Abstract

In this paper, a complete simulation of a trombone using finite-difference time-domain (FDTD) methods is proposed. In particular, we propose the use of a novel method to dynamically vary the number of grid points associated to the FDTD method, to simulate the fact that the physical dimension of the trombone's resonator dynamically varies over time. We describe the different elements of the model and present the results of a real-time simulation.

1   Introduction

The trombone is a musical instrument that presents distinct challenges from the perspective of physical modelling synthesis. In particular, the excitation mechanism between the lips and the player has been extensively studied, and simulated mostly using a simple mass-spring damper system [1]. Because the majority of the bore is cylindrical, nonlinear effects can appear at high blowing pressures [2], leading to changes in timbre, or brassiness; such effects have been investigated and simulated [1, 3, 4]. However, the defining characteristic of the trombone is that the physical dimensions of the resonator vary during playing. Synthesis techniques such as digital waveguides allow an approach to dynamic resonator changes in a simple and computationally efficient way, simply by varying the length of the corresponding delay line. This feature has been used in real-time sound synthesis [5], for simplified bore profiles suitable for modelling in terms of travelling waves.

However, when attempting more fine-grained modelling of the trombone resonator using finite-difference time-domain (FDTD) methods, the issue of the change in the tube length is not trivial. Previous implementations of brass instruments using these methods focus on the trumpet [6] and various brass instruments (including the trombone bore) under static conditions [7]. To our knowledge, the simulation of a trombone varying the shape of the resonator in real time using FDTD methods has not been approached. We can tackle this problem by having a grid that dynamically changes while the simulation is running as presented in a companion paper [8]. Briefly described, we modify the grid configurations of the FDTD method by adding and removing grid points based on parameters describing the system.

In this paper, we propose a full simulation of a trombone, describing all its elements in detail with a specific focus on the dynamic grid simulation. Section 2 presents the models for the tube and lip reed interaction in continuous time. Section 3 briefly introduces FDTD methods and the discretisation of the aforementioned continuous equations. Section 4 presents the dynamic grid used to simulate the trombone slide and details on the implementation are provided in Section 5. Section 6 presents simulation results, and some

concluding remarks appear in Section 7.

2 Continuous System

Wave propagation in an acoustic tube can be approximated using a 1-dimensional (1D) model, for wavelengths that are long relative to the largest lateral dimension of the tube. Consider a tube of time-varying length $L = L(t)$ (in m) defined over spatial domain $x \in [0, L]$ and time $t \geq 0$. Using operators ∂_t and ∂_x denoting partial derivatives with respect to time t and spatial coordinate x, respectively, a system of first-order partial differential equations (PDEs) describing the wave propagation in an acoustic tube can then be written as:

$$\frac{S}{\rho_0 c^2} \partial_t p = -\partial_x(Sv), \tag{1a}$$

$$\rho_0 \partial_t v = -\partial_x p, \tag{1b}$$

with acoustic pressure $p = p(x, t)$ (in N/m^2), particle velocity $v = v(x, t)$ (in m/s) and (circular) cross-sectional area $S(x)$ (in m^2). Furthermore, ρ_0 is the density of air (in kg/m^3) and c is the speed of sound in air (in m/s). System (1) can be condensed into a second-order equation in p alone, often referred to as Webster's equation [9]. For simplicity, effects of viscothermal losses have been neglected in (1). For a full time domain model of such effects in an acoustic tube, see, e.g. [10].

System (1) requires two boundary conditions, one at either end of the domain. The left boundary condition, at $x = 0$, will be set according to an excitation model to be described in Section 2.1. The right boundary, at $x = L$, is set according to a radiation condition. The radiation model used here, is the one for the unflanged cylindrical pipe proposed by Levine and Schwinger in [11] and discretised by Silva *et al.* in [12]. As this model is not important for the contribution of this work it will not be detailed here in full. The interested reader is instead referred to [7, 13] for a comprehensive explanation.

2.1 Coupling to a Lip Reed

To excite the system, a lip reed can be modelled as a mass-spring-damper system including two nonlinearities due to flow, and the collision of the lip against the mouthpiece. In the following, y can be seen as the moving upper lip where the lower lip is left static and rigid. A diagram of the full lip-reed model is shown in Figure 1. Using dots to indicate time-derivatives, the lip reed is modelled as

$$M_{\mathrm{r}} \ddot{y} = -M_{\mathrm{r}} \omega_{\mathrm{r}}^2 y - M_{\mathrm{r}} \sigma_{\mathrm{r}} \dot{y} + \psi(\dot{\psi}/\dot{\eta}) + S_{\mathrm{r}} \Delta p, \tag{2}$$

with displacement from the equilibrium $y = y(t)$, lip mass M_r (in kg), externally supplied (angular) frequency of oscillation $\omega_r = \omega_r(t) = \sqrt{K_r/M_r}$ (in rad/s) and stiffness $K_r = K_r(t)$ (in N/m).

We extend the existing models of lip reeds [1] by introducing a nonlinear collision between the lips based on potential quadratisation proposed by [14]. The collision potential is defined as

$$\psi(\eta) = \left(\frac{2K_c}{\alpha_c + 1} [\eta]_+^{\alpha_c + 1} \right)^{1/2}, \tag{3}$$

with collision stiffness $K_c > 0$ and dimensionless nonlinear collision coefficient $\alpha_c \geq 1$, The inverted distance between the lips $\eta = \eta(t) \triangleq -y - H_0$ (in m), for static equilibrium separation H_0 (in m). $[\eta]_+ = 0.5(\eta + |\eta|)$ indicates the "positive part of η". Notice, that if $\eta \geq 0$, the lips are closed and the collision potential will be non-zero. This quadratic form of a collision potential allows for a non-iterative implementation [14]. This will be explained further in Section 3.

