Aerophones in Flatland: Interactive Wave Simulation of Wind Instruments

Andrew Allen Nikunj Raghuvanshi
Microsoft Research

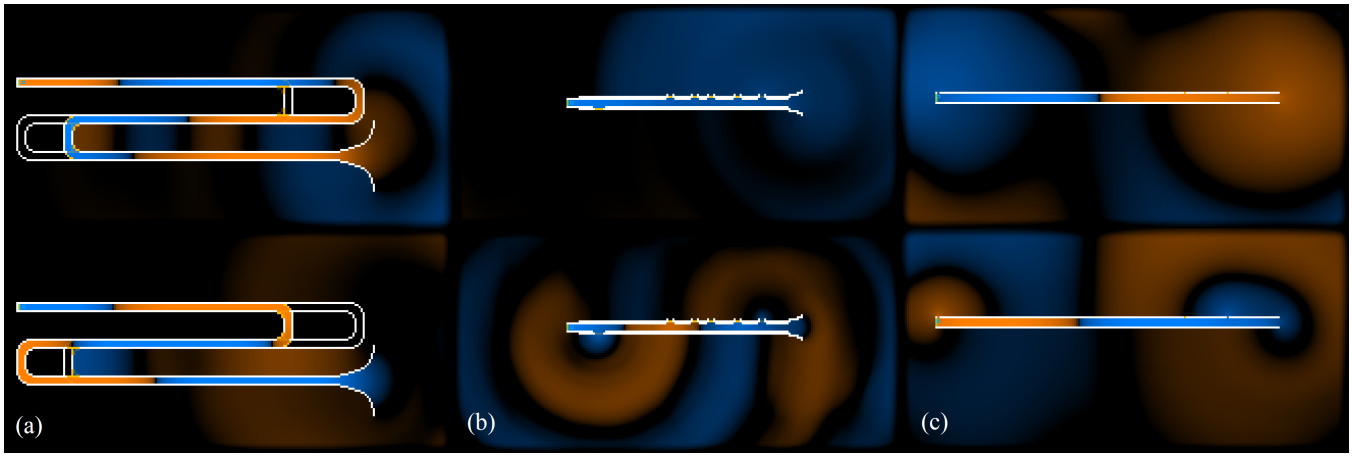


Figure 1: Wave fields for 2D wind instruments simulated in real-time on a graphics card. A few examples are shown, which are simplified virtual models of (a) trumpet, (b) clarinet, and (c) flute. Our interactive wave solver lets the user design and instantly perform such virtual instruments, promoting experimentation with novel designs. Dynamic changes such as opening and closing tone holes or manipulating valves automatically changes the resulting sound and radiation pattern. Synthesized musical notes can be heard in the accompanying demonstrations.

Abstract

We present the first real-time technique to synthesize full-bandwidth sounds for 2D virtual wind instruments. A novel interactive wave solver is proposed that synthesizes audio at 128,000Hz on commodity graphics cards. Simulating the wave equation captures the resonant and radiative properties of the instrument body automatically. We show that a variety of existing non-linear excitation mechanisms such as reed or lips can be successfully coupled to the instrument's 2D wave field. Virtual musical performances can be created by mapping user inputs to control geometric features of the instrument body, such as tone holes, and modifying parameters of the excitation model, such as blowing pressure. Field visualizations are also produced. Our technique promotes experimentation by providing instant audio-visual feedback from interactive virtual designs. To allow artifact-free audio despite dynamic geometric modification, we present a novel time-varying Perfectly Matched Layer formulation that yields smooth, natural-sounding transitions between notes. We find that visco-thermal wall losses are crucial for musical sound in 2D simulations and propose a practical approximation. Weak non-linearity at high amplitudes is incorporated to improve the sound quality of brass instruments.

CR Categories: H.5.5 [Information Interfaces and Presentation]: Sound and music computing—modeling; G.1.8 [Numerical Analysis]: Partial differential equations—finite difference methods; I.3.1 [Computer Graphics]: Hardware architecture—graphics processors

Keywords: wind instruments, wave equation, radiation, scattering, graphics processor (GPU), sound synthesis

1 Introduction

Making wind instruments (aerophones) is an innate human activity dating back at least 20,000 years [Baines 1967]. Wind instruments are non-linear dynamical systems with an active excitation mechanism, such as a trumpet player's buzzing lips, undergoing coupled oscillation with the resonant cavity formed by the body of the instrument. This two-way coupling is essential to their operation, irreducible to a feed-forward model. For instance, the sound of oscillating lips filtered through a trumpet's resonant acoustic response results in a comb-filtered buzzing sound, not a steady musical note. The complex physics underlying their behavior has naturally invited enduring curiosity from physicists [Helmholtz 1885]. Ever since the rise of digital computers, the musical acoustics community has held sustained interest in performing direct physical simulation to create virtual instruments. In computer graphics, physically-based sound synthesis and propagation have seen rising interest over the past couple decades, with the goal of modeling sounds that are perceptually similar to reality. Our work shares this motivation: while the simulations are based closely on physical principles, the primary objective is to actively support user interaction and synthesize sounds that resemble everyday experience.

