AN ALGORITHM FOR A VALVED BRASS INSTRUMENT SYNTHESIS ENVIRONMENT USING FINITE-DIFFERENCE TIME-DOMAIN METHODS WITH PERFORMANCE OPTIMISATION

Reginald L Harrison, Stefan Bilbao
Acoustics and Audio Group,
University of Edinburgh
Edinburgh, UK
r.l.harrison-3@sms.ed.ac.uk *
s.bilbao@ed.ac.uk

James Perry
Edinburgh Parallel Computing Centre,
University of Edinburgh
Edinburgh, UK
j.perry@epcc.ed.ac.uk

ABSTRACT

This paper presents a physical modelling sound synthesis environment for the production of valved brass instrument sounds. The governing equations of the system are solved using finite-difference time-domain (FDTD) methods and the environment is implemented in the C programming language. Users of the environment can create their own custom instruments and are able to control player parameters such as lip frequency, mouth pressure and valve openings through the use of instrument and score files.
The algorithm for sound synthesis is presented in detail along with a discussion of optimisation methods used to reduce run time. Binaries for the environment are available for download online for multiple platforms.

1. INTRODUCTION

The problem of synthesis of brass instrument sounds, such as those from the instrument shown in Fig. 1, has been an active research topic within the field of physical modelling and there are various avenues of approach. Modal methods are used, e.g., in the MoReeSC [1] software package. Digital waveguide methods model wave propagation using delay lines, where the effects of loss and dispersion are lumped into terminating filters [2, 3, 4]. A combination of methods that involves parallel convolution and modal methods along with one way nonlinear wave propagation have also been successful [5, 6].
Although these are all efficient methods of synthesis, the assumptions made to improve performance can lead to awkward implementation of time varying tube configurations—as is the case for articulated valved instrument sounds. More general numerical methods, such as finite difference time domain techniques (FDTD), can be used in this case and are suitable for modelling the valved brass instrument system [7, 8]. Although computationally more intensive, FDTD algorithms for brass instrument synthesis can be run on modern personal computers and, with the use of optimisation methods in the C programming language, simulation times can approach or surpass real-time.
After the model equations for the valved brass instrument system are described in Section 2, the FDTD approximations to continuous operators are introduced in Section 3 and then used in Section 4 to create discrete versions of the system equations and their corresponding updates. Optimisation techniques in the C programming environment are discussed along with benchmarking times in Section 5 and finally concluding remarks and planned extensions to the environment are presented in Section 6. Supplementary materials including the environment binaries, example sounds and test files are available for download at
http://www2.ph.ed.ac.uk/~s0916351/dafx15.html.

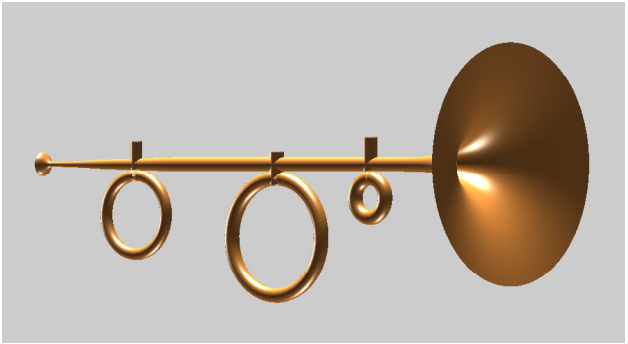

Figure 1: Functional representation of a valved brass instrument. Loops below the main instrument bore are the additional lengths of tubing that airflow can be diverted into through the use of pistons.

2. BRASS INSTRUMENT SYSTEM

A typical valved brass instrument can be separated, functionally, into three components: the input (excitation mechanism), resonator (instrument bore and valve sections), and radiation (interaction of the bore with the surrounding acoustic space). Waves propagate within the instrument bore which includes additional pieces of tubing that waves can be partitioned into through the use of valves. The input and radiation models can be defined separately and then coupled to the extremities of the instrument bore. This section presents the system equations for each component of the model.

2.1. Wave Propagation

Starting in the frequency domain, a model for wave propagation inside a section of an acoustic tube that includes viscous and thermal losses [9] may be written as

$$\partial_x \left(S\hat{v} \right) = -S\hat{Y}\hat{p}, \quad \partial_x \hat{p} = -\hat{Z}\hat{v} \tag{1}$$

* Author for contact

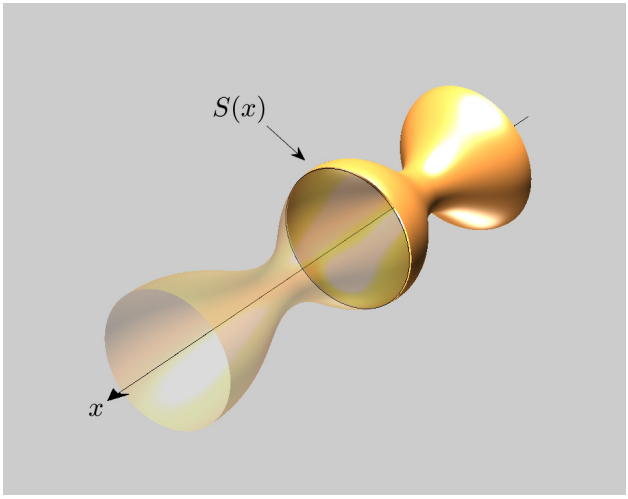

Figure 2: An acoustic tube showing the variation of the surface area, $S(x)$, with axial coordinate x.

where $\hat{p}(x,\omega)$ and $\hat{v}(x,\omega)$ are the acoustic pressure and particle velocity as functions of an angular frequency ω and axial coordinate x, where $x \in [0, L]$ and L is the total length of the instrument. $S(x)$ is the bore cross-section; see Fig. 2. $\hat{Z}(\omega)$ and $\hat{Y}(\omega)$ represent the characteristic series impedance and shunt admittance per unit length of the system that include wave propagation and the viscous and thermal losses. The $\hat{(\cdot)}$ operator denotes a frequency domain function and ∂_x denotes first order partial differentiation with respect to axial coordinate x.
A complete description of the impedance and admittance in equation (1) relies on Bessel functions for tubes of circular cross section. In practical time domain implementations, various approximations are necessary. A typical approximation strategy involves a high frequency approximation leading to a series of fractional powers of $j\omega$ as per, e.g., [10], accompanied by truncation of the series. Transformation to the time domain follows from the replacements $(j\omega)^u \rightarrow \partial_{t^u}$, where ∂_{t^u} is the u^{th} partial derivative with respect to time, t. Equations (1) become