Finally, S_r (in m^2) is the effective surface area and

$$\Delta p = P_m - p(0, t) \tag{4}$$

is the difference between the externally supplied pressure in the mouth $P_m = P_m(t)$ and the pressure in the mouthpiece $p(0, t)$ (all in Pa). This pressure difference causes a volume flow velocity following the Bernoulli equation

$$U_B = w_r[-\eta]_+ \text{sgn}(\Delta p) \sqrt{\frac{2|\Delta p|}{\rho_0}}, \tag{5}$$

(in m^3/s) with effective lip-reed width w_r (m). Another volume flow is generated by the lip reed itself according to

$$U_r = S_r \frac{dy}{dt} \tag{6}$$

(in m^3/s). Assuming that the volume flow velocity is conserved, the total air volume entering the system is defined as

$$S(0)v(0, t) = U_B(t) + U_r(t). \tag{7}$$

This condition serves as a boundary condition at $x = 0$ for system (1).
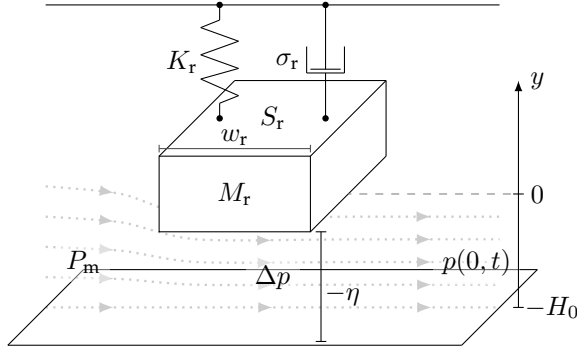
Fig. 1: *Diagram of the lip-reed system with the equilibrium at 0 and the distance from the lower lip H_0. The various symbols relate to those used in Eq. (2).*

3 Discretisation

The continuous system described in the previous section is discretised using FDTD methods, through an approximation over a grid in space and time. Before presenting this discretisation, we briefly summarize the operation of FDTD methods.

3.1 Numerical Methods

Consider a 1D system of (static) length L described by state variable $u = u(x, t)$ with spatial domain $x \in [0, L]$ and time $t \geq 0$. The spatial domain can be disctretised according to $x = lh$ with spatial index $l \in \{0, \ldots, N\}$, number of intervals between the grid points N, grid spacing h (in m) and time as $t = nk$ with temporal index $n \in \mathbb{Z}^{0+}$ and time step k (in s). The grid function u_l^n represents an approximation to $u(x, t)$ at $x = lh$ and $t = nk$.

Shift operators can then be applied to grid function u_l^n. Temporal and spatial shift operators are

$$
\begin{aligned}
e_{t+} u_l^n = u_l^{n+1}, \quad & e_{t-} u_l^n = u_l^{n-1}, \\
e_{x+} u_l^n = u_{l+1}^n , \quad & e_{x-} u_l^n = u_{l-1}^n,
\end{aligned}
\tag{8}
$$

from which more complex operators can be derived. First-order derivatives can be approximated using forward, backward and centred difference operators in time

$$
\delta_{t+} = \frac{e_{t+} - 1}{k}, \ \delta_{t-} = \frac{1 - e_{t-}}{k}, \ \delta_{t\cdot} = \frac{e_{t+} - e_{t-}}{2k},
\tag{9}
$$

(all approximating ∂_t) and space

$$\delta_{x+} = \frac{e_{x+} - 1}{h}, \ \delta_{x-} = \frac{1 - e_{x-}}{h}, \ \delta_{x\cdot} = \frac{e_{x+} - e_{x-}}{2h}, \tag{10}$$

(all approximating ∂_x) where 1 is the identity operator.

Furthermore, forward, backward and centred averaging operators can be defined in time

$$\mu_{t+} = \frac{e_{t+} + 1}{2}, \ \mu_{t-} = \frac{1 + e_{t-}}{2}, \ \mu_{t\cdot} = \frac{e_{t+} + e_{t-}}{2}, \tag{11}$$

and space

$$\mu_{x+} = \frac{e_{x+} + 1}{2}, \ \mu_{x-} = \frac{1 + e_{x-}}{2}, \ \mu_{x\cdot} = \frac{e_{x+} + e_{x-}}{2}. \tag{12}$$

Finally, an approximation δ_{tt} to a second time derivative may be defined as

$$\delta_{tt} = \delta_{t+}\delta_{t-} = \frac{1}{k^2}\left(e_{t+} - 2 + e_{t-}\right). \tag{13}$$

3.2 Discrete Tube

As a first step, the domain $x \in [0, L]$ can be subdivided into N equal segments of length h (the grid spacing). Interleaved grid functions approximating p and v may then be defined. Grid function p_l^n with $l \in \{0, \ldots, N\}$ approximates $p(x, t)$ at coordinates $x = lh$, $t = nk$ and $v_{l+1/2}^{n+1/2}$ with $l \in \{0 \ldots, N-1\}$ approximates $v(x, t)$ at coordinates $x = (l + 1/2)h$, $t = (n + 1/2)k$. In addition, a discrete cross-sectional area $S_l \approx S(x = lh)$ with $l \in \{0, \ldots, N\}$ is assumed known. System (1) can then be discretised as

$$\frac{\bar{S}_l}{\rho_0 c^2}\delta_{t+}p_l^n = -\delta_{x-}(S_{l+1/2}v_{l+1/2}^{n+1/2}), \tag{14a}$$

$$\rho_0\delta_{t-}v_{l+1/2}^{n+1/2} = -\delta_{x+}p_l^n, \tag{14b}$$

where $S_{l+1/2} = \mu_{x+}S_l$ and $\bar{S}_l = \mu_{x-}S_{l+1/2}$ are approximations to the continuous cross-sectional area $S(x)$. The values for $\bar{S}_l$ at the boundaries, i.e., $\bar{S}_0$ and $\bar{S}_N$, are set equal to $S(0)$ and $S(L)$.

Expanding the operators, we obtain the following recursion

$$p_l^{n+1} = p_l^n - \frac{\rho_0 c\lambda}{\bar{S}_l}(S_{l+1/2}v_{l+1/2}^{n+1/2} - S_{l-1/2}v_{l-1/2}^{n+1/2}), \tag{15a}$$

$$v_{l+1/2}^{n+1/2} = v_{l+1/2}^{n-1/2} - \frac{\lambda}{\rho_0 c}(p_{l+1}^n - p_l^n), \tag{15b}$$

where $\lambda = ck/h$ is referred to as the Courant number and

$$\lambda \leq 1 \quad \implies \quad h \geq ck \tag{16}$$

in order for the scheme to be stable [15]. In implementation, the following steps are taken to calculate λ:

$$h := ck, \;\; N := \left\lfloor \frac{L}{h} \right\rfloor, \;\; h := \frac{L}{N}, \;\; \lambda := \frac{ck}{h}, \tag{17}$$

where $\lfloor \cdot \rfloor$ denotes the flooring operation and is necessary because N is an integer. This causes (16) to not be satisfied with equality for all choices of L.