Existing real-time virtual wind instruments are based on Digital Waveguides and related techniques [Smith 2010]. The essential idea is that constant-speed wave propagation in one dimension can be expressed as a superposition of two opposing traveling waves, implementable efficiently as digital delay lines. Assuming propagating wavefronts are planar or spherical (infinite cylindrical or

ACM Reference Format
Allen, A., Raghuvanshi, N. 2015. Aerophones in Flatland: Interactive Wave Simulation of Wind Instruments. ACM Trans. Graph. 34, 4, Article 134 (August 2015), 11 pages. DOI = 10.1145/2767001 http://doi.acm.org/10.1145/2767001.

Copyright Notice
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGGRAPH '15 Technical Paper, August 09 – 13, 2015, Los Angeles, CA.
Copyright 2015 ACM 978-1-4503-3331-3/15/08 ... $15.00.
DOI: http://doi.acm.org/10.1145/2767001

conical bores respectively), wind instruments are reduced to circuits of such delay lines. Externally-specified filters at delay line junctions compensate for important wave effects that one dimensional wave propagation lacks inherently, such as scattering and radiation losses at tone holes and the bell (open end). A given instrument geometry is conceptually broken into segments with roughly constant propagation impedance to build an analogous virtual instrument circuit. Coefficients for the filters at circuit junctions are determined from analytic approximation, offline simulation or experiments. Stability and realizability of the resulting circuit have to be ensured carefully. The resulting flexible signal processing framework has been used to create many virtual instrument models with compelling sounds [Smith 1996; Smith 2004].

However, constructing the digital waveguide circuit for a given instrument geometry is a manual process requiring significant expertise in acoustic signal processing. An alternative approach that we explore in this paper is to directly simulate the wave equation on the instrument geometry. All linear acoustic phenomena including radiation and scattering are modeled automatically, thus obviating the need for externally specified filters or manual analysis of geometry for reduction to a 1D circuit. Wave simulation on geometry also constrains exploration to physically plausible sounds, unlike manipulation of filter coefficients. Unfortunately, such wave simulation is extremely compute-hungry. Resolving wind instruments' geometric features and dynamics requires spatial and temporal resolutions in the millimeter and microsecond range respectively. Consequently, no work exists on such real-time wave simulation.

We present the first real-time technique to synthesize audio for virtual wind instruments using 2D wave simulation of the resonator for all audible frequencies. As shown in Figure 1, these simulations naturally model changes in sound and radiation pattern due to shape modification. For instance, opening a tone hole automatically changes the local radiation and scattering characteristics. This affects the net frequency-dependent impedance at the mouthpiece, causing a change in musical note. Our interactive system lets the user edit shape to create and modify instrument profile, tone holes, valve systems, flaring bells, or mutes. The designed virtual instruments can be performed by mapping user input from a keyboard or any digital controller to excitation parameters such as breath pressure, and geometric modification, such as opening a valve in a trumpet-like instrument. The instant audio-visual feedback during design and performance promotes experimentation with virtual instruments. Besides creative applications, the interactive environment is useful for educational purposes. The generated audio-visuals can also be a valuable tool for studying instrument physics in a simplified setting.

Our interactive technique is enabled by a novel wave solver based on Finite-Difference Time-Domain (FDTD) that runs on commodity graphics cards simulating audio at 128,000Hz. We introduce a time-varying Perfectly Matched Layer (PML) formulation that supports on-the-fly shape modification without generating auditory artifacts. Each simulation cell is allowed to smoothly change between solid and open states with user-controlled transition time. This results in natural note transitions when performing the instrument which is critical to believable music synthesis. We show that modeling visco-thermal losses at instrument walls is essential in multi-dimensional wave simulation and propose a practical approximation suitable for real-time evaluation. Absence of such modeling can result in overwhelming high-frequency parasitic resonant modes. Weakly non-linear propagation at high amplitudes is also modeled using an efficient formulation to improve the sound quality of brass instruments. Finally, we demonstrate that a variety of non-linear excitation models from existing 1D virtual instrument literature can be successfully coupled to our 2D wave fields, resulting in natural sounding simulations.

2 Related Work

Musical acoustics is a vast area of research with rich literature. We limit discussion to closely related work below. For a general overview of the area, we refer the reader to the well-regarded text [Fletcher and Rossing 1998]. For a review of more recent research on wind instrument physics, see [Fabre et al. 2012]. We discuss literature on particular physical aspects relevant to our work alongside technical discussion in Sections 3, 4 and 5.

2.1 Real-time virtual instruments

McIntyre, Schumacher and Woodhouse [1983] are generally credited for introducing the framework for fast digital sound synthesis of wind instruments: an active, non-linear excitation mechanism coupled in a feedback loop with a passive resonator with linear response. Assuming a fixed instrument configuration (e.g., pattern of open/closed tone holes), the resonator is represented by a time-invariant impulse response that relates the excitation's input at the bore entrance to the resulting feedback of the instrument air column at the same point. This affords time-domain modeling of the oscillatory phenomena essential to wind instrument sound production. This technique was used to create the first demonstrations of physical modeling synthesis for a variety of wind instruments.