$$\rho\partial_t v + fv + g\partial_{t^{1/2}} v + \partial_x p = 0 \tag{2a}$$

$$\frac{S}{\rho c^2}\partial_t p + q\partial_{t^{1/2}} p + \partial_x (Sv) = 0 \tag{2b}$$

where

$$f = \frac{3\eta\pi}{S}, \quad g = 2\sqrt{\frac{\rho\eta\pi}{S}}, \quad q = \frac{2(\gamma-1)}{\nu c^2}\sqrt{\frac{\eta\pi S}{\rho^3}} \tag{3}$$

and ∂_t and $\partial_{t^{1/2}}$ are first and half order partial derivatives with respect to time. Under further approximations this model reduces to the Webster-Lokshin model [11], as illustrated in [7]. The symbols ρ, c, η, ν^2 and γ are respectively: the density, speed of sound, viscosity, Prandtl number and ratio of specific heats. Values of these constants as a function of temperature are given by Benade [12] and are reprinted by Keefe [10].

2.2. Valve Junctions

In instruments such as the trumpet, valves are employed to direct airflow from the main bore into additional pieces of tubing, the gross

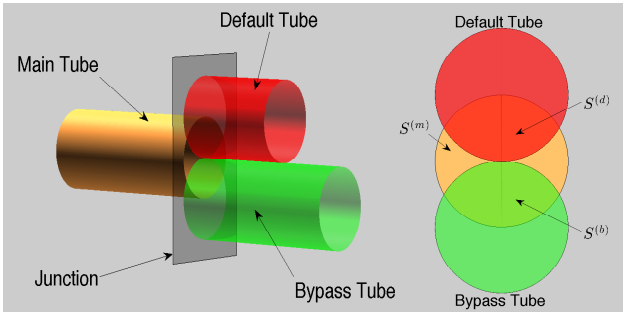

Figure 3: Left: Schematic of a valve junction showing the main tube in copper, the default valve tube in red and the bypass tube in green. The grey rectangle represents the junction surface. Right: Overlap of tubes at valve junction. Notice that in the case of circular tubes, the total overlapped surface between the three tubes will be less than the surface area of the main tube at the junction.

effect of which is to lengthen the instrument and lower its resonances. In normal use players either have the valve fully engaged or not at all, therefore only one possible path is available at a time for waves to propagate within the instrument. It is possible, however, to hold the valve in a partially open configuration, in which case the interaction between three pieces of tubing must be considered which results in more complex instrument resonances and sounds of a multiphonic timbre. Consider the system in Fig. 3, where one piece of tubing, labelled *main*, with cross-sectional surface area $S^{(m)}$ at the junction is overlapping two separate pieces of tubing, labelled *default* and *bypass*, with cross sectional areas $S^{(d)}$ and $S^{(b)}$ at the junction. The surface areas at the junction are defined by

$$S^{(d)} = q^{(d)} S^{(m)}, \quad S^{(b)} = q^{(b)} S^{(m)} \tag{4}$$

where $q^{(d)}(t)$ and $q^{(b)}(t)$ define the ratios of overlap between the default or bypass tubes with the main tube so that

$$q^{(d)} + q^{(b)} \leq 1 \tag{5}$$

A wave propagates through the main tube until it meets the junction between the three tubes. At this point the wave is then split between the default and bypass tubes. At the junction, the pressure is assumed constant at the point of contact of the three tubes and the volume velocity at the junction sums to zero [13, 8].

$$p^{(m)} = p^{(d)} = p^{(b)} \tag{6a}$$

$$S^{(m)} v^{(m)} = S^{(d)} v^{(d)} + S^{(b)} v^{(b)} \tag{6b}$$

Changing the overlap in (5) changes the partitioning of the airflow in (6b) and therefore modifies the resonance's of the instrument.

2.3. Radiation

Radiation of sound from a brass instrument, to the simplest approximation, can be considered in the same way as that from an unflanged cylinder, for which Levine and Schwinger proposed a suitable model [14]. To apply this in the time domain, rational approximations must again be made which result in a radiation impedance in terms of a normalised frequency $\omega' = a\omega/c$ [15]

$$Z_R = \rho c \left(\frac{(1+\Gamma)\Lambda j\omega' + \Gamma\Lambda (j\omega')^2}{1 + \Gamma + (\Lambda + \Gamma\Theta) j\omega' + \Gamma\Lambda\Theta (j\omega')^2} \right) \tag{7}$$

where

$$\Gamma = 0.505, \quad \Lambda = 0.613, \quad \Theta = 1.111 \tag{8}$$

and a is the radius of the tube at its radiating end. The system described by equation (7) has an equivalent circuit form that is described in [7]. For brevity the network equations are presented in combined form

$$\bar{v} = v_R + \left(\frac{1}{\Gamma\rho c} + \frac{\Theta a}{\rho c^2}\frac{d}{dt}\right) p_R \tag{9a}$$

$$\bar{p} = \Lambda\rho a \frac{dv_R}{dt} \tag{9b}$$

$$\bar{p} = \left(1 + \frac{1}{\Gamma} + \frac{\Theta a}{c}\frac{d}{dt}\right) p_R \tag{9c}$$

where $\bar{v} = v(L,t)$ and $\bar{p} = p(L,t)$ are values taken at the end of the instrument, v_R and p_R denote network variables (equivalent current and voltage) and d/dt denotes differentiation respect to time.

2.4. Lip Reed

The subject of lip reed modelling has seen a large amount of investigation, and models of varying degrees of complexity are available. For this synthesis environment a simple one degree of freedom, outward striking reed model been chosen as the excitation mechanism of the instrument [16]. The reed dynamics are described by

$$\frac{d^2y}{dt^2} + \sigma\frac{dy}{dt} + \omega_0^2 y = \frac{S_r \Delta p}{\mu_r} \tag{10a}$$

where $y(t)$ is the reed displacement from its equilibrium position H, and ω_0 and σ are the reed's natural resonance angular frequency and damping parameter. S_r and μ_r are the effective surface area and mass of the reed. $\Delta p(t)$ is the pressure difference between the mouth pressure, $p_m(t)$, and that within the instrument embouchure so that

$$\Delta p = p_m - p\,(0,t) \tag{10b}$$

The dynamics of the lip reed are coupled to the instrument through a Bernoulli term, generated by the pressure difference between the mouth and instrument, and by a volume velocity produced by the motion of the reed

$$u_m = w\,[y + H]_+ \sqrt{\frac{2|\Delta p|}{\rho}} \mathrm{sign}\,(\Delta \mathrm{p}) \tag{10c}$$

$$u_r = S_r \frac{dy}{dt} \tag{10d}$$

$$S\,(0)\,v\,(0,t) = u_m + u_r \tag{10e}$$

where w is the effective lip width and $u_m(t)$ and $u_r(t)$ denote volume velocities generated by the pressure difference and the lip motion. The function $[\cdot]_+$ is defined as $[\cdot]_+ = \max(\cdot, 0)$, meaning that when the lips are closed there is no flow.

