E_p H^3 / 12(1-\nu^2)$, Young’s modulus E_p (Pa), dimensionless Poisson’s ratio ν , and loss coefficients $\sigma_{0,p}$ (s^{-1}) and $\sigma_{1,p}$ (m^2/s). The boundary conditions of the plate are set to be clamped so that

$$u_p = \mathbf{n} \cdot \nabla u_p = 0. \quad (10)$$

where ∇u_p is the gradient of u_p , and where \mathbf{n} indicates a normal to the plate boundary.

2.4 Collisions

It can be argued that the greatest contributor to the characteristic sound of the tromba marina is the rattling bridge colliding with the body. A collision can be modelled by including a term to the PDEs described above describing the potential energy of the system (further referred to as *the potential*) [?]. For the bridge-body interaction this potential is defined as follows

$$\phi_{\text{mp}}(\eta_{\text{mp}}) = \frac{K_{\text{mp}}}{\alpha_{\text{mp}} + 1} [\eta_{\text{mp}}]_+^{\alpha_{\text{mp}} + 1}, \quad (11)$$

$$K_{\text{mp}} > 0, \quad \alpha_{\text{mp}} \geq 1, \quad \eta_{\text{mp}} \triangleq u_p(x_{\text{mp}}, y_{\text{mp}}, t) - u_m(t)$$

where K_{mp} is the collision stiffness (N/m if $\alpha_{\text{mp}} = 1$), α_{mp} is the dimensionless non-linear collision coefficient, and $\eta_{\text{mp}} = \eta_{\text{mp}}(t)$ is the distance between the rattling part of the bridge and the body at the point of collision (m). Furthermore, and $[\eta_{\text{mp}}]_+ = 0.5(\eta_{\text{mp}} + |\eta_{\text{mp}}|)$ is the positive part of η_{mp} . The term which can then be included in the PDEs is $\phi'_{\text{mp}} = d\phi_{\text{mp}} / d\eta_{\text{mp}}$. As described in [?, ?, ?, ?], using this

form of the potential requires using iterative methods for solving its discrete counterpart. The authors proposed to rewrite the potential to

$$\psi = \sqrt{2\phi}, \quad (12)$$

and the term included in the PDEs to

$$\phi' = \psi\psi' = \psi \frac{d\psi}{d\eta} \xrightarrow{\text{chain rule}} \psi \frac{\dot{\psi}}{\dot{\eta}} \quad (13)$$

where the dot operator denotes a derivative with respect to time. Equation (??), as can be seen in Section ??, can be explicitly calculated.

As the string rests on the bridge, the interaction between these components needs to be modelled as well. Even though the bridge-body interaction is perpendicular to the string-bridge interaction, we can model them as being parallel, assuming that a “horizontal” movement of the string causes a “vertical movement” of the rattling part of the bridge. We can use an alternative version of the potential in Equation (??) described in [?] to make the collision two-sided acting as a connection:

$$\phi_{sm}(\eta_{sm}) = \frac{K_{sm}}{\alpha_{sm} + 1} |\eta_{sm}|^{\alpha_{sm}+1}, \quad (14)$$

$$K_{sm} > 0, \quad \alpha_{sm} \geq 1, \quad \eta_{sm} \triangleq u_m(t) - u_s(x_{sm}, t)$$

where $\eta_{sm} = \eta_{sm}(t)$ is the distance between the string at the location of the bridge and the bridge itself. Including the effect of the bow from Equation (??), Equation (??) for all components can be rewritten to get the complete system:

$$\left\{ \begin{array}{l} \mathcal{L}_s u_s = -\delta(x - x_b) F_b \Phi(v_{rel}) \\ \quad + \delta(x - x_{sm}) \psi_{sm} \psi'_{sm} \end{array} \right. \quad (15a)$$

$$\left\{ \begin{array}{l} \mathcal{L}_m u_m = -\psi_{sm} \psi'_{sm} + \psi_{mp} \psi'_{mp}, \end{array} \right. \quad (15b)$$

$$\left\{ \begin{array}{l} \mathcal{L}_p u_p = -\delta(x - x_{mp}, y - y_{mp}) \psi_{mp} \psi'_{mp}, \end{array} \right. \quad (15c)$$