Around the same time, Smith [1986] introduced the highly influential digital waveguide technique. Assuming cylindrical or conical bore profiles, real-time synthesis could be achieved using digital delay lines that can be implemented efficiently as shift buffers. Since 1D wave propagation doesn't include scattering and radiation, externally-determined filters at delay line junctions are specified to compensate. This effectively allowed expressing the resonator impulse response in a much more flexible digital circuit form. It was shown later in [Scavone and Smith 1996] that by carefully modifying coefficients for tone hole radiation and scattering filters, a time-varying impulse response could be achieved, resulting in the sound of the instruments being performed. Digital waveguides and closely related techniques such as wave digital filters [van Walstijn and Campbell 2003] today form the basis for a large body of real-time techniques for 1D time-domain simulation. Surveys of the area can be found in [Smith 1996; Smith 2004; Välimäki et al. 2006; van Walstijn 2007] and a discussion of the underlying techniques in [Smith 2010; Cook 2002].

2.2 Physical modeling

Real-time virtual instruments represent only a small part of the much larger research area of physical modeling of musical instruments. Digital waveguide modeling was one of the first techniques to afford computationally feasible time-domain simulation and has served as a natural tool for such digital experiments. The primary goal of instrument physics research, however, is to design accurate models that closely match the acoustic characteristics of real instruments, in order to advance scientific understanding. For gradually-varying cross sections, 1D finite difference models have been proposed that can accurately model continuous effects such as wall losses and dispersion [Bilbao 2009], although fast-flaring horns used in brass instruments are troublesome for 1D modeling in general. Nevertheless, such models have been used to impressive effect to perform offline synthesis of brass instrument sounds [Bilbao and Chick 2013]. With advancing computational power, offline 3D simulation is being employed increasingly. To pick a few examples, there has been work on elucidating the dynamics of the air jet in flue instruments using 3D Navier-Stokes solution [Bader 2013; Giordano 2014], and detailed simulation studies of the fluid-structure interaction in single reeds [Ricardo et al. 2007]. Frequency-domain 3D FEM simulation is employed in [Lefebvre and Scavone 2012]

Symbol	Meaning	Value
ρ	mean density	$1.1760\,(1 - 0.00335\Delta T)$
c	speed of sound	$3.4723 \times 10^2(1 + 0.00166\Delta T)$
γ	adiabatic index	$1.4017\,(1 - 0.00002\Delta T)$
μ	dynamic viscosity	$1.8460 \times 10^{-5}(1 + 0.0025\Delta T)$
$\mathcal{P}$	Prandtl number	$0.7073\,(1 - 0.0004\Delta T)$
j	imaginary unit	$\sqrt{-1}$
p	pressure	–
$\mathbf{v}$	particle velocity	–
$\mathbf{n}$	surface unit normal	–
Δ_t	simulation time step	7.81×10^{-6}
Δ_s	simulation cell size	3.83×10^{-3}

Table 1: Symbols and meanings. All values are in SI units. Physical constant values are for air at reference temperature of $26.85°\mathrm{C}\,(300°K)$, taken from [Keefe 1984]. ΔT is temperature difference from reference, and values are accurate for $|\Delta T| < 10$.

to study the acoustic characteristics of individual woodwind tone holes with the intent of integrating the resulting filters into real-time digital waveguide simulations. There has also been work on offline numerical shape optimization of musical instruments, such as an improved clarinet [Noreland et al. 2013] and a more melodious trumpet [Macaluso and Dalmont 2011].

2.3 Wave simulation and graphics processors (GPUs)

Limited work exists on real-time wave simulation for full audible bandwidth in more than one dimension. With thousands of lightweight cores, GPUs are a natural fit for parallel wave solvers. Offline band-limited room acoustics calculations have attracted increased interest recently [Savioja et al. 2010; Hamilton and Webb 2013] and more generally for auralization [Tsingos et al. 2011]. Real-time 3D FDTD simulation of a small room is presented in [Savioja 2010]. Simulations are limited to a usable bandwidth of $\nu_m = 1.5$kHz, typical for current state-of-the-art time-domain room acoustic solvers. GPUs for sound synthesis have also gained interest recently [Bilbao et al. 2013]. Specifically, there has been work on offline sound synthesis for timpani drums [Webb 2014]. Simulation on coarse-resolution 2D drum membranes is presented in [Hsu and Pérez 2013], we report much higher performance by better utilizing GPU parallelism as discussed in section 6.1.

2.4 Interactive applications

There is a large body of literature in computer graphics for visual simulation of physical phenomena. Over the last decade sound synthesis has received increasing attention, such as the sound of water [Zheng and James 2009] and fire [Chadwick and James 2011]. Rigid body sounds have attracted considerable interest in particular, especially using precomputed modal analysis [O'Brien et al. 2002; Chadwick et al. 2009; Chadwick et al. 2012; Langlois et al. 2014]. Auralizing physical events such as objects breaking, rolling and sliding in synchronization with visual cues aids immersion in virtual environments. There is also existing work on interactive musical simulation for percussive instruments [Ren et al. 2012]. In all cases the aim is to produce sounds that are believable and closely tied to an interactive experience. Our work is similarly motivated.