3. FDTD SCHEME

3.1. Difference Operators

Before presenting the update schemes for the system equations it is useful to define the discrete operators used in this work. Consider a grid function, ζ_l^n, defined for integer $l = 0, \ldots, N$ and $n = 0, 1 \ldots,$. Such a grid function represents an approximation to an underlying function $\zeta(x,t)$, as $\zeta_l^n \approxeq \zeta\,(lh, nk)$, where here, h is the grid spacing and k is the time step (in audio applications, the inverse of the sampling frequency).
Let $e_{x\pm}$ and $e_{t\pm}$ be spatial and temporal shift operators

$$e_{x\pm}\zeta_l^n = \zeta_{l\pm1}^n, \quad e_{t\pm}\zeta_l^n = \zeta_l^{n\pm1} \tag{11}$$

Combinations of these basic shift operators can then be used to arrive at various approximations to partial derivatives as well as averaging operators, which approximate a multiplication by unity, that can centre schemes; see Table 1 for a full list of discrete operators used in this work and the operators they approximate.

Table 1: *List of discrete operators and the continuous operators they approximate. Note that the constants a_r and b_r are defined in Section 3.2.*

Spatial Operators	Expression	Approximates
Forwards Difference, δ_{x+}	$(e_{x+} - 1)/h$	∂_x
Backwards Difference, δ_{x-}	$(1 - e_{x-})/h$	
Backwards Average, μ_{x-}	$(1 + e_{x-})/2$	1
Temporal Operators	**Expression**	**Approximates**
Forwards Difference, δ_{t+}	$(e_{t+} - 1)/k$	$\partial_t, \frac{d}{dt}$
Backwards Difference, δ_{t-}	$(1 - e_{t-})/k$	
Centered Difference, $\delta_{t\cdot}$	$(e_{t+} - e_{t-})/(2k)$	
Forwards Average, μ_{t+}	$(e_{t+} + 1)/2$	1
Backwards Average, μ_{t-}	$(1 + e_{t-})/2$	
Centered Average, $\mu_{t\cdot}$	$(e_{t+} + e_{t-})/2$	
Fractional Derivative, $\delta_{t^{1/2}}$	$\sqrt{\frac{2}{k}}\frac{\sum_{r=0}^{M} b_r e_{t-}^r}{\sum_{r=0}^{M} a_r e_{t-}^r}$	$\partial_{t^{1/2}}$

3.2. Approximation of Fractional Derivatives

Implementation of fractional derivatives can be performed through the use of an IIR filter constructed using a truncated Continued Fraction Expansion (CFE) of the bilinear transform which is used to approximate the square root of $j\omega$ [17]

$$(j\omega)^{1/2} \approx \sqrt{\frac{2}{k}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)} \tag{12}$$

where $z = e^{j\omega k}$. The expansion of the numerator and denominator of this expression can then be used along with Viscovatov's algorithm [18] for computing the CFE. In essence, this algorithm requires adding 0 to each new level of the CFE which allows for factorisation of z^{-1} and inversion of the rest of the fraction. Neglecting the factor $\sqrt{2/k}$, the process for converting the expansion, whose numerator and denominator coefficients are $\beta_r^{(0)}$ and $\alpha_r^{(0)}$, of the bilinear transform to a CFE is as follows

$$\frac{\left(1 - z^{-1}\right)^{1/2}}{\left(1 + z^{-1}\right)^{1/2}} \approx \underbrace{\frac{\sum_{r=0}^{\infty} \beta_r^{(0)} z^{-r}}{\sum_{r=0}^{\infty} \alpha_r^{(0)} z^{-r}}}_{\text{Expansion}} + \underbrace{\frac{\beta_0^{(0)}}{\alpha_0^{(0)}} - \frac{\beta_0^{(0)}}{\alpha_0^{(0)}}}_{+0} \tag{13}$$

$$= \frac{\beta_0^{(0)}}{\alpha_0^{(0)}} + \frac{\sum_{r=1}^{\infty}\left(\beta_r - \frac{\beta_0^{(0)}}{\alpha_0^{(0)}}\alpha_r^{(0)}\right) z^{-r}}{\sum_{r=0}^{\infty}\alpha_r^{(0)} z^{-r}}$$

$$= \xi^{(0)} + \frac{j\omega}{\left[\frac{\sum_{r=0}^{\infty}\alpha_r^{(0)} z^{-r}}{\sum_{r=1}^{\infty}\left(\beta_r^{(0)} - \xi^{(0)}\alpha_r^{(0)}\right) z^{-(r-1)}}\right]}$$

where $\xi^{(0)} = \beta_0^{(0)}/\alpha_0^{(0)}$. For each iteration, i, of this process, the expansion coefficients within the square brackets can be rewritten as in the form of equation (13) by using $\beta_r^{(i)} = \alpha_r^{(i-1)}$, $\alpha_r^{(i)} = \beta_{r+1}^{(i-1)} - \xi^{(i-1)}\alpha_{r+1}^{(i-1)}$ and $\xi^{(i)} = \beta_0^{(i)}/\alpha_0^{(i)}$. Applying this process $2M$ times leads to a truncated CFE

$$(j\omega)^{1/2} \approx \xi^{(0)} + \cfrac{z^{-1}}{\xi^{(1)} + \cfrac{z^{-1}}{\ddots + \cfrac{z^{-1}}{\xi^{(2M)}}}}$$

The two lowest levels can be rewritten as

$$\frac{\xi^{(2M-1)}\xi^{(2M)} + z^{-1}}{\xi^{(2M)}} = \frac{b_0^{(1)} + b_1^{(1)} z^{-1}}{a_0^{(1)}}$$

which can be considered as a new series expansion with numerator and denominator coefficients $b_r^{(i)}$ and $a_r^{(i)}$. This can then be iteratively inverted to create a new polynomial fraction where $b_0^{(i)} = \xi^{(2M-i)} b_0^{(i-1)}$, $b_r^{(i)} = a_{r-1}^{(i-1)} + \xi^{(2M-i)} b_r^{(i-1)}$ for $r > 0$ and $a_r^{(i)} = b_r^{(i-1)}$ for $r \geq 0$.
Transforming into the time domain using $z^{-1} \rightarrow e_{t-}$ and reintroducing the factor $\sqrt{2/k}$ leads to a discrete approximation to the fractional derivative.

$$\delta_{t^{1/2}} = \sqrt{\frac{2}{k}} \frac{\sum_{r=0}^{M} b_r e_{t-}^r}{\sum_{r=0}^{M} a_r e_{t-}^r} \tag{14}$$

where b_r and a_r are the final iteration inversion of the CFE and have been normalised so that $a_0 = 1$. For sound synthesis at a sample rate of 44.1kHz, $M = 20$ is a suitable order filter for typical brass instrument geometries. This significantly increases the storage requirements of the algorithm as a whole, thus increasing the operation count and creating a bottleneck in performance.