Equations (15a) and (15b) hold for $l \in \{0, \ldots, N\}$ and $l \in \{0, \ldots, N-1\}$ respectively, and thus, in analogy with the continuous case, two numerical boundary conditions are required in order to update p_0^{n+1} and p_N^{n+1}. These are provided by numerical equivalents of the excitation condition (see Section 3.3 below) and the radiation condition (in [13]).

3.3 Lip reed

As the lip reed interacts with the particle velocity of the tube via Eq. (7), it is discretised to the interleaved temporal grid, but to the regular spatial grid as it interacts with the boundary at $x = 0$. Equations (2) - (7) are then discretised as follows:

$$M_{\mathrm{r}} \delta_{tt} y^{n+1/2} = - M_{\mathrm{r}} (\omega_{\mathrm{r}}^{n+1/2})^2 \mu_{t \cdot} y^{n+1/2}$$
$$- M_{\mathrm{r}} \sigma_{\mathrm{r}} \delta_{t \cdot} y^{n+1/2} + (\mu_{t+} \psi^n) g^{n+1/2} + S_{\mathrm{r}} \Delta p^{n+1/2}, \tag{18a}$$

$$\Delta p^{n+1/2} = P_{\mathrm{m}}^{n+1/2} - \mu_{t+} p_0^n, \tag{18b}$$

$$U_{\mathrm{B}}^{n+1/2} = w_{\mathrm{r}} [-\eta^{n+1/2}]_+ \mathrm{sgn}(\Delta p^{n+1/2}) \cdot \sqrt{2|\Delta p^{n+1/2}|/\rho_0}, \tag{18c}$$

$$U_{\mathrm{r}}^{n+1/2} = S_{\mathrm{r}} \delta_{t \cdot} y^{n+1/2}, \tag{18d}$$

$$\mu_{x-}(S_{1/2} v_{1/2}^{n+1/2}) = U_{\mathrm{B}}^{n+1/2} + U_{\mathrm{r}}^{n+1/2}. \tag{18e}$$

Here, following [14],

$$g^{n+1/2} = \begin{cases} \kappa \sqrt{\dfrac{K_{\mathrm{c}}(\alpha_{\mathrm{c}} + 1)}{2}} \cdot (\eta^{n+1/2})^{\frac{\alpha_{\mathrm{c}}-1}{2}} & \text{if } \eta^{n+1/2} \geq 0 \quad (19a) \\[2ex] -2 \dfrac{\psi^n}{\eta^\star - \eta^{n-1/2}} & \text{if } \eta^{n+1/2} < 0 \quad (19b) \\[2ex] 0, & \begin{array}{l} \text{if } \eta^{n+1/2} < 0 \\ \text{and } \eta^\star = \eta^{n-1/2} \end{array} \quad (19c) \end{cases}$$
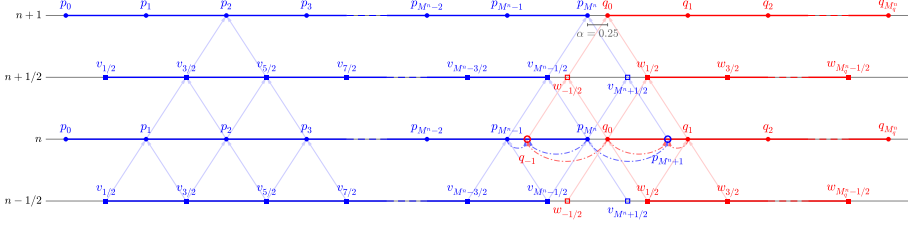
Fig. 2: *Schematic showing data flow of how different grid points at time index $n + 1$ are calculated with $\alpha = 0.25$ in Eq. (25). To prevent cluttering, arrows going straight up (indicating that the state of a grid point at time step n is needed to calculate the state of that grid point at $n + 1$) are suppressed. As an example of the usual case (refer to Eq. (15)), the points required to calculate p_2^{n+1} are shown. Furthermore, the points needed to calculate $p_{M^n}^{n+1}$ and q_0^{n+1} are shown. The most important difference with the usual case is that the virtual grid points $p_{M^n+1}^n$ and q_{-1}^n are the result of the interpolation of known pressure values at n using Eq. (27).*

where $\kappa = 1$ if $\psi^n \geq 0$, otherwise $\kappa = -1$. It should be noted that condition (19c) has been added to the definition of g from [14] to prevent a division by 0 in (19b). Finally, $\eta^\star = -y^\star - H_0$ where $y^\star$ is the value of $y^{n+3/2}$ calculated using system (18) (after expansion) without the collision potential. This means that system (18) needs to be calculated twice every iteration, once without the collision term and once with. The process of calculating the pressure difference $\Delta p^{n+1/2}$ in (18) will not be given here, but the interested reader is referred to [13, Ch. 5] for a derivation.

4   Dynamic grid

The defining feature of the trombone is its slide that alters the length of the tube, changing the resonant frequencies. In a companion paper [8], we present a method to dynamically change grid configurations of FD schemes by inserting and deleting grid points based on an instantaneous value of the time-varying wave speed $c(t)$. Although here, the tube length $L(t)$ is varied, the method still applies. Note that this method only works for slow (sub-audio rate) parameter changes.

We can split a tube with time-varying length L^n into two smaller sections with lengths L_p^n and L_q^n (in m) such that $L^n = L_p^n + L_q^n$. Splitting the schemes in (14) in this way yields two sets of first-order systems. The pressure and particle velocity of the first (left) system $p_{l_p}^n$ and $v_{l_p+1/2}^{n+1/2}$ are both defined over discrete domain $l_p \in \{0, \ldots, M^n\}$, and those of the second (right) system $q_{l_q}^n$ and $w_{l_q-1/2}^{n+1/2}$ are defined over discrete domain $l_q \in \{0, \ldots, M_q^n\}$, with

$$M^n = \lceil L_p^n/h \rceil, \quad \text{and} \quad M_q^n = \lfloor L_q^n/h \rfloor \tag{20}$$

where $\lceil \cdot \rceil$ denotes the ceiling operation. Note, that the domains for v and w have an extra grid point when compared to the regular case in (14) and that w is indexed with $l_q - 1/2$ rather than $l_q + 1/2$. The resulting system of FD schemes then becomes