Significant research exists on the complementary problem of modeling sound propagation in virtual scenes, using ray-based [Funkhouser et al. 1998; Chandak et al. 2008; Schissler et al. 2014] and precomputed wave-based techniques [James et al. 2006; Mehra et al. 2013; Raghuvanshi and Snyder 2014]. Propagation modeling improves the sense of presence in virtual 3D scenes by capturing effects such as occlusion and reverberation due to scene geometry. Our simulations generate sounds along with the near-field radiation pattern of the instruments, although in 2D. Techniques similar to [James et al. 2006] could be employed along with existing propagation systems to embed our simulations in an interactive virtual environment, for instance, to provide the wind instrument component of a virtual orchestra [Huopaniemi et al. 1994].

3 Background

Sound resonance and radiation in wind instruments is well-approximated by a set of coupled linear wave equations –

$$\frac{\partial p}{\partial t} = -\rho c^2 \nabla \cdot \mathbf{v}, \tag{1a}$$

$$\frac{\partial \mathbf{v}}{\partial t} = -\frac{1}{\rho}\nabla p. \tag{1b}$$

Table 1 lists the symbols and values employed. These equations predict the spatio-temporal evolution of deviations in pressure p and particle velocity $\mathbf{v}$ from quiescent conditions. Boundary conditions are prescribed as time-dependent normal velocity $v_n\,(\mathbf{x}, t) = \mathbf{v} \cdot \hat{n}$ on the surface of the instrument. Acoustic wave fields are complex and can depend sensitively on instrument shape. Referring to Figure 1, opening a tone hole results in high frequencies being strongly radiated while low frequencies are largely scattered back into the instrument. This changes the frequency-dependent impedance presented by the resonator to the excitation mechanism, changing the note's pitch. Dynamic changes to the tone holes result in a complex interplay of these effects. The resulting transients during note onset are critical for each wind instrument's distinct audible timbre [Rossing et al. 2001].

We simulate the above equations in 2D, which qualitatively model all linear wave effects such as interference, radiation, scattering and propagation delay. However, quantitatively accurate prediction for real instruments requires 3D simulation. For example, although one would expect the scattering and radiation amplitudes due to a tone hole or bell to follow the same general trend with increasing frequency in 2D and 3D, the exact quantitative values at each frequency can be different. Nevertheless, we observe in our results that 2D simulations have musical timbre identifiably similar to real instruments. Note that this limitation is mainly computational, we expect our technique to extend unmodified to 3D.

3.1 Numerical simulation

The Finite-Difference Time-Domain (FDTD) technique performs a uniform discretization of space into cells. The pressure field, p, is sampled at the cell centers, while velocity components v_x and v_y are sampled on a staggered grid, lying on vertical and horizontal cell edges respectively. Staggering the grid affords second-order accurate spatial derivatives. In the case of wind instruments, the cell size, Δ_s, is limited by geometric resolution rather than smallest simulated wavelength. We use $\Delta_s = 3.83$mm which allows real-time computation while still being able to represent the main geometric features required to design virtual instruments. The time-step, Δ_t is restricted by the Courant–Friedrichs–Lewy condition in two dimensions as $\Delta_t \leq \Delta_s/\sqrt{2}c$. We observe stable simulations at the upper bound for $\Delta_t = 7.81 \times 10^{-6}$, corresponding to an update rate of 128,000Hz, which is used in all our simulations.

3.2 Perfectly matched layer

At the edges of the rectangular simulation domain, outgoing radiation from the instrument needs to be absorbed. Insufficient absorp-

tion leads to spurious domain-wide resonance. We employ the Perfectly Matched Layer (PML) technique to ensure high-efficiency attenuation. PML is obtained by analytic continuation of the wave equation into complex, stretched spatial coordinates where propagating waves decay exponentially with distance. Absorptivity is controlled using two stretching fields $\{\sigma_x(\mathbf{x}), \sigma_y(\mathbf{x})\}$ which control the local rate of absorption for propagation along both axial directions. In the continuous case, PML is perfectly reflection-free. Discretization results in reflections that increase with larger jumps in absorptivity. This necessitates a layer of cells across which absorptivity is increased gradually.

PML is usually formulated with σ_x and σ_y controlled separately for maximum generality [Gedney 1996], allowing anisotropic absorption which can be tuned to any application. This requires auxiliary differential equations and variables. We assume a single absorption field by constraining $\sigma_x = \sigma_y$. This allows us to express PML as a particularly simple modification to Eq. 1 –

$$\frac{\partial p}{\partial t} + \sigma p = -\rho c^2 \nabla \cdot \mathbf{v}, \tag{2a}$$

$$\frac{\partial \mathbf{v}}{\partial t} + \sigma \mathbf{v} = -\frac{1}{\rho}\nabla p. \tag{2b}$$

This form is very efficient to compute, involving no auxiliary variables while still remaining effective at absorbing outgoing radiation in our application. A 6-cell thick layer is sufficient to remove audible reflections, with σ decreasing linearly from $0.5/\Delta_t$ at the domain edges to 0 inside.