3.3. Interleaved Grid and Difference Equations

For FDTD simulation of wave propagation in an acoustic tube it is useful to employ an interleaved time-space grid for the pressure and velocity fields, similar to work presented in electro-magnetics [19]. The pressure field is sampled at integer multiple time-space locations and the velocity field on the half integer points so that $p_l^n \approx p\left(lh, nk\right)$ and $v_{l+1/2}^{n+1/2} \approx v\left(\left(l+1/2\right)h, \left(n+1/2\right)k\right)$; see Fig. 4 at top. This leads to a representation in terms of $N+1$ pressure points and N velocity points, where $N = \text{floor}(L/h)$. The bore profile of the instrument must also be defined on both spatial grids. In this work the bore profile is sampled using linear interpolation on the velocity grid so that $S_{l+1/2} = S\left(\left(l+1/2\right)h\right)$. For the pressure grid the bore profile is defined as the spatial average of the neighbouring points on the velocity grid, $\bar{S}_l = 0.5\left(S_{l+1/2} + S_{l-1/2}\right)$; see Fig. 4 at bottom. Under these conditions, it can be shown that the Courant condition [20] must be satisfied for stability of simulations; that is $\lambda = ck/h$ where $\lambda \leq 1$ [7]. The different discretisations of the bore cross section mean that the functions in (3) must also be sampled: q is sampled on the pressure grid using $\bar{S}_l$, f and g are sampled on the velocity grid using $S_{l+1/2}$. See Table 2. For the valve sections, $S^{(d)}(x)$ is set from the location of the default tube in the main bore, although scaled by $q^{(d)}$. The default tube is therefore the path straight through the valve when the valve is not pressed. The basic profile of $S^{(b)}(x)$ is a cone whose entrance and exit areas are equal to the area of the main bore where the valve section begins and ends and has a length different to the default path so that the instrument's resonances are modified when waves propagate through this tube. The entrance and exit of the bypass tube are then scaled by $q^{(b)}$ to represent the constriction where the tube leaves and reenters the valve; the length of these constricted sections being equal to the length of the default tube. As the lengths of the tubes are distinct, the grid spacing must be chosen separately for the main, default and bypass sections; these are written as h_m, h_d and h_b respectively.

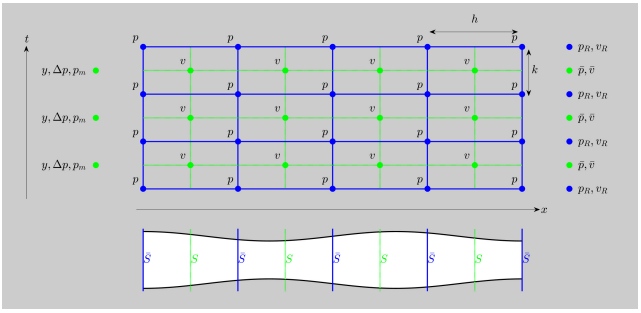

Figure 4: Top: Interleaved grid for pressure (blue lines and nodes), and velocity (green lines and nodes) fields with spatial and temporal step sizes of h and k. The network variables, p_R and v_R, are computed simultaneously with pressure values whereas $\bar{p}$ and $\bar{v}$ and the lip variables (y, Δp and p_m) are computed with the velocity variables. Bottom: Discretisation of bore profile with S at the velocity field locations and $\bar{S}$ at the pressure field locations.

The network variables in the radiation model are aligned with integer time steps whereas the main radiation variables and the lip model values are aligned with half integer time steps.

4. UPDATE EQUATIONS

Individual update equations can be derived by applying the difference operators in Table 1 to the system equations in Section 2. The operators can then be expanded and the equations rearranged to yield an update for the next time-step.

4.1. Wave Propagation Update

For the wave propagation in the bore, equations (2b) and (2a) become

$$\left(\rho\delta_{t-} + f_{l+1/2}\mu_{t-} + g_{l+1/2}\delta_{t^{1/2}}\mu_{t-}\right) v_{l+1/2}^{n+1/2} + \delta_{x+} p_l^n = 0 \tag{15a}$$

$$\left(\frac{\bar{S}_l}{\rho c^2}\delta_{t+} + q_l\delta_{t^{1/2}}\mu_{t+}\right) p_l^n + \delta_{x-}\left(S_{l+1/2} v_{l+1/2}^{n+1/2}\right) = 0 \tag{15b}$$

Due to the occurrences of the discrete fraction derivative operator $\delta_{t^{1/2}}$, it is then necessary to multiply through by $\sum_{r=0}^{M} a_r e_{t-}^r$ and then rearrange to get the update equations

$$v_{l+1/2}^{n+1/2} = \sum_{r=0}^{M} Q_{vv}^{(r)} e_{t-}^r v_{l+1/2}^{n-1/2} - Q_{vp}^{(r)} e_{t-}^r p_l^n \tag{16a}$$

$$p_l^{n+1} = \sum_{r=0}^{M} Q_{pp}^{(r)} e_{t-}^r p_l^n - Q_{pv}^{(r)} e_{t-}^r v_{l+1/2}^{n+1/2} \tag{16b}$$

Where the multiplying coefficients are defined in Table 2.