$$\frac{\bar{S}_l}{\rho_0 c^2} \delta_{t+} p_{l_p}^n = -\delta_{x-}(S_{l+1/2} v_{l_p+1/2}^{n+1/2}), \tag{21a}$$

$$\rho_0 \delta_{t-} v_{l_p+1/2}^{n+1/2} = -\delta_{x+} p_{l_p}^n, \tag{21b}$$

$$\frac{\bar{S}_l}{\rho_0 c^2} \delta_{t+} q_{l_q}^n = -\delta_{x+}(S_{l-1/2} w_{l_q-1/2}^{n+1/2}), \tag{21c}$$

$$\rho_0 \delta_{t-} w_{l_q-1/2}^{n+1/2} = -\delta_{x-} q_{l_q}^n. \tag{21d}$$

Here, due to the different indexing for w, the spatial derivatives for the right system are flipped (δ_{x+} became δ_{x-} and vice versa). Also note, that l is still used for the spatial indices of $\bar{S}$ and S which now approximate $S(x)$ according to

$$S_l \approx \begin{cases} S(x = lh) & \text{for } x \in [0, L_p^n], \\ S(x = L^n - (M_q^n - l)h) & \text{for } x \in [L_p^n, L^n]. \end{cases} \tag{22}$$

The conditions for the outer boundaries of this system, i.e., at $l_p = 0$ and $l_q = M_q^n$, are the same as for the full system. The inner boundaries, $l_p = M^n$ and $l_q = 0$ are connected according to the method described in [8] to be explained shortly. To be able to calculate $p_{M^n}^{n+1}$ and q_0^{n+1}, the domains of v and w have been extended at the inner boundaries to include $v_{M^n+1/2}^{n+1/2}$ and $w_{-1/2}^{n+1/2}$. These, however, require points outside of the domains of $p_{l_p}^n$ and $q_{l_q}^n$, i.e., $p_{M^n+1}^n$ and q_{-1}^n. In [8] we propose to calculate these *virtual grid points* based on known values of the system. Despite the fact that [8] presents the method using a second-order system, it can still be applied here. The process of how $p_{M^n}^{n+1}$ and q_0^{n+1} are calculated is visualised in in Figure 2. Notice that all time steps use the same value of M^n and M_q^n. In other words, the expansion of the temporal operators in (9) do not affect the temporal indices n in M^n and M_q^n.

4.1 Changing the Tube Length

In the following, the location of a grid point u_l along the grid (in m from the left boundary) at time index n is denoted as $x_{u_l}^n$.

The two pairs of first order systems in (21) are placed on the same domain x with

$$x_{p_{l_p}}^n = l_p h, \quad \text{and} \quad x_{q_{l_q}}^n = L^n - (M_q^n - l_q)h, \tag{23}$$

describing the locations of the left system and right system respectively. Here, it can be observed that as the tube length L^n changes, the locations of the grid points of the right system will change. More specifically, as the trombone-slide

is extended and L^n increases, all grid points of the right system move to the right, and to the left for a contracting slide. If L^n is changed in a smooth fashion, the continuous domain $x \in [0, L^n]$ will not necessarily be subdivided into an integer amount of intervals N^n (of size $h = ck$). This is where a *fractional number of intervals* is introduced and is defined as

$$\mathcal{N}^n = L^n / h, \tag{24}$$

which is essentially the calculation of N in Eq. (17) without the flooring operation, and $N^n = \lfloor \mathcal{N}^n \rfloor$. The fractional part of $\mathcal{N}^n$ can then be calculated using

$$\alpha = \alpha^n = \mathcal{N}^n - N^n, \tag{25}$$

which describes the distance between the inner boundaries along the grid in terms of how many times h would fit in-between (which is always less than once). If $\mathcal{N}^n = N^n$ and $\alpha = 0$, the inner boundary locations perfectly overlap, and $x^n_{p_{M^n}} = x^n_{q_0}$. This also means that the domain x can be exactly divided into N^n equal intervals of size $h = ck$. As the virtual grid points $p^n_{M^n+1}$ and q^n_{-1} perfectly overlap with q^n_1 and $p^n_{M^n-1}$ respectively, these values can be used directly to calculate the grid points at the inner boundaries. This situation effectively acts as a rigid connection between the grid points at the inner boundaries defined as

$$p^n_{M^n} = q^n_0, \quad \text{if } \alpha = 0. \tag{26}$$

If $\alpha \neq 0$, some other definition for $p^n_{M^n+1}$ and q^n_{-1} needs to be found. We use quadratic Lagrangian interpolation according to

$$p^n_{M^n+1} = \frac{\alpha - 1}{\alpha + 1} p^n_{M^n} + q^n_0 - \frac{\alpha - 1}{\alpha + 1} q^n_1, \tag{27a}$$

$$q^n_{-1} = -\frac{\alpha - 1}{\alpha + 1} p^n_{M^n-1} + p^n_{M^n} + \frac{\alpha - 1}{\alpha + 1} q^n_0, \tag{27b}$$

which can then be used to calculate $v^{n+1/2}_{M^n+1/2}$ and $w^{n+1/2}_{-1/2}$ and consequently $p^{n+1}_{M^n}$ and q^{n+1}_0 (see Figure 2). This process is repeated every sample. It can be shown through the rigid connection in (26), that if $\alpha = 0$, the definitions in (27) reduce to $p^n_{M^n+1} = q^n_1$ and $q^n_{-1} = p^n_{M^n-1}$ as stated before.

4.2 Adding and removing grid points

As the tube length L^n changes, L^n_p and L^n_q also change according to

$$L^n_p = L^{n-1}_p + 0.5 L^n_{\text{diff}}, \quad L^n_q = L^{n-1}_q + 0.5 L^n_{\text{diff}}, \tag{28}$$

where
$$L_{\text{diff}}^n = L^n - L^{n-1}, \tag{29}$$

which causes the number of intervals between grid points M^n and M_q^n to change as well, according to Eq. (20).

The following state vectors are introduced for the pressure, defined for $n+1$ and n
$$\mathbf{p}^n = [p_0^n, p_1^n, ..., p_{M^n}^n]^T, \ \mathbf{q}^n = [q_0^n, q_1^n, ..., q_{M_q^n}^n]^T, \tag{30}$$

and for the velocity, defined for $n + 1/2$ and $n - 1/2$
$$\begin{aligned}
\mathbf{v}^{n-1/2} &= [v_{1/2}^{n-1/2}, v_{3/2}^{n-1/2}, ..., v_{M^n+1/2}^{n-1/2}]^T, \\
\mathbf{w}^{n-1/2} &= [w_{-1/2}^{n-1/2}, w_{1/2}^{n-1/2}, ..., w_{M_q^n-1/2}^{n-1/2}]^T,
\end{aligned} \tag{31}$$

and contain the different states over the discrete domains defined at the beginning of this section. Here, T denotes the transpose operation.