4 Simulating the resonator

4.1 Dynamic geometry using time-varying PML

Performing virtual instruments requires on-the-fly modifications to geometry, such as when opening a tone hole, closing a valve or applying a mute. It is quite challenging to ensure that such introduction and removal of geometry within a running FDTD simulation remains stable and produces natural-sounding note transitions without pops or clicks. We propose a novel time-varying PML formulation for this purpose. We define a time-varying dynamic geometry field $\beta(\mathbf{x}, t)$ bounded between 0 and 1 globally, allowed to vary slowly in time (compared to Δ_t) at each point in space. A value of $\beta = 1$ indicates air and $\beta = 0$ corresponds to enforcement of some prescribed particle velocity $\mathbf{v_b}(t)$. As discussed later, this velocity could be provided either by an external source, such as an excitation mechanism, or by lossy wall reflections.

We consider the following modification to PML (Eq. 2) –

$$\frac{\partial p}{\partial t} + \sigma' p = -\rho c^2 \nabla \cdot \mathbf{v}, \tag{3a}$$

$$\beta \frac{\partial \mathbf{v}}{\partial t} + \sigma' \mathbf{v} = -\beta^2 \frac{\nabla p}{\rho} + \sigma' \mathbf{v}_b. \tag{3b}$$

Here we have defined the effective absorptivity $\sigma' = 1 - \beta + \sigma$, where σ is the PML absorptivity used at domain boundaries as defined in Section 3.2.

We do not perform geometry editing inside the absorber layer, where we set $\beta = 1$, $\mathbf{v}_b = 0$, reducing these equations to an absorbing PML layer per Eq. 2, as desired. Outside the absorbing layer, $\sigma = 0$ (no PML absorption), thus reducing to $\sigma' = 1 - \beta$. With this substitution, Eq. 3b corresponds to interpolating between two equations using weight β: the momentum equation (1b) and boundary condition enforcement, $\mathbf{v} = \mathbf{v_b}$. As β decreases from 1 to 0 for a fixed differential volume, the contained fluid gradually becomes unresponsive to pressure gradients, instead moving with the specified velocity, $\mathbf{v}_b$. Due to such enforcements of $\mathbf{v}_b$, there can be large net inflow or outflow in the volume, as given by the divergence term in the right hand side of Eq. 3a, causing large changes in pressure. The $\sigma' p$ term on the left hand side dampens such fluctuations in pressure, corresponding to adding or removing mass to the volume. Finally, the momentum weight β^2 in Eq. 3b can be any function that varies monotonically with β between 0 and 1. We choose this specific form because it generates natural-sounding transients in our experiments.

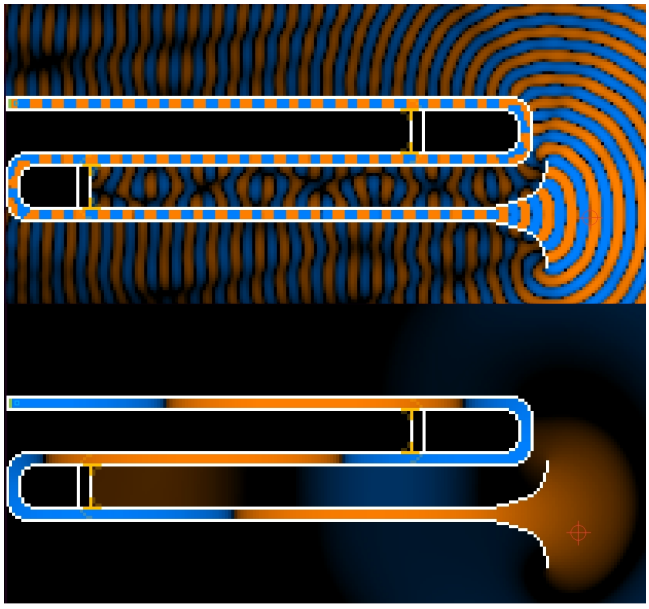


Figure 2: Without wall loss modeling (top), parasitic high-frequency resonance develops. Wall losses are necessary to suppress them (bottom).

These equations lead to the following discrete update rules, where we denote with $\tilde{\nabla}$ the standard discrete spatial derivatives as performed in FDTD.

$$p^{(n+1)} = \frac{p^{(n)} - \rho c^2 \Delta_t \tilde{\nabla} \cdot \mathbf{v}^{(n)}}{1 + \sigma' \Delta_t}, \tag{4a}$$

$$\mathbf{v}^{(n+1)} = \frac{\beta \mathbf{v}^{(n)} - \beta^2 \Delta_t \tilde{\nabla} p^{(n+1)}/\rho + \sigma' \Delta_t \mathbf{v}_b^{(n+1)}}{\beta + \sigma' \Delta_t}. \tag{4b}$$

Note that a semi-implicit scheme has been used for the $\sigma' p$ and $\sigma' \mathbf{v}$ terms, required for stability with this formulation. This equation provides an elegant framework for all the physical phenomena we model, through the unifying dynamic geometry field, β. Time-domain wave simulations are generally quite susceptible to numerical instability. We have observed in our experiments that the resulting numerical implementation with our scheme is robustly stable in the face of arbitrary shape manipulation within a live simulation.