4.2. Valve Junctions Update

At the valve junctions, that is where the main tube meets the default and bypass tubes, equations (6a) and (6b) become

$$p_{N_J}^{(m)} = p_0^{(d)} = p_0^{(b)} = p_J \tag{17a}$$

$$\mu_{x-}\left(S_{N_J+1/2}^{(m)} v_{N_J+1/2}^{(m)}\right) = \mu_{x-}\left(S_{1/2}^{(d)} v_{1/2}^{(d)}\right) + \mu_{x-}\left(S_{1/2}^{(b)} v_{1/2}^{(b)}\right) \tag{17b}$$

where N_J denotes the index in the main tube where the valve lies. The following identity can be used to couple each tube together using the volume velocities

$$\delta_{x-} = \frac{2}{h}\left(1 - \mu_{x-}\right) = \frac{2}{h}\left(\mu_{x-} - e_{x-}\right) \tag{18}$$

Using this and that the pressure in all three tubes is the same at the valve junction, equation (15b) can be written as

$$p_J^{n+1} = \sum_{r=0}^{M} Q_{Jp}^{(r)} e_{t-}^r p_J^n + Q_{Jv}^{(r)} e_{t-}^r \left(S_{N_J+1/2}^{(m)} v_{N_J+1/2}^{(m)} - S_{1/2}^{(d)} v_{1/2}^{(d)} - S_{1/2}^{(b)} v_{1/2}^{(b)}\right) \tag{19}$$

at the valve boundary. When the default and bypass tubes combine back into the main tube, the sign of $Q_{Jv}^{(r)}$ is inverted. See Table 3 for coefficient definitions. Fig. 5 shows simulated wave propagation within a trumpet with partially open valves. When the wave encounters the valve junction it splits as it travels through the default and bypass tubes.

4.3. Radiation Update

The network variables in equations (9a), (9b) and (9c) become

$$\bar{v}^{n+1/2} = \mu_{t+} v_R^n + \left(\frac{\mu_{t+}}{\Gamma \rho c} + \frac{\Theta a}{\rho c^2}\delta_{t+}\right) p_R^n \tag{20a}$$

$$\bar{p}^{n+1/2} = \Lambda \rho a \delta_{t+} v_R^n \tag{20b}$$

$$\bar{p}^{n+1/2} = \left(\left(1 + \frac{1}{\Gamma}\right)\mu_{t+} + \frac{\Theta a}{c}\delta_{t+}\right) p_R^n \tag{20c}$$

where $\bar{S}_N \bar{v}^{n+1/2} = \mu_{x-}\left(S_{N+1/2} v_{N+1/2}^{n+1/2}\right)$ and $\bar{p}^{n+1/2} = \mu_{t+} p_N^n$. The updates for v_R and p_R can be written in terms of unknown pressure values at the end of the instrument

$$v_R^{n+1} = v_R^n + \frac{k}{2\Lambda\rho a}\left(p_N^{n+1} + p_N^n\right) \tag{21a}$$

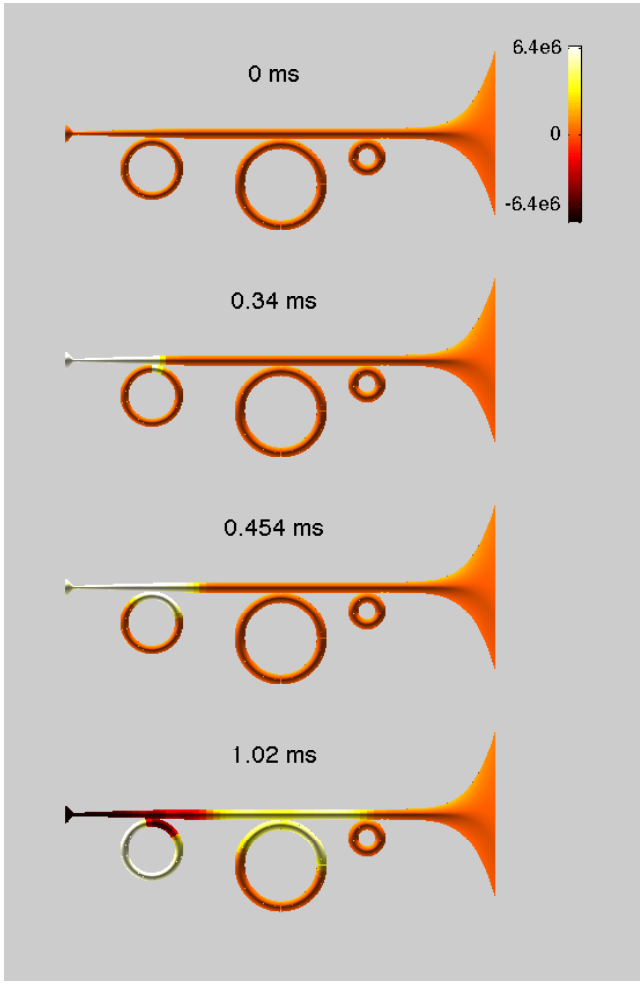

Figure 5: Pressure wave propagation within a valved brass instrument at several time steps simulated at 44.1 kHz excited with a raised cosine pulse and with all valves in a half-open configuration, at times as indicated, and illustrating splitting of traveling waves at valve junctions.

$$p_R^{n+1} = \left[\frac{1}{2}\left(1 + \frac{1}{\Gamma}\right) + \frac{\Theta a}{ck}\right]^{-1} \left(\frac{p_N^{n+1} + p_N^n}{2} + \left[\frac{\Theta a}{ck} - \frac{1}{2}\left(1 + \frac{1}{\Gamma}\right)\right] p_R^n\right) \tag{21b}$$

These relations along with the second identity in equation (18) can be used to update equation (15b) at the radiating end of the instrument. It is sufficient to use the lossless version of this equation (neglecting the fractional derivative term) to get

$$p_N^{n+1} = Q_{rp} p_N^n + Q_{rv} v_{N-1/2}^{n+1/2} + Q_{rv_R} v_1^n + Q_{rp_R} p_R^n \tag{22}$$

See Table 4 for radiation update coefficients.

4.4. Lip Reed Update

Equations (10a) to (10e) can be written as

$$\left(\delta_{t+}\delta_{t-} + \sigma\delta_{t\cdot} + \omega_0^2\mu_{t\cdot}\right) y^{n+1/2} = \frac{S_r \Delta p^{n+1/2}}{\mu_r} \tag{23a}$$

$$\Delta p^{n+1/2} = p_m^{n+1/2} - \mu_{t+} p_0^n \tag{23b}$$

$$u_m^{n+1/2} = w\left[y^{n+1/2} + H\right]_+ \sqrt{\frac{2|\Delta p^{n+1/2}|}{\rho}} \operatorname{sign}\left(\Delta p^{n+1/2}\right) \tag{23c}$$

$$u_r^{n+1/2} = S_r \delta_{t\cdot} y^{n+1/2} \tag{23d}$$

$$\mu_{x-}\left(S_{1/2} v_{1/2}^{n+1/2}\right) = u_m^{n+1/2} + u_r^{n+1/2} \tag{23e}$$

Equation (23a) becomes

$$y^{n+3/2} = \left(Q_{y1} y^{n+1/2} + Q_{y2} y^{n-1/2} + Q_{yp} \Delta p^{n+1/2}\right) \tag{24}$$

The lip model can be coupled to the instrument in a similar manner to a clarinet reed [21]. Equation (23a) can be rewritten in terms of $\delta_{t\cdot} y^{n+1/2}$ and $\Delta p^{n+1/2}$ using

$$\delta_{t+}\delta_{t-} = \frac{2}{k}\left(\delta_{t\cdot} - \delta_{t-}\right), \quad \mu_{t\cdot} = k\delta_{t\cdot} + e_{t-} \tag{25}$$

Then using equations (23c), (23d) and (23e) leads to an expression in terms of $\Delta p^{n+1/2}$ and $\mu_{x-}\left(S_{1/2} v_{1/2}^{n+1/2}\right)$.