$$\left\{ \begin{array}{l} \eta_{sm} = u_m(t) - u_s(x_{sm}, t), \end{array} \right. \quad (15d)$$

$$\left\{ \begin{array}{l} \eta_{mp} = u_p(x_{mp}, y_{mp}, t) - u_m(t), \end{array} \right. \quad (15e)$$

where $x_{sm} \in \mathcal{D}_s$ is the location of the bridge along the string and $(x_{mp}, y_{mp}) \in \mathcal{D}_p$ is where the bridge collides with the body. Note that the same symbol x for is used for both the string and the body, but are completely independent as they apply to the domains of their respective model.

3. DISCRETISATION

The system found in the Equations in (??) is discretised using FDTD methods to obtain finite difference schemes (FDSs). This is the way this terminology is used correct? I always get confused when to use which.. These methods subdivide the continuous system in grid points in space and samples in time. To approximate the state of a system in isolation we use

$$u(\mathbf{x}, t) \approx u_l^n, \quad (16)$$

where grid function u_l^n is a discrete version of $u(\mathbf{x}, t)$ sampled at $t = nk$ with time step k (s), sample $n \in \mathbb{N}$ and location \mathbf{l} that depends on domain \mathcal{D} of the system at hand.

In the case of the string, we use $x = lh_s$ with grid spacing h_s (m), $\mathbf{l} = l \in [0, \dots, N - 1]$ and total number of grid points $N = L/h_s$ to yield $u_s(x, t) \approx u_{l,s}^n$.

As the bridge is point-like, a definition of \mathbf{l} is unnecessary and $u_m(t) \approx u_m^n$.

In the case of the body, we use $x = lh_p$ and $y = mh_p$ to get $u_p(x, y, t) \approx u_{(l,m),p}^n$ where $\mathbf{l} = (l, m)$ with $l \in [0, \dots, N_x - 1]$ and $m \in [0, \dots, N_y - 1]$. Here, $N_x = L_x/h_p$ and $N_y = L_y/h_p$ are the horizontal and vertical number of grid points respectively.

The discretisation of and expansion of operator $\mathcal{L} \approx \ell$ in the case of stiff strings, mass-spring systems and plates using FDTD methods are well covered in the literature [?]. To obtain the highest accuracy possible while keeping the system explicit (except for the bow), centered differences have been chosen where possible (see Equation (??)).

For stability, grid spacings h should satisfy the conditions below. In the case of the damped stiff string,

$$h_s \geq \sqrt{\frac{c^2 k^2 + 4\sigma_1 k + \sqrt{(c^2 k^2 + 4\sigma_1 k)^2 + 16\kappa_s^2 k^2}}{2}}, \quad (17)$$

with wave speed $c = \sqrt{T/\rho_s A}$ and stiffness coefficient $\kappa_s = \sqrt{E_s I/\rho_s H}$ and in the case of the plate,

$$h_p \geq 2 \sqrt{k \left(\sigma_1 + \sqrt{\kappa_p^2 + \sigma_1^2} \right)}, \quad (18)$$

with stiffness coefficient $\kappa_p = \sqrt{D/\rho_p H}$. The closer the grid spacings are to these conditions, the higher the accuracy of the approximation.

The bowing term in Equation (??) is discretised as follows

$$\ell_s u_{l,s}^n = -J_3(x_b) F_b^n \phi(v_{rel}^n) \quad (19)$$

where, using the centered difference operator presented below in Equation (??),

$$v_{rel}^n = \delta_t u_{l,b}^n - v_b^n, \quad (20)$$

which needs to be calculated using iterative methods.

3.1 Collisions using non-iterative methods

Before going into discretising the collision and connection terms in the system described in (??), some finite difference operators are introduced.