4.2 Visco-thermal wall losses

Multi-dimensional wave simulations naturally support arbitrary propagation directions in the bore. This is essential to their ability to automatically model the effect of geometric features such as tone holes and flaring bells. Propagation paths transverse to the instrument's axis correspond to high frequency resonant modes. We observe that such high-frequency transverse oscillations can build up in 2D virtual instrument models instead of the desired lengthwise resonant modes. This is perhaps because of the much shorter feedback delay with the active excitation mechanism. As we discuss shortly, physical wall losses in wind instruments increase with

frequency. We observe that accounting for wall losses successfully discourages such high-frequency parasitic resonant modes, as shown in Figure 2. Wall loss modeling thus seems essential for multi-dimensional wind instrument simulation. Prior 1D models avoid this difficulty due to their inability to model any propagation directions within the bore other than the instrument's axis.

4.2.1 Lossy boundary condition

The adiabatic, inviscid flow conditions assumed by the linear wave equation are violated near instrument walls. A boundary layer forms between the laminar flow in the interior on one side and no-slip (zero-velocity), isothermal conditions at the boundary. Fortunately, high-resolution Navier-Stokes simulation of the boundary layer can be avoided. The net losses in the laminar flow due to the boundary layer can be approximated analytically. These analytic formulae can then be expressed as a boundary condition relating pressure and normal velocity.

Such a boundary condition suitable for 3D frequency-domain FEM simulation has been proposed by [Bossart et al. 2003]. The main idea is to compute virtual wall velocities so that they capture the analytically determined loss and phase relations. In frequency domain, assuming implicit time dependence of the form $e^{j\omega t}$, this is expressed as an admittance boundary condition: $\widehat{v_n} = \widehat{Y}\hat{p}$ where hat denotes Fourier transform of the corresponding time-domain signal. The wall loss filter $\widehat{Y}$ is given by,

$$\begin{aligned} L(\theta) &= -\frac{l_v \sin^2\theta + l_t(\gamma-1)}{\rho c^2\sqrt{2}}, \\ \widehat{Y} &= (1+j)\,L(\theta)\sqrt{\omega}, \end{aligned} \tag{5}$$

where $l_v = \sqrt{\mu/\rho}$ and $l_t = \sqrt{\mu/\rho\mathcal{P}}$ are constants that determine thickness of the viscous and thermal boundary layers via $l_v/\sqrt{\omega}$ and $l_t/\sqrt{\omega}$ respectively. The angle of incidence is θ, computable from particle velocity as: $\sin^2\theta = 1 - (\mathbf{v}\cdot\widehat{n}/|\mathbf{v}|)^2$ where $\widehat{n}$ is the unit normal pointing into the domain. This term is absent in 1D models that only support propagation paths along the instrument axis [Abel et al. 2003]. Table 1 lists physical constants used above. Because $\mathbf{v}$ is dependent on the solution for p, the solution in frequency-domain is non-trivial, requiring iterative solution starting from a guess for θ [Bossart et al. 2003]. However, in time domain it can be computed straightforwardly, as the time-varying particle velocity is readily available.

4.2.2 Efficient time-domain wall losses

To incorporate Eq. (5) in our time domain simulations, we need a fast and reasonably accurate time-domain digital filter Y' such that its Fourier transform approximates the desired filter ($\widehat{Y'} \approx \widehat{Y}$). Wall losses can then be applied in time-domain as $v_n = Y' * p$, where $*$ denotes convolution. IIR (Infinite Impulse Response) filters allow one to realize the convolution using a recursive update rule. The filter's $\sqrt{\omega}$-dependence in Eq. 5 corresponds to a fractional-derivative in time, making time-domain implementation non-trivial. [Bilbao and Chick 2013] propose using 20^{th} order IIR filters for offline 1D FDTD simulations. While this technique is accurate, it is unfortunately not practical for real-time computation. We find a second-order IIR filter by brute-force sampling optimization of the three feed-forward and two feedback coefficients of Y'. It can provide reasonable accuracy while costing limited additional memory and negligible additional computation.

Rather than directly comparing the trial filter's Fourier transform $\widehat{Y'}$ to the objective filter $\widehat{Y}$, we use an optimization error metric that compares their resulting propagation characteristics. We analytically compute both filters' resulting phase and amplitude for plane-wave propagation along ($\theta = \pi/2$) an infinite cylindrical duct over unit distance (1m). For a reasonable upper bound on error, we assume a duct radius of 5mm. The optimization objective is then the mean of absolute relative errors in the resulting phase and amplitude for $\widehat{Y'}$ and $\widehat{Y}$, computed over the frequency range of $60 - 8000$Hz. The resulting feed-forward and feedback coefficients for the optimized filter Y' are $L(\theta) \times \{495.9, -857.0, 362.8\}$ and $\{-1.35, 0.40\}$ respectively. Note the implicit time-dependence in the filter because of the $L(\theta)$ term which in turn depends on the time-domain particle velocity $\mathbf{v}$. The relative amplitude error stays below 20% and phase errors stay below 0.5% from $350 - 10,000$Hz, providing a reasonable approximation.