At $l = 0$ equation (16b) can be rewritten in terms of $\Delta p^{n+1/2}$ and $\mu_{x-}\left(S_{1/2} v_{1/2}^{n+1/2}\right)$ by using the first identity in equation (18) along with

$$\delta_{t+} = \frac{2}{k}\left(\mu_{t+} - 1\right) \tag{26}$$

and equation (23b) when the fractional derivatives are neglected. This then combines to create the quadratic expression

$$|\Delta p^{n+1/2}| + \frac{d_3^n}{d_2 + c_1}\sqrt{|\Delta p^{n+1/2}|} - \left|\frac{d_1^n - c_2^n}{d_2 + c_1}\right| = 0 \tag{27}$$

provided

$$\operatorname{sign}\left(\Delta p^{n+1/2}\right) = -\operatorname{sign}\left(\frac{d_1^n - c_2^n}{d_2 + c_1}\right)$$

The value of the pressure difference can then be used to update the lip position and pressure at the input of the acoustic tube.

$$p_0^{n+1} = 2\left(p_m^{n+1/2} - \Delta p^{n+1/2}\right) - p_0^n \tag{28}$$

See Table 5 for lip update coefficients.

5. ENVIRONMENT DEVELOPMENT AND OPTIMISATION

The brass environment was originally developed in the MATLAB prototyping platform [22] and then ported to the C programming language. Input to the environment consists of instrument and score files; output is in .wav format, and is drawn from the pressure at the radiating end of the instrument. Bore profile definition is in the hands of the user, either through manual specification of coordinate/radius pairs, or using a simplified set of concatenated sections; valve positions and lengths of the default and bypass tubes may also be supplied. The score file consists of breakpoint functions which specify the time variation of mouth pressure, lip parameters and valve openings.

For a single-valved trumpet of length 1.381m, with default and bypass tube lengths of 0.02m and 0.2m, the MATLAB code takes approximately 14s to generate 1 second of output on an Intel Core i5-4300U running at 2.5GHz. See Table 6 for a list of optimisation methods and respective accelerations. A direct translation from MATLAB to C, using the same algorithms and data structures, results in a 5x speed up—typical of a MATLAB-to-C conversion. To further improve performance, two methods of parallel execution

Table 2: *Coefficients in wave propagation update.*

$$Q_{vv}^{(r)} = \frac{2\rho(a_r - a_{r+1}) - k\left(f_{l+1/2}(a_r + a_{r+1}) + g_{l+1/2}(b_r + b_{r+1})\right)}{\left(2\rho + k f_{l+1/2}\right) a_0 + k g_{l+1/2} b_0}$$

$$Q_{vp}^{(r)} = \frac{2k a_r \delta_{x+}}{\left(2\rho + k f_{l+1/2}\right) a_0 + k g_{l+1/2} b_0}$$

$$Q_{pp}^{(r)} = \frac{2\bar{S}_l\left(a_r - a_{r+1}\right) - \rho c^2 k q_l \left(b_r + b_{r+1}\right)}{2\bar{S}_l a_0 + \rho c^2 k q_l b_0}$$

$$Q_{pv}^{(r)} = \frac{2\rho c^2 k a_r \delta_{x-}\left(S_{l+1/2}\,\cdot\right)}{2\bar{S}_l a_0 + \rho c^2 k q_l b_0}$$

$$f_{l+1/2} = \frac{3\eta\pi}{S_{l+1/2}},\ g_{l+1/2} = 2\sqrt{\frac{\rho\eta\pi}{S_{l+1/2}}},\ q_l = \frac{2\left(\gamma - 1\right)}{\nu c^2}\sqrt{\frac{\eta\pi\bar{S}_l}{\rho^3}}$$

Table 3: *Coefficients in valve junction update.*

$$\begin{aligned} Q_{Jp}^{(r)} = \ & Q_{J0}\left[\left(h_m + h_d q^{(d)} + h_b q^{(b)}\right)\frac{\bar{S}_{N_J}}{2\rho c^2 k}\left(a_r - a_{r+1}\right)\right. \\ & \left. -\left(h_m + h_d\sqrt{q^{(d)}} + h_b\sqrt{q^{(b)}}\right)\frac{\gamma - 1}{2\nu c^2}\sqrt{\frac{\eta\pi\bar{S}_{N_J}}{\rho^3}}\left(b_r + b_{r+1}\right)\right] \\ Q_{Jv}^{(r)} = \ & a_m Q_{J0} \\ Q_{J0} = \ & \left[\left(h_m + h_d q^{(d)} + h_b q^{(b)}\right)\frac{\bar{S}_{N_J}}{2\rho c^2 k} a_0\right. \\ & \left. + \left(h_m + h_d\sqrt{q^{(d)}} + h_b\sqrt{q^{(b)}}\right)\frac{\gamma - 1}{2\nu c^2}\sqrt{\frac{\eta\pi\bar{S}_{N_J}}{\rho^3}} b_0\right]^{-1} \end{aligned}$$