If $N^n > N^{n-1}$, points are added to the left and right system in an alternating fashion:
$$\begin{cases} \mathbf{p}^n = [(\mathbf{p}^n)^T, I_3\mathbf{r}^n]^T \\ \mathbf{v}^{n-1/2} = [(\mathbf{v}^{n-1/2})^T, I_3\mathbf{z}_v^{n-1/2}]^T \end{cases} \quad \text{if } N^n \text{ is odd,}$$
$$\begin{cases} \mathbf{q}^n = [I_3^{\leftarrow}\mathbf{r}^n, (\mathbf{q}^n)^T]^T \\ \mathbf{w}^{n-1/2} = [I_3^{\leftarrow}\mathbf{z}_w^{n-1/2}, (\mathbf{w}^{n-1/2})^T]^T \end{cases} \quad \text{if } N^n \text{ is even,} \tag{32}$$

where
$$\begin{aligned}
\mathbf{r}^n &= [p_{M^n-1}^n, p_{M^n}^n, q_0^n, q_1^n]^T, \\
\mathbf{z}_v^{n-1/2} &= [v_{M^n-1/2}^{n-1/2}, v_{M^n+1/2}^{n-1/2}, w_{1/2}^{n-1/2}, w_{3/2}^{n-1/2}]^T - \boldsymbol{\eta}, \\
\mathbf{z}_w^{n-1/2} &= [v_{M^n-3/2}^{n-1/2}, v_{M^n-1/2}^{n-1/2}, w_{-1/2}^{n-1/2}, w_{1/2}^{n-1/2}]^T + \boldsymbol{\eta}^{\leftarrow},
\end{aligned} \tag{33}$$

and cubic Lagrangian interpolator
$$I_3 = \left[-\frac{\alpha(\alpha+1)}{(\alpha+2)(\alpha+3)} \quad \frac{2\alpha}{\alpha+2} \quad \frac{2}{\alpha+2} \quad -\frac{2\alpha}{(\alpha+3)(\alpha+2)} \right]. \tag{34}$$

Here,
$$\boldsymbol{\eta} = \boldsymbol{\eta}^{n-1/2} = \left(w_{-1/2}^{n-1/2} - v_{M^n+1/2}^{n-1/2} \right) \cdot [0, 0, 1, 1]^T \tag{35}$$

adds an offset to half of the elements in the $\mathbf{z}$ vectors depending on the difference between $v_{M^n+1/2}^{n-1/2}$ and $w_{-1/2}^{n-1/2}$. Why this is necessary will be further explained in Section 4.3. Finally, $I_3^{\leftarrow}$ and $\boldsymbol{\eta}^{\leftarrow}$ are flipped versions of (34) and (35) respectively.

If $N^n < N^{n-1}$, points are simply removed from the vectors according to

$$
\begin{cases}
\mathbf{p}^n = [p_0^n, \ldots, p_{M^n-1}^n]^T \\
\mathbf{v}^{n-1/2} = [v_{1/2}^{n-1/2}, \ldots, v_{M^n-1/2}^{n-1/2}]^T
\end{cases}
\quad \text{if } N^n \text{ is even,}
$$

$$
\begin{cases}
\mathbf{q}^n = [q_1^n, \ldots, q_{M_q^n}^n]^T \\
\mathbf{w}^{n-1/2} = [w_{1/2}^{n-1/2}, \ldots, w_{M_q^n-1/2}^{n-1/2}]^T
\end{cases}
\quad \text{if } N^n \text{ is odd.}
$$

(36)

Notice that the even and odd conditions in Eqs. (32) and (36) can be swapped. To stay as close to the desired location of adding and removing grid points as possible, this requires the ceiling and flooring operations in (20) to be swapped as well.

4.3 Drift of w

The inner boundaries of the pressure states p and q are connected by (27), but no such connection exists for the velocity states v and w. As the radiating boundary is implemented on the pressure grid, this leaves w without any boundary condition; it is only "held in place" by the pressure values of q, or more specifically, by derivatives (both spatial and temporal). As FD schemes are an approximation, it does not give a perfect solution and w tends to 'drift' during the simulation, especially when L^n is changed.

Luckily, as the pressure values are also calculated from derivatives of the velocity, the absolute state of w does not matter. The difference in values at the connection point is also irrelevant as there is no spatial derivative taken between v and w (refer to Figure 2). Finally, the pressure values are used for the output audio of the simulation, so the drift does not affect the audio.

The absolute states of the velocity vectors do, however, need to be accounted for when adding points to the $\mathbf{v}$ and $\mathbf{w}$ using (32). The current drift can be approximated by observing the difference between $w_{-1/2}^{n-1/2}$ and $v_{M^n+1/2}^{n-1/2}$, as these have approximately the same x location ($x_{w_{-1/2}}^n \approx x_{v_{M^n+1/2}}^n$) when a grid point is added. This is then used in a drift-correction vector $\boldsymbol{\eta}^{n-1/2}$ presented in (35). When a point is added to v, the values of w in $\mathbf{z}_v$ are offset by the aforementioned difference and when a point is added to w the same happens (inverted) for the values of v in $\mathbf{z}_w$. This way, the drift is allowed, but does not affect the state of the newly added grid points. Notice that the drift does not affect the operations of point removal in (36).

4.4 State Correction

As L^n, and consequently the number of grid points, is decreased, it might occur that the grid points at the inner boundaries $p_{M^n}^n$ and q_0^n have a very

different value when $\alpha \gtrsim 0$, i.e., right before a point is removed. This violates the rigid connection in Eq. (26).