3.1.1 Operators

The identity and temporal shift operators are defined as

$$1\eta^n = \eta^n, \quad e_{t+}\eta^n = \eta^{n+1}, \quad e_{t-}\eta^n = \eta^{n-1}. \quad (21)$$

Using these, the operators for the forward, backward and centered time differences can be defined as

$$\delta_{t+} = \frac{e_{t+} - 1}{k}, \quad \delta_{t-} = \frac{1 - e_{t-}}{k}, \quad \delta_t = \frac{e_{t+} - e_{t-}}{2k}, \quad (22)$$

and are all approximations to a first-order time derivative. Furthermore, forwards and backwards averaging operators are defined as

$$\mu_{t+} = \frac{e_{t+} + 1}{2}, \quad \mu_{t-} = \frac{1 + e_{t-}}{2}. \quad (23)$$

and can be used to describe interleaved grid points $n + 1/2$ and $n - 1/2$.

3.1.2 Explicit collisions

For the discrete-time definitions of the potential in (??) we can use

$$\psi \approx \mu_{t+} \psi^{n-1/2} \quad \text{and} \quad \psi' \approx \frac{\delta_{t+} \psi^{n-1/2}}{\delta_t \cdot \eta^n}, \quad (24)$$

where ψ at interleaved grid point $n - 1/2$ is defined as

$$\psi^{n-1/2} = \mu_{t-} \psi^n. \quad (25)$$

For a system that has a single (upward) collision we get

$$\ell u_l^n = J(\mathbf{x}_c) \left(\mu_{t+} \psi^{n-1/2} \right) \frac{\delta_{t+} \psi^{n-1/2}}{\delta_t \cdot \eta^n}, \quad (26)$$

where $J(\mathbf{x}_c)$ is a spreading operator applying the collision force to coordinate \mathbf{x}_c , which, in the simplest case, is defined as [?]

$$J(\mathbf{x}_c) = \begin{cases} \frac{1}{h^d}, & l = l_c = \text{round}(\mathbf{x}_c/h) \\ 0, & \text{otherwise.} \end{cases} \quad (27)$$

Here, d is the number of dimensions of domain \mathcal{D} that \mathbf{x} is defined for, i.e., $d = 0$ for the bridge, $d = 1$ for the string, and $d = 2$ for the plate.

Continuing, we use the identity

$$\mu_{t+} \psi^{n-1/2} = \frac{k}{2} \delta_{t+} \psi^{n-1/2} - \psi^{n-1/2} \quad (28)$$

and define

$$g^n = \frac{\delta_{t+} \psi^{n-1/2}}{\delta_t \cdot \eta^n} \quad (29)$$

which can be rewritten to

$$\delta_{t+} \psi^{n-1/2} = g^n \delta_t \cdot \eta^n. \quad (30)$$

Then, inserting (??) into (??) and this together with (??) into (??) we get

$$\ell u_l^n = J(\mathbf{x}_c) \left(\frac{k}{2} g^n \delta_t \cdot \eta^n - \psi^{n-1/2} \right) g^n \quad (31)$$

where g^n may be explicitly calculated using the analytic expressions for ψ and ϕ [?]:

$$g^n = \psi' \Big|_{\eta=\eta^n} = \frac{\phi'}{\sqrt{2\phi}} \Big|_{\eta=\eta^n}. \quad (32)$$

Numerical stability of this scheme is shown in [?]. When writing out (??) we can obtain definitions for g_{sm}^n using (??)

$$g_{sm}^n = \text{sgn}(\eta_{sm}^n) \sqrt{\frac{K_{sm}(\alpha_{sm} + 1)}{2}} |\eta_{sm}^n|^{\frac{\alpha_{sm}-1}{2}}, \quad (33)$$

and g_{mp}^n using (??)

$$g_{mp}^n = \sqrt{\frac{K_{mp}(\alpha_{mp} + 1)}{2}} [\eta_{mp}^n]_+^{\frac{\alpha_{mp}-1}{2}}. \quad (34)$$