4.3 High-amplitude non-linearity

Brass instruments such as the trumpet or trombone can have very high bore pressures, exceeding $10kPa$ on high notes. At such levels, propagation in air is no longer perfectly linear. To first order, high pressure amplitude changes the local speed of sound with peaks propagating faster than troughs. This results in waveform steepening and spectral enrichment at higher frequencies, resulting in the characteristic timbre of brass instruments [Myers et al. 2012]. Digital waveguide simulation for such amplitude-dependent propagation has been proposed in [Cooper and Abel 2010]. The effect is quite straightforward to include in our simulations. The current acoustic pressure deviation, p, is used at each cell to compute the local speed of sound, $c_n = c(1 + \beta_c p/P_0)$, and used in place of c in Eq. 3a. The atmospheric pressure is $P_0 = 101325\,Pa$ and $\beta_c = 1.2$ for air. Stability is ensured by clamping the computed speed c_n to $1.1c$ and satisfying the more restrictive CFL condition $\Delta_t < \Delta_s/1.1\sqrt{2}c$. Although this scheme reduces the accuracy of the simulation to first-order in areas of high pressure variation, the resulting improvement in sound quality is significant.

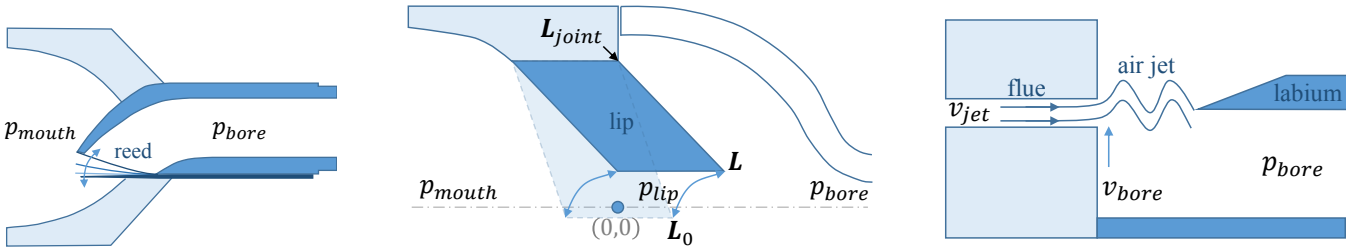
5 Excitation mechanisms

The non-linear physics of wind instrument excitation mechanisms is quite complex [Fletcher and Rossing 1998]. We have integrated a few of the well-known simplified models for single-reed (clarinet, saxophone), buzzing lips (trumpet, tuba) and air jet (recorder, flute). This selection is certainly not exhaustive. Our motivation is to consider non-linear excitation mechanisms covering a wide range of physical phenomena and demonstrate that although they are usually integrated with 1D virtual instrument models, they can be successfully adapted to 2D resonator simulations. We start by discussing some general issues faced while coupling these mechanisms to our simulations and then discuss each in detail along with our modifications. Physical parameter values used are listed in Table 2.

5.1 Coupling with the resonator

Dynamical equations modeling the feedback between excitation models and resonator can result in a simultaneous system of equations relating physical state of the excitation model to the wave field in the resonator. Such global solutions preclude real-time computation. Adding a time-step of delay allows fast explicit schemes but can result in unstable simulations, especially for large time steps. In our case, the spatial resolution required by 2D modeling of resonator geometry enforces a time step of $\Delta_t \approx 8\mu s$. This has the fortunate consequence that adding a single-step delay to the discrete formulation incurs negligible phase error. We observe robustly stable simulation in all our tests when adding such a delay, allowing efficient and modular integration of dynamical excitation models.

Excitation models generate the volume flow entering into the instru-



(a) single reed **(b)** lips **(c)** air jet

Figure 3: Excitation mechanisms.

ment from the mouth of the player, U_{bore}. Assuming an instrument-dependent height in the third dimension, H, we divide the flow uniformly between all excitation cells. Their particle velocity is set as $|\mathbf{v_b}| = U_{bore}/H\Delta_s n$, where n is the number of user-drawn excitation cells. The direction of excitation velocities is user-controlled and shared across all excitation cells to generate locally-plane waves. The input to the excitation model such as bore pressure or velocity is taken from a single cell specified by the user.

The output volume flow, U_{bore}, can have unwanted frequency content above 22kHz. Numerical propagation errors in this range can couple with non-linear excitation in some cases, resulting in noise in the output sound. We low-pass the output of all excitation mechanisms above 22kHz using a second-order Butterworth filter which has nearly linear phase response in the audible range. The feed-forward and feedback coefficients for the filter are $0.160863 \times \{1, 2, 1\}$ and $\{-0.590436,\ 0.233886\}$ respectively.