We propose in [8] to add an artificial spring-like connection between the grid points at the inner boundaries that "corrects" the state of these points. Applying this to system (21) extends Eqs. (21a) and (21c) according to

$$\frac{\bar{S}_l}{\rho_0 c^2}\delta_{t+}p_{l_p}^n = -\delta_{x-}(S_{l+1/2}v_{l_p+1/2}^{n+1/2}) + J_p(x_{p_M n}^n)F_{\text{sc}}^n, \tag{37a}$$

$$\frac{\bar{S}_l}{\rho_0 c^2}\delta_{t+}q_{l_q}^n = -\delta_{x+}(S_{l-1/2}w_{l_q-1/2}^{n+1/2}) - J_q(x_{q_0}^n)F_{\text{sc}}^n, \tag{37b}$$

where the spreading operators are defined as

$$J_p(x_i^n) = \begin{cases} \frac{1}{h}, & l_p = \lfloor x_i^n/h \rfloor \\ 0, & \text{otherwise}, \end{cases} \quad \text{and} \tag{38}$$

$$J_q(x_i^n) = \begin{cases} \frac{1}{h}, & l_q = \lfloor x_i^n/h \rfloor - M^n \\ 0, & \text{otherwise}. \end{cases}$$

Furthermore, the correction effect is defined as

$$F_{\text{sc}}^n = \beta\left(\mu_{t\cdot}\eta_{\text{sc}}^n + \sigma_{\text{sc}}\delta_{t\cdot}\eta_{\text{sc}}^n\right), \tag{39}$$

with spring damping σ_{sc}, pressure difference

$$\eta_{\text{sc}}^n \triangleq q_0^n - p_{M^n}^n, \tag{40}$$

and scaling coefficient

$$\beta = \beta(\alpha) = \frac{1-\alpha}{\alpha + \varepsilon}. \tag{41}$$

Here, $\varepsilon \ll 1$ to prevent division by 0. Just like in [8], the implementation of the correction effect allows for an infinite β when $\alpha = \varepsilon = 0$ acting like a rigid connection between Eqs. (37a) and (37b).

5 Implementation

The implementation has been done in C++ using the JUCE framework [1], and is available online[2] as well as a demo showcasing it.[3] The audio output of the system can be retrieved by selecting a grid point on the pressure grid and listening to this at the given sample rate f_{s}. Here, the radiating boundary $q_{M_q^n}^n$ is chosen, as this is where the sound enters the listening space in the real

---

[1]https://juce.com/
[2]https://github.com/SilvinWillemsen/cppBrass/releases/
[3]https://youtu.be/Ht5gVNrshYo

Table 1: *Geometry of a measured trombone taken from [16]. Numbers correspond to Figure 3.*

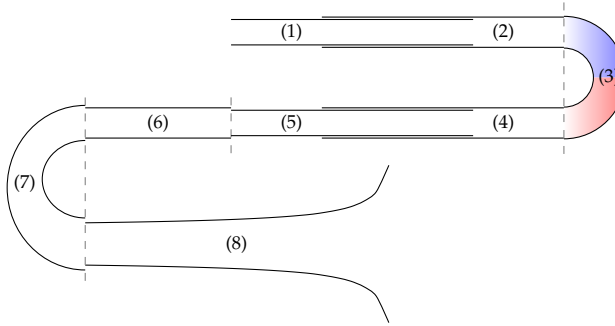| Part of tube	Length (cm)	Radius (cm)
Inner slide (1)	70.8	0.69
Outer slide (extended) (2)	53	0.72
Slide crook (3)	17.7	0.74
Outer slide (extended) (4)	53	0.72
Inner slide (5)	71.1	0.69
Gooseneck (6)	24.1	0.71
Tuning slide (7)	25.4	0.75, 1.07
Bell flare (8)	50.2	1, 10.8

Fig. 3: *Diagram showing the trombone geometry (not to scale). Numbers correspond to the parts of the tube found in Table 1 and dashed lines highlight where the different parts are separated. The tube is split in the middle of the slide crook with the colours corresponding to those in Figure 2.*

world. To mimic low-pass filtering happening due to a distributed radiating area, a 4th-order low-passing Butterworth filter with a cutoff frequency of $f_c = \sqrt{c^2\pi/S(L)} \approx 3245$ Hz is used. This equation is retrieved by choosing the listening point to be at the bell surface and integrating over the bell area.

5.1 Parameters

For the most part, the parameters used in the simulation have been obtained from [13, 16, 17]. The lengths and radii of different parts of the tube can be found in Table 1 and a diagram showing this geometry is shown in Figure 3. The system is split in the middle of the slide crook such that the ranges for the lengths of the two tubes are $L_p^n \in [0.797, 1.327]$ and $L_q^n \in [1.796, 2.326]$.

Other parameters used in the simulation can be found in Table 2. Not included here is λ, which has been set slightly lower than the stability condition in (16), i.e., $\lambda = 0.999$. Although the implementation works when $\lambda = 1$, this is done to tolerate (much) higher speeds of change in L^n before instability occurs (see Section 5.2). Not satisfying condition (16) causes bandlimiting and

Table 2: *List of parameter values used for the simulation. Taken from ⋆[16], *[13] or **[17] with temperature $T = 26.85°C$.*

Name	Symbol (unit)	Value
Tube		
Length	L (m)	$2.593 \leq L \leq 3.653^\star$
Air density	ρ_0 (kg/m^3)	1.1769**
Wave speed	c (m/s)	347.23**
Geometry	S (m^2)	See Table 1.
Lip reed		
Mass	M_r (kg)	$5.37 \cdot 10^{-5}$*
Frequency	ω_r (rad/s)	$20 \leq \omega_r/2\pi \leq 1000$
Mouth pressure	P_m (Pa)	$0 \leq P_m \leq 6000$
Damping	σ_r (s^{-1})	5*
Eff. surface area	S_r (m^2)	$1.46 \cdot 10^{-5}$*
Width	w_r (m)	0.01*
Equilibrium sep.	H_0 (m)	$2.9 \cdot 10^{-4}$*
Coll. stiffness	K_c (N/m)	10^4
Nonlin. coll. coeff.	α_c (-)	3
Other		
State corr. damping	σ_{sc}	1
Sample rate	f_s (Hz)	44100

dispersive effects [15], but such a small deviation from the condition has no perceptual influence on the output sound and outweighs the problems caused by instability.

As the tube acts mainly as an amplifier for specific resonant frequencies it is important to match the frequency of the lip reed to a resonating mode of the tube. This frequency depends on L^n in the following way

$$\omega_r^{n+1/2} = \mathcal{F}\frac{2\pi c}{\rho_0 L^{n+1/2}} \, , \tag{42}$$

where $L^{n+1/2} = L^n$ and scalar multiplier $\mathcal{F} = 2.4$ was heuristically found to best match the 4th resonating mode of the tube and generates a recognisable brass sound.