Lastly, introducing for brevity,

$$q^n = \frac{k}{2} g^n \delta_t \cdot \eta^n - \psi^{n-1/2}, \quad (35)$$

the discrete counterpart of the complete system described in Equations (??) will be

$$\begin{cases} \ell_s u_{l,s}^n &= -J_3(x_b^n) F_b \Phi(v_{rel}^n) + J(x_{sm}) q_{sm}^n g_{sm}^n, \\ \ell_m u_m^n &= -q_{sm}^n g_{sm}^n + q_{mp}^n g_{mp}^n, \end{cases} \quad (36a)$$

$$\begin{cases} \ell_p u_{(l,m),p}^n &= -J(x_{mp}, y_{mp}) q_{mp}^n g_{mp}^n, \\ \eta_{sm}^n &= u_m^n - u_{l,sm,s}^n, \\ \eta_{mp}^n &= u_{(x_{mp}, y_{mp}),p}^n - u_m^n, \end{cases} \quad (36c)$$

$$\begin{cases} \eta_{sm}^n &= u_m^n - u_{l,sm,s}^n, \\ \eta_{mp}^n &= u_{(x_{mp}, y_{mp}),p}^n - u_m^n, \end{cases} \quad (36e)$$

where g_{sm}^n and g_{mp}^n can be explicitly computed according to Equation (??). Furthermore, the spreading operator for the bow J_3 uses cubic interpolation for finer control [?]. This leaves us with two different types of update equations, one where u_l^{n+1} is calculated and one where $\psi^{n+1/2}$ is calculated.

One might think that due to the centered differences $\delta_t \cdot \eta^n$ still present in Equation (??), our system remains implicit, but as we can insert the definitions for Equations (??) and (??) evaluated at the next sample $n + 1$ which are already present in ℓu , the Equations in (??) reduce to a system of linear equations solved by a single division. Does this make sense? I thought to add this paragraph to emphasise that the system is now actually explicit.

4. IMPLEMENTATION

The real-time implementation of the system has been done in C++ using the JUCE framework [?] and will be controlled using the Sensel Morph (or simply Sensel) – an expressive touch controller. This section will first elaborate some important considerations regarding the setup of the system. Then, the algorithm, together with the parameter design will be presented. Finally, the graphical user interface (GUI) will be detailed together with the Sensel and its mapping to the application.

4.1 Introducing an Offset

Firstly, for more realistic and expressive sounds, we model the bridge – and with that, the string – to rest slightly above the body. Expanding $\ell_m u_m$ in (??) and including the offset yields

$$\ell_m u_m^n = M \delta_{tt} u_m^n + M \omega_0^2 (u^n - u_{off}) + M R \delta_t u_m^n \quad (37)$$

where $u_{off} \geq 0$ is a predefined offset between the body and the bridge. Furthermore, the second-order time derivative can be defined from the definitions in (??) as

$$\delta_{tt} = \delta_{t+} \delta_{t-}. \quad (38)$$

The boundary condition of the string will also change to:

$$u_s = u_{off} \quad \text{and} \quad \partial_x^2 u_s = 0. \quad (39)$$

A schematic plot of the full system, including the offset described above (Eq. (??)) can be found in Figure ??.

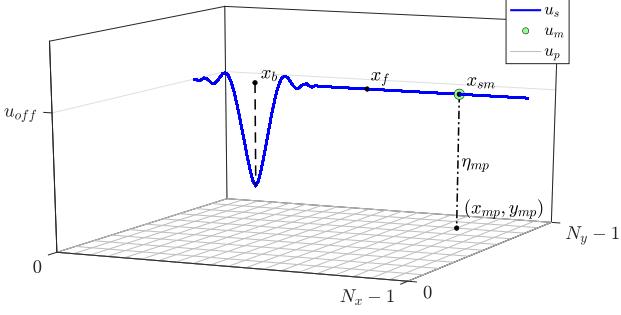


Figure 3. The virtual system in Equations (??) including the offset in Eq. (??), with different important coordinates highlighted. Note that η_{sm} (Eq. (??)) is not shown as it is close to 0 at all times.