5.2 Single reed

Many woodwind instruments such as the clarinet employ a single reed, as shown in Figure 3a. The first vibrational mode frequency for typical clarinet reeds is much higher than the playing range. This allows reasonable approximation with a quasi-static model in which reed inertia is ignored [Scavone and Smith 1996], commonly employed in virtual instrument models [Smith 2010]. The reed's elasticity is characterized by a single spring constant per unit area, k_r, measured experimentally [Dalmont et al. 2003]. The pressure difference in the player's mouth and instrument bore, $\Delta p = p_{mouth} - p_{bore}$ displaces the reed to $x_r = \Delta p/k_r$, controlling flow into the instrument. The reed motion is constrained as $x_r \in (0, h_r)$ corresponding to its equilibrium open position and maximum displacement that closes the air channel respectively. Air flow from the player's mouth is in the form of a thin jet through the small reed opening that transfers its kinetic energy into the instrument bore over a distance much shorter than acoustic wavelengths. Thus incompressible flow is assumed and steady-state Bernoulli's equation yields the particle velocity $\sqrt{2\,|\Delta p|\,/p}$. After some simplification, this yields the volume flow into the bore,

$$U_{bore} = w_j h_r \left(1 - \frac{\Delta p}{\Delta p_{max}}\right) \sqrt{\frac{2\Delta p}{\rho}}$$

when $\Delta p > 0$, $U_{bore} = 0$ otherwise. Here $\Delta p_{max} \equiv k_r h_r$ and w_j is the effective width of the jet, its cross-section area being $w_j h_r$.

Our modifications: When the reed is nearly shut or open in this model, we observe high-frequency oscillations audible as a "grainy" sound. In the near-shut case, this results from the inability of the simplified model to capture the complex aero-acoustics in a thin channel [Fletcher and Rossing 1998]. We enforce smoothness during closure by multiplying U_{bore} with a tight, decreasing shelf function: $0.5 + 0.5\,\tanh\left(4\left(-1 + \left(\Delta p_{max} - \Delta p\right)/w\Delta p_{max}\right)\right)$, with $w = 0.01$. For the open position, spurious oscillations can result because U_{bore} varies sensitively to changes in Δp when $\Delta p \approx 0$. Substituting $\Delta p \leftarrow \max\{\alpha_r \Delta p_{max}, 0\}$ in the above expression with $\alpha_r = 0.05$ avoids such behavior while leaving the main operating range near $\Delta p = \Delta p_{max}/3$ unaffected.

5.3 Buzzing lips

Instruments in the brass family such as trumpets and tubas are performed by buzzing the player's lips into a mouthpiece. Lip stiffness and mouth pressure largely determine produced pitch. Compared to a stiff reed fixed on one end, lips exhibit more degrees of freedom. Refer [Campbell 2004] for a review of brass excitation models. We employ the well-regarded two-dimensional model proposed in [Adachi and Sato 1996] that produces elliptical lip motion as observed in measurements [Copley and Strong 1996].

Referring to Figure 3b, the origin is located as shown. Air flows from the player's mouth at pressure p_{mouth}, to between the lips where pressure changes to p_{lip}, exiting into the bore at pressure p_{bore}. Lip motion is assumed symmetric about the horizontal (X) axis. The primary dynamical variable is the position of the lip, $\mathbf{L} = (l_x, l_y)$, obeying the following equation of motion,

$$m_l \frac{\partial^2 \mathbf{L}}{\partial t^2} + \frac{\sqrt{m_l k_l}}{Q} \frac{\partial \mathbf{L}}{\partial t} + k_l\left(\mathbf{L} - \mathbf{L_0}\right) = 2\mathbf{F}_{\Delta p} + 2\mathbf{F}_{Bern} + \mathbf{F}_{coll}. \tag{6}$$

The left hand side describes a damped simple harmonic oscillator with rest position $\mathbf{L_0}$, and damping expressed via the quality factor, Q. The spring constant k_l and mass m_l are derived from the user-controlled lip resonance frequency f_l as, $m_l = 1.5/\left((2\pi)^2 f_{lip}\right)$ and $k_l = 1.5 f_l$. Denoting width of lip opening w_c and lip thickness l_c, the normal force due to pressure difference across the lips is $\mathbf{F}_{\Delta p} = w_c\left(p_{mouth} - p_{bore}\right)\hat{z} \times \left(\mathbf{L} - \mathbf{L}_{joint}\right)$. The pressure between the lips, p_{lip} results in a transverse Bernoulli force $\mathbf{F}_{Bern} = w_c\, l_c\, p_{lip}\hat{y}$. Finally, the lip collision force is given by $\mathbf{F}_{coll} = \max\left\{-3kl_y, 0\right\}\hat{y}$ with quality factor reduced to $Q = 0.5$ on collision ($l_y < 0$). Lip motion within the cup generates a volume flow of $U_{lip} = \hat{z} \cdot \left(w_c\left(\mathbf{L} - \mathbf{L}_{joint}\right) \times d\mathbf{L}/dt\right)$. Air passing between the lips accounts for most of the volume flow, $U_{ac} = S\, v_e$, where $S = \max\left\{2w_c l_y, 0\right\}$ is the lip opening area and v_e is particle velocity at the lip exit. The latter obeys the following relations to pressure –

$$\rho l_c \frac{\partial v_e}{\partial t} + \frac{\rho}{2} v_e\,|v_e| + p_{lip} - p_{mouth} = 0 \tag{7}$$

$$-\rho v_e\,|v_e|\,\frac{S}{S_{cup}}\left(1 - \frac{S}{S_{cup}}\right) + p_{bore} - p_{lip} = 0