5.2 Limit on speed of change

To reduce audible artifacts and instability issues from adding and removing points, and to stay in the sub-audio rate regime, a limit can be placed on (29) as

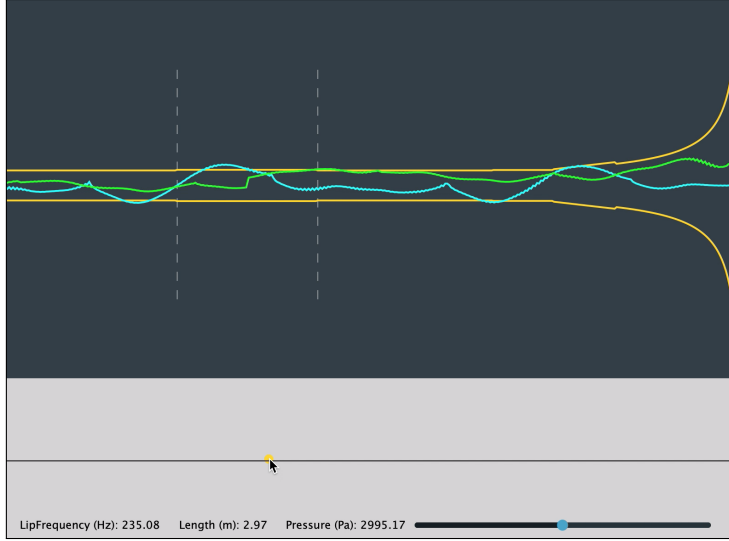$$L_{\text{diff}}^n \leq \mathcal{N}_{\text{maxdiff}}h, \tag{43}$$

Fig. 4: *Screenshot of the graphical user interface (GUI). The geometry (in orange) as well as the states of the pressure (in blue) and velocity scaled by S (in green) are shown. For clarity, the start and end of the outer slide are denoted by dashed lines. The drift of w as explained in Section 4.3 is visible from the "kink" in the green line exactly in the middle of the outer slide.*

where $\mathcal{N}_{\mathrm{maxdiff}}$ is the maximum change in $\mathcal{N}$ per sample and has been set to $\mathcal{N}_{\mathrm{maxdiff}} = 1/20$. This means that a grid point can be added or removed every 20 samples and allows the entire range of L to be traversed in ca. 0.06 s at a sample rate of $f_{\mathrm{s}} = 44100$ Hz.

5.3 State correction

The introduction of system states at $n+1$ through the centred operators in Eq. (39) seem to make the scheme implicit. It is, however, possible to calculate F_{sc} explicitly [15, 18]. The same operators also introduce the need for values at $n-1$, i.e., $p_{M^n}^{n-1}$ and q_0^{n-1}. Therefore, the vectors $\mathbf{p}^{n-1}$ and $\mathbf{q}^{n-1}$ will need to be stored, and the operations to add and remove grid points as described in 4.2 need to be applied to these as well. One could argue that only two points at the inner boundaries are needed for the calculation and to create $\mathbf{r}$ in (33) at $n-1$. For generality, we continue with the entire vectors defined over the same domains as $\mathbf{p}^n$ and $\mathbf{q}^n$ respectively.

5.4 Graphical User Interface and Control Mapping

A screenshot of the graphical user interface (GUI) is shown in Figure 4. The geometry of the tube is plotted along with paths showing the pressure states

```
while application is running do
    Retrieve new parameters          (Lⁿ, ωᵣⁿ and Pₘⁿ)
    Update Lₚⁿ and L_qⁿ               (Eqs. (29), (43) & (28))
    Calc. 𝒩ⁿ and Nⁿ                  (Eqs. (24) and (17))
    Calc. αⁿ                         (Eq. (25))
    if Nⁿ ≠ Nⁿ⁻¹ then
        Add or remove point          (Eq. (32) or (36))
        Update Mⁿ and M_qⁿ           (Eq. (20))
    end
    Calc. p_{Mⁿ+1}ⁿ and q_{-1}ⁿ      (Eqs. (27))
    Calc. vⁿ⁺¹ᐟ² and wⁿ⁺¹ᐟ²          (Eqs. (21b) and (21d))
    Calc. yⁿ⁺³ᐟ² w/o collision        (Eqs. (18))
    Calc gⁿ⁺¹ᐟ²                       (Eq. (19))
    Calc. yⁿ⁺³ᐟ² with collision       (Eqs. (18))
    Calc. U_Bⁿ⁺¹ᐟ² and U_rⁿ⁺¹ᐟ²      (Eqs. (18c) and (18d))
    Calc. pⁿ⁺¹ and qⁿ⁺¹              (Eqs. (37))
    Retrieve output
    Update system states             (pⁿ⁻¹ = pⁿ, pⁿ = pⁿ⁺¹) (same for
                                      vⁿ⁻¹ᐟ² = ...,
                                      yⁿ⁻¹ᐟ², yⁿ⁺¹ᐟ², and ψⁿ)
    Update Nⁿ⁻¹                       (Nⁿ⁻¹ = Nⁿ)
    Increment n
end
```

Algorithm 1: *Pseudocode showing the order of calculations of the algorithm implementing the trombone.*

in blue and the velocity (scaled by the geometry S) in green. The audio thread of the application runs at 44100 Hz whereas the graphics are updated at a rate of 15 Hz.

The real-time application is controlled by interacting with the bottom panel using the mouse. The x-axis is mapped to tube-length L^n and also modifies the lip-reed frequency ω_r according to Eq. (42). The y-axis changes the multiplier $\mathcal{F}$ in Eq. (42) and the black line in the vertical middle of the control panel is mapped to $\mathcal{F} = 2.4$. The pressure is modulated by a slider at the bottom of the control panel. As of now, no focus has been put on intuitive parameter mapping; it has only been implemented for simple parameter exploration.

5.5 Order of Calculation

Algorithm 1 shows the order in which the different parts of the system presented in this paper are calculated.

6 Results and Discussion

The real-time implementation has been tested on a MacBook Pro with a 2.2 GHz Intel i7 processor and was informally evaluated by the authors. The speed of the algorithm was tested with and without the graphics-thread and using three different styles of interaction: static excitation at the shortest and longest length, and rapidly (and continuously) changing L and ω_r between their minimum and maximum values given in Table 2. The pressure was kept at $P_{\mathrm{m}} = 3000$ Pa at all times. The results are shown in Table 3. Differences in CPU usage between a short and long tube length are because more grid points need to be calculated in the long case. The recalculation of the geometry maximally once every 20 samples in the rapidly moving case explains the increase in CPU usage there. These results show that the implementation can easily be used as an audio plugin, with or without graphics.