Table 4: *Coefficients in radiation update.*

$$Q_{rp} = \frac{1 - Q_{r0}}{1 + Q_{r0}} \quad Q_{rv} = \frac{2\rho c^2 k S_{N-1/2}}{\bar{S}_N h\left(1 + Q_{r0}\right)}$$

$$Q_{rv_R} = -\frac{2\rho c^2 k}{h\left(1 + Q_{r0}\right)}$$

$$\begin{aligned} Q_{rp_R} = Q_{rv_R}\Bigg(\frac{1}{2\Gamma\rho c} - \frac{\Theta a}{\rho c^2 k} + \left(\frac{1}{2\Gamma\rho c} + \frac{\Theta a}{\rho c^2 k}\right) \times \\ \left[\frac{1}{2}\left(1 + \frac{1}{\Gamma}\right) + \frac{\Theta a}{c k}\right]^{-1}\left(\frac{\Theta a}{c k} - \frac{1}{2}\left(1 + \frac{1}{\Gamma}\right)\right)\Bigg) \end{aligned}$$

$$\begin{aligned} Q_{r0} = \frac{\rho c^2 k}{h}\Bigg(\frac{k}{2\Lambda\rho a} + \\ \left(\frac{1}{2\Gamma\rho c} + \frac{\Theta a}{\rho c^2 k}\right)\left[\frac{1}{2}\left(1 + \frac{1}{\Gamma}\right) + \frac{\Theta a}{c k}\right]^{-1}\Bigg) \end{aligned}$$

Table 5: *Coefficients in lip update.*

$$Q_{y1} = Q_{y0}\frac{2}{k^2}, \quad Q_{y2} = Q_{y0}\left(\frac{\sigma}{2k} - \frac{1}{k^2} - \frac{\omega_0^2}{2}\right)$$
$$Q_{yp} = Q_{y0}\frac{S_r}{\mu_r}, \quad Q_{y0} = \left(\frac{1}{k^2} + \frac{\sigma}{2k} + \frac{\omega_0^2}{2}\right)^{-1}$$
$$d_1^n = S_r\frac{\left(2\delta_{t-} - k\omega_0^2 e_{t-}\right)y^{n+1/2}}{2+\sigma k+\omega_0^2 k^2}, \quad d_2 = \frac{kS_r^2}{\mu_r\left(2+\sigma k+\omega_0^2 k^2\right)}$$
$$d_3^n = w|y^{n+1/2} + H|_+\sqrt{\frac{2}{\rho}}$$
$$c_1 = \frac{\bar{S}_0 h}{\rho c^2 k}, \quad c_2^n = S_{1/2}v_{1/2}^{n+1/2} + c_1\left(p_m - p_0^n\right)$$

have been considered along with modification to the set up of the control stream and are presented below.

5.1. GPU Acceleration

Use of NVIDIA GPUs for simulation of other instrument systems programmed using the CUDA platform [23] has shown significant performance increases by solving different parts of the problem on individual cores [24, 25].
These methods are extremely effective for large scale problems, such as e.g., systems defined in 2D and 3D, but in the case of the small 1D brass instrument system the overheads required to transfer data between GPUs results in performance that is 4.7x slower than serial C when run on Tesla K20c GPUs.

5.2. Vectorisation

Modern CPUs contain powerful vector units capable of performing multiple floating point or integer operations with a single instruction (a programming model known as Single Instruction Multiple Data). This gives parallelism at a much finer-grained level than the method described above, and because the program is still running as a single thread on one CPU core, the synchronisation and data transfer bottlenecks of GPU methods can be avoided. Although compilers can sometimes automatically vectorise code, it is often still necessary to vectorise manually to get the best results, especially for more complex operations. Manual vectorisation involves using compiler intrinsics (special functions that map directly to machine instructions) or assembly language to program the vector unit directly.
The brass code was vectorised using the AVX instructions available on modern Intel and AMD CPUs [26]. The AVX unit provides 256-bit wide vector registers capable of storing 4 double precision floating point values, and parallel execution units capable of operating on all 4 values at once. The inner loops of the pressure and velocity updates for the main and bypass tubes were vectorised, as these are by far the most time consuming elements of the code. The default tube updates were not vectorised as the default tube is generally very small and takes very little time to process even in serial. This more than doubles performance relative to serial C and is 11.5x faster than MATLAB.

5.3. Interpolation of Control Stream

A final optimisation for both the serial and AVX versions of the code involves interpolating control stream values for the lip and valve inputs on-the-fly at each time step instead of pre-computing them all at startup and storing them in arrays. The intention here is primarily to save memory, but in fact it also has the effect of significantly speeding up both versions of the code: the serial C performance for the trumpet became 7.1x faster than the original MATLAB, and the AVX performance increased to 16.3x faster than MATLAB. These improvements are due to a reduction in the amount of data read from the main memory at each time step, relieving traffic on the memory bus which is often a bottleneck on modern systems.

Table 6: *Run times and speed increases for different optimisation methods for 1s of output sound using a trumpet bore of length* 1.381m*, with a single valve with default tube length* 0.02m *and bypass tube length* 0.2m*. Test instrument and score files are available online along with final environment binaries. Times were taken from a machine with Intel Core i5-4300U except for those on GPU which were run on Tesla K20c GPUs.*

Code version	Run time (s)	Speed-up over MATLAB
MATLAB	14.02	1x
Serial C	2.69	5.2x
GPU	12.75	1.1
AVX	1.22	11.5x
Serial C (memory optimised)	1.98	7.1
AVX (memory optimised)	0.86	16.3x

6. CONCLUSIONS AND FUTURE WORK

The environment binaries are available online at `http://www2.ph.ed.ac.uk/~s0916351/dafx15.html`, along with user documentation. This work is still in the early stages of development and there are multiple extensions planned to improve performance and to add additional features. The dominant time-intensive feature of the algorithm lies with the approximation of the fractional derivatives, which themselves are features of the approximations to the impedance and admittance formulae. It would therefore be of interest to lower the order of this filter. Thompson *et al* [27] present an analog filter structure to model the viscous and thermal losses present in an acoustic tube that does not require the use of fractional derivatives. Although their method still requires a relatively high order of time steps to be stored, optimisation techniques, such as those used by Abel *et al* [28], can be used to create lower order filter designs; see [29] for preliminary work on this problem.
Due to the (usually) short length of the default tubes, when discretising these sections the spacing, h, is increased which leads to a reduction in the bandwidth over these sections; for normal instrument geometries this is up to about 16kHz. This could be improved through interpolation of the position of the valve junctions.
Currently, the output from the environment is a mono sound file that is taken from the very end of the instrument. To add spatialisation to the produced sound it would be possible to embed the instrument within a 3D space by replacing the current radiation model with an energy coupling between the instrument and room, similar to work that has been done with percussion instruments [24, 25]. In this case the room simulation can be performed using GPUs, with potentially very large acceleration.
The basic design described here could be used in order to simulate a clarinet by a small alteration to the excitation mechanism (to

simulate a reed rather than a lip), and in replacing valve sections by toneholes. The excitation model could also be extended to include collisions between the lips or the reed and lay. See [30] for some preliminary results.