4.2 Pitch Control

Secondly, as briefly mentioned in Section ??, the way different pitches are played on the tromba marina, is to slightly rest a knuckle or finger on nodal points along the string to induce harmonics. Thus, a damping finger is implemented. Using the cubic interpolation operator I_3 [?], the FDS for the string in (??) can be extended to

$$\ell_s u_{l,s}^n = \dots - J_3(x_f) I_3(x_f) \sigma_f (u_l^n - u_{off}), \quad (40)$$

where $0 \leq \sigma_f \leq 1,$

which essentially subtracts its own state at location $x_f \in [0, 0.5x_{sm}]$ according to the damping coefficient $\sigma_f (\text{kg} \cdot \text{s}^{-2})$ applied. As done in [?], the fractional part used in the spreading operator ($\alpha_i = x_f/h - \text{floor}(x_f/h)$) is raised to the 7th power as it has been empirically found to scale finger position to pitch more properly in the context of FDSs. As the string is bowed above the damping finger (at the other side of the rattling bridge) it is essential that the energy from the bow reaches the rattling bridge. For lower values for σ_f , this is still the case.

4.3 Other Considerations

Realistic initialisation both η_{sm} and η_{mp} is essential. In this case (at $n = 0$) $\eta_{sm}^0 = 0$ and $\eta_{mp}^0 \leq 0$ so that no collision is present at initialisation. If this is not done, the value for the potential at the previous time step $\psi^{n-1/2}$ would not make sense (**perhaps elaborate**).

After h_p is calculated in Equation (??), we check whether it is smaller than a set value $h_{p,\min} = 0.01$. This reduces the quality of the model, but increases the speed, allowing for real-time implementation.

4.4 Order of calculation

The order of calculation is shown in the pseudocode in Algorithm ???. In theory, in order to iteratively calculate the bow force, the collision and connection forces should be included in this. **Hmm.. but the rest of the system is explicit. Shouldn't the system be calculated in its entirety before the NR calculation?** However, as the string is practically never bowed at the bridge position x_{sm} , these can be calculated independently.

while application is running do

1. calculate schemes
2. apply bow to string
3. apply damping finger
4. calculate g_{sm}^n and g_{mp}^n
5. calculate collision and connection forces and add to schemes
6. Update states

(ℓu in Eqs. (??-c))

(Eq. (??))

(Eq. (??))

(Eqs. (??) and (??))

(Eqs. (??-c))

$$u^{n-1} = u^n$$

$$u^n = u^{n+1}$$

$$\psi^{n-1/2} = \psi^{n+1/2}$$

end

Algorithm 1: Pseudocode showing the order of calculations after initialisation.

4.5 Parameter design

The list of parameters used can be found in Table ???. As the authors had a real (recreated) tromba marina (presented in [?]) at their disposal, some parameters have been measured in accordance to the real instrument.

Regarding the output of the system, through empirical testing it was decided to retrieve the output from the state of the plate right at the point of collision (x_{mp}, y_{mp}) combined with the sound of the string at a lower volume. It can be argued that the loudest sound comes from the collision between the bridge and the body making it logical to select this point as the output position.

4.6 Graphical User Interface

A screenshot of the GUI is shown in Figure ???. The GUI is divided in four sections, three showing the states of the string, bridge and body respectively and one control section.

Firstly, the string section shows the state of the string $u_{l,s}^n$ as a cyan-coloured path and the bow as a yellow rectangle with bow position x_b and its opacity depending on the bow force F_b . Furthermore, the bridge state u_m^n is shown as a green circle at location (of the bridge along the string) x_{sm} . Finally, the position of the damping finger x_f is displayed as a yellow circle, the size of which depends on damping coefficient σ_f . The position of the finger triggers lines

Name	Symbol (unit)	Value
String		
Length	L (m)	1.90*
Material density	ρ_s ($\text{kg}\cdot\text{m}^{-3}$)	7850
Radius	r (m)	0.0005
Fundamental freq.	f_0 (s^{-1})	32.0*
Young's modulus	E_s (Pa)	$2 \cdot 10^{11}$
Freq. indep. loss	$\sigma_{0,s}$ (s^{-1})	0.1
Freq. dep. loss	$\sigma_{1,s}$ (m^2/s)	0.05
Bow		
Bow force	F_b (N)	$0 \leq F_b \leq 0.1$
Bow velocity	v_b (m/s)	$-0.5 \leq v_b \leq 0.5$
Free parameter	a (-)	100
Bridge		
Mass	M (kg)	0.001
Fundamental freq.	$f_{0,m}$ (s^{-1})	500
Damping	R (s^{-1})	0.05
Body		
Length	L_x (m)	1.35*
Width	L_y (m)	0.18*
Material density	ρ_p ($\text{kg}\cdot\text{m}^{-3}$)	50
Thickness (m)	H (m)	0.01
Young's modulus	E_p (Pa)	$2 \cdot 10^5$
Poisson's ratio	ν (-)	0.3
Freq. indep. loss	$\sigma_{0,p}$ (s^{-1})	2
Freq. dep. loss	$\sigma_{1,p}$ (m^2/s)	0.05
String-bridge connection		
Stiffness Coefficient	K_{sm} (N/m)	$5 \cdot 10^6$
Non-lin. col. coeff.	α_{sm} (-)	1.0
Bridge location	x_{sm} (m)	1.65*
Bridge-body collision		
Stiffness Coefficient	K_{mp} (N/m)	$5 \cdot 10^8$
Non-lin. col. coeff.	α_{mp} (-)	1.0
Bridge location	(x_{mp}, y_{mp}) (m,m)	(1.08, 0.135)*
Other		
Offset	u_{off} (m)	$5 \cdot 10^{-6}$
Damp. finger coeff.	σ_f ($\text{kg}\cdot\text{s}^{-2}$)	$0 \leq \sigma_f \leq 1$

Table 1. List of parameter values used for the simulation. *These values have been taken from a real (recreated) tromba marina [?].

showing the locations of the closest nodes along the string according to the following equation

$$x_{node}^i = \frac{i \cdot x_{sm}}{n} \quad \text{for } i = [1, \dots, n-1], \quad (41)$$

where $n = \text{round}(x_{sm}/x_f)$ is an integer closest to the ratio between the string length until the bridge location and the damping finger position. These lines are drawn to help the user place the damping finger at nodes along the string.

Secondly, the bridge section shows the displacement of the bridge u_m^n as a green circle, the state of the body at the collision location $u_{(x_{sm}, y_{sm}), p}^n$, both moving vertically according to their respective displacements and finally, a grey horizontal line denoting the offset u_{off} , i.e., the resting position of the bridge.

Thirdly, the body section shows the state of the body as a grid of grey-scale rectangles changing colour according to their displacement (white for positive and black for negative).

Finally, the control section contains three sliders that control the levels of the string (s), bridge (m) and body (p)

respectively and a reset button to re-initialise the system.

2020 paper template LaTeX/applicationDescription.png

applicationDescription.png

Figure 4. The GUI showing the excited system with text highlighting different components. A detailed description of this figure can be found in Section ??.

4.7 Sensel Morph and Mapping

The Sensel is an expressive touch controller using ~20,000 pressure sensitive sensors laid out in an hexagonal grid [?]. It retrieves x and y-positions and pressure at a rate of 150 Hz from which velocities and accelerations can be obtained.

The first finger registered by the Sensel is mapped to the bow: x-position is mapped to bow position x_b , y-velocity to bow velocity v_b (y-position is shown in the GUI but does not influence the model directly) and pressure to bow force F_b . The second finger is mapped to the damping finger: x-position is mapped to finger location x_f and pressure to damping coefficient σ_f .

5. DISCUSSION

As can be seen from Table ??, the non-linear collision coefficients α_{sm} and α_{mp} are set to 1. We found some odd behaviour when increasing this value, where the bridge would ‘get stuck’ behind the plate, i.e., values for ψ_{mp} would be negative for a prolonged period of time. The explicit technique used in this work allows for this to happen (and can be proven to still be stable in this case [?]), but it is unphysical to have a negative potential. Further investigation will be necessary to figure out a way to prevent this behaviour.

6. CONCLUSIONS

Comparison between NR and Non-iterative methods (both speed and quality)

Parameter design

For a more physical implementation of the damping finger, it might be a good idea to model it as another mass colliding with the string rather than imposing a ‘hacky’ way of damping onto the string state.

Acknowledgments

This work is supported by NordForsk’s Nordic University Hub Nordic Sound and Music Computing Network Nordic-SMC, project number 86892. Ducceschi’s work was supported by an Early Career Fellowship from the Leverhulme Trust.