Table 3: *Average CPU usage (in %) for different graphics settings and various interactions with the application.*

Tube length	Graphics (%)	No graphics (%)
Short ($L^n = 2.593$ m)	12.1	4.3
Long ($L^n = 3.653$ m)	14.4	5.2
Rapidly changing	17.7	10.1

Informal listening tests by the authors confirm that the audio output of the simulation exhibits brass-like qualities. However, the implementation requires some further refinements to be considered as a complete trombone. Possible extensions to improve the realism of the simulation sound could be to add viscothermal losses [19] or nonlinear effects [3]. Furthermore, for lower values of the lip frequency ω_r, the sound exhibits extra oscillatory behaviour making the output "non-smooth". This might be due a higher average displacement of y for lower ω_r and the nonlinear collision present in the lip model will have a greater effect on its displacement. Variable collision stiffness might solve this issue but is left for future work.

Informal listening by the authors shows that the method used to implement the dynamic grid does not introduce perceivable audible artifacts, even when L^n is changed very rapidly. Naturally, this needs to be confirmed by formal listening tests. Despite the limit placed on the speed of change of L^n in (43) the control of the application does not exhibit a noticeable delay and changes in L^n feel immediate.

The main difference between the method in [8] and the version used here, is that the method is applied to a system of first-order equations rather than the second-order 1D wave equation. Because the connection between the inner boundaries is only applied to the grid functions describing pressure, a drift

occurs in w as it is left without boundary conditions. Although this drift does not have an effect on the output sound, as discussed in Section 4.3, too high or low values might cause rounding errors in the simulation. As it is expected that this only happens at extremely high or low values after a long simulation length, the drift is not considered an issue at this point.

7   Conclusion

In this paper, we have presented a full implementation of the trombone including a lip reed, radiation and a tube, discretised using FDTD methods on a dynamic grid. Informal evaluation by the authors shows that the implementation exhibits no audible artifacts when grid points are added and removed, even under relatively fast variation in tube length. Naturally, this needs to be confirmed by formal listening tests. Moreover, the simulation easily runs in real-time allowing it to be used as an audio plugin.

Future work will include extending the tube model to include more realistic viscothermal and nonlinear effects and variable collision stiffness in the lip model. Furthermore, the investigation of more intuitive control parameter mappings is a necessary step towards a real-time instrument.

8   Acknowledgements

9   References

[1] M. Campbell, "Brass instruments as we know them today," *Acta Acustica united with Acustica*, vol. 90, no. 4, pp. 600–610, 2004.

[2] A. Hirschberg, J. Gilbert, R. Msallam, and A. Wijnands, "Shock waves in trombones," *J. Acoust. Soc. Am.*, vol. 99, no. 3, pp. 1754–1758, 1996.

[3] R. Msallam, S. Dequidt, S. Tassart, and R. Causse, "Physical model of the trombone including non-linear propagation effects," in *ISMA: International Symposium of Music Acoustics*, 1997.

[4] R. Msallam, S. Dequidt, R. Causse, and S. Tassart, "Physical model of the trombone including nonlinear effects. Application to the sound synthesis of loud tones," *Acta Acustica united with Acustica*, vol. 86, no. 4, pp. 725–736, 2000.

[5] P. R. Cook, *Real sound synthesis for interactive applications*. CRC Press, 2002.

[6] R. L. Harrison, S. Bilbao, J. Perry, and T. Wishart, "An environment for physical modeling of articulated brass instruments," *Computer Music Journal*, vol. 39, no. 4, pp. 80–95, 2015.

[7] S. Bilbao and J. Chick, "Finite difference time domain simulation for the brass instrument bore," *J. Acoust. Soc. Am.*, vol. 134, no. 6, pp. 3860–3871, 2013.

[8] S. Willemsen, S. Bilbao, M. Ducceschi, and S. Serafin, "Dynamic grids for finite-difference schemes in musical instrument simulations," in *Proc. of the 23rd Int. Conf. on Digital Audio Effects (DAFx)*, 2021.

[9] A. Webster, "Acoustical impedance, and the theory of horns and of the phonograph," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 5, no. 7, pp. 275–282, 1919.

[10] S. Bilbao and R. Harrison, "Passive time-domain numerical models of viscothermal wave propagation in acoustic tubes of variable cross section," *J. Acoust. Soc. Am.*, vol. 140, pp. 728–740, 2016.

[11] H. Levine and J. Schwinger, "On the radiation of sound from an unflanged circular pipe," *Physical Review*, vol. 73, no. 2, pp. 383–406, 1948.

[12] F. Silva, P. Guillemain, J. Kergomard, B. Mallaroni, and A. Norris, "Approximation formulae for the acoustic radiation impedance of a cylindrical pipe," *Journal of Sound and Vibration*, vol. 322, pp. 255–263, 2009.

[13] R. L. Harrison-Harsley, "Physical modelling of brass instruments using finite-difference time-domain methods," Ph.D. dissertation, University of Edinburgh, 2018.

[14] M. Ducceschi, S. Bilbao, S. Willemsen, and S. Serafin, "Linearly-implicit schemes for collisions in musical acoustics based on energy quadratisation," *Journal of the Acoustical Society of America (JASA)*, vol. 149, pp. 3502–3516, 2021.

[15] S. Bilbao, *Numerical Sound Synthesis*. United Kingdom: John Wiley & Sons, 2009.

[16] T. Smyth and F. S. Scott, "Trombone synthesis by model and measurement," *EURASIP Journal on Advances in Signal Processing*, 2011.

[17] A. H. Benade, "On the propagation of sound waves in a cylindrical conduit," *Journal of the Acoustical Society of America*, vol. 44, no. 2, pp. 616–623, 1968.

[18] S. Bilbao, "A modular percussion synthesis environment," in *Proc. of the 12th Int. Conference on Digital Audio Effects (DAFx)*, 2009.

[19] S. Bilbao and R. L. Harrison, "Passive time-domain numerical models of viscothermal wave propagation in acoustic tubes of variable cross section," *The Journal of the Acoustical Society of America*, vol. 140, pp. 728–740, 2016.