A more fundamental alteration to the environment would be to model nonlinear wave propagation within the instrument bore. These processes are responsible for the characteristic "brassy" or "cuivré" sound present when instruments such as the trumpet and trombone are played at high dynamic levels [31, 32]. These systems, however, require entirely new numerical design techniques in order to cope with severe stability issues as shocks develop at the wave front [33].

7. ACKNOWLEDGMENTS

This work was supported by the European Research Council, under grant number StG-2011-279068-NESS.

8. REFERENCES

[1] F. Silva, C. Vergez, P. Guillemain, J. Kergomard, and V. Debut, "Moreesc: A framework for the simulation and analysis of sound production in reed and brass instruments," *Acta Acust united Ac*, vol. 100, pp. 126–138, 2014.

[2] P. Cook, "Tbone: An interactive waveguide brass instrument synthesis workbench for the next machine," in *Proc. of the Int. Computer Music Conference*, Montreal, Canada, 1991, pp. 297–299.

[3] T. Hélie, R. Mignot, and D. Matignon, "Waveguide modeling of lossy flared acoustic pipes: Derivation of a kelly-lochbaum structure for real-time simulations," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, October 2007.

[4] R. Mignot, T. Hélie, and D. Matignon, "Digital waveguide modeling for wind insturments: Building a state-space representation based on webster-lokshin model," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 4, pp. 843–854, 2010.

[5] C. Vergez, *Trompette et Trompettiste: Un Système Dynamique Non Linéaire à Analyser, Modéliser et Simuler dans un Contexte Musical*, Ph.D. thesis, Universite Paris 6, 2000.

[6] C. Vergez and P. Tisserand, "The brass project, from physical models to virtual musical instruments: Playability issues," in *Lecture Notes in Computer Science: Computer Music Modeling and Retrieval*, R. Kronland-Martinet, T. Voinier, and S. Ystad, Eds., vol. 3902, pp. 24–33. Springer, Berlin/Heidelberg, Germany, September 2006.

[7] S. Bilbao and J. Chick, "Finite difference time domain simulation for the brass instrument bore," *J Acoust Soc Am*, vol. 134, no. 5, pp. 3860–3871, 2013.

[8] R. L. Harrison and J. Chick, "A single valve brass instrument model using finite-difference time-domain methods," in *Proc. of the Int. Symp. on Musical Acoustics.*, Le Mans, France, 2014.

[9] C. Zwikker and C.W. Kosten, *Sound absorbing materials*, Elsevier Pub. Co., 1949.

[10] D. H. Keefe, "Acoustical wave propagation in cylindrical ducts: Transmission line parameter approximations for isothermal and nonisothermal boundary conditions," *J Acoust Soc Am*, vol. 75, no. 1, pp. 58–62, 1984.

[11] T. Hélie, "Unidimensional models of acoustic propagation in axisymmetric waveguides," *J Acoust Soc Am*, vol. 114, no. 5, pp. 2633–2647, 2003.

[12] A. H. Benade, "On the propagation of sound waves in a cylindrical conduit," *J Acoust Soc Am*, vol. 44, no. 2, 1968.

[13] S. Bilbao, "Modelling of brass instrument valves," in *Proc. of the 14^{th} Int. Conference on Digital Audio Effects*, Paris, France, September 2011, pp. 337–343.

[14] H. Levine and J. Schwinger, "On the radiation of sound from an unflanged circular pipe," *Phys Rev*, vol. 73, no. 4, pp. 383–406, 1948.

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

[16] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments, Second Edition*, Springer, 1998.

[17] Y. Chen, B. M Vinagre, and I. Podlubny, "Continued fraction expansion approaches to discretizing fractional order derivatives - an expository review," *Nonlinear Dynamics*, vol. 38, pp. 115–170, 2004.

[18] A. Cuyt, *Handbook of continued fractions for special functions. [electronic resource].*, [Dordrecht, Netherlands] : Springer, [2008], ©2008., 2008.

[19] K. S. Yee, "Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media," *IEEE Trans. Antennas Propag.*, vol. 14, no. 3, pp. 302–307, 1966.

[20] R. Courant, K. Friedrichs, and H. Lewy, "On the partial differential equations of mathematical physics," *Mathematische Annalen*, vol. 100, pp. 32–74, 1928.

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

[22] MathWorks, "Matlab - the language of technical computing," October 2014, Last accessed October 2014.

[23] J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable parallel programming with cuda," *Queue*, vol. 6, no. 2, pp. 40–53, 2008.

[24] S. Bilbao and C. Webb, "Timpani drum synthesis in 3d on gpgpus," in *Proc. of the 15^{th} Int. Conference on Digital Audio Effects*, York, UK, September 2012.

[25] A. Torin and S. Bilbao, "A 3-d multi-plate environment for sound synthesis," in *Proc. of the 16^{th} Int. Conference on Digital Audio Effects*, Maynooth, Ireland, September 2013.

[26] N Firasta, M. Buxton, P. Jinbo, Ka. Nasri, and S. Kuo, "Intel® avx : New frontiers in performance improvements and energy efficiency," *Intel White Paper*, 2008, Available online https://software.intel.com/en-us/articles/intel-avx-new-frontiers-in-performance-improvements-and-energy-efficiency.

[27] S. C. Thompson, T. B. Gabrielson, and D. M. Warren, "Analog model for thermoviscous propagation in a cylindrical tube," *J Acoust Soc Am*, vol. 135, no. 2, pp. 585–590, 2014.

[28] J. Abel, T. Smyth, and J. O. Smith III, "A simple, accurate wall loss filter for acoustic tubes," in *Proc. of the 6^{th} Int. Conference on Digital Audio Effects*, London, UK, September 2003.

[29] S. Bilbao, R. Harrison, J. Kergomard, B. Lombard, and C. Vergez, "On Passive Models of Viscothermal Wave Propagation in Acoustic Tubes," *J Acoust Soc Am*, vol. 138, no. 2, pp. 555–558, 2015.

[30] S. Bilbao, A. Torin, and V. Chatziioannou, "Numerical modeling of collisions in musical instruments," *Acta Acust united Ac*, vol. 101, pp. 292–299, 2015.

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

[32] J. Gilbert, L. Menguy, and M. Campbell, "A simulation tool for brassiness studies(l)," *J Acoust Soc Am*, vol. 123, no. 4, pp. 1854–1857, 2008.

[33] O. Richoux, B. Lombard, and J-F. Mercier, "Generation of acoustic solitary waves in a lattice of Helmholtz resonators," *Wave Motion*, vol. 56, pp. 85–99, 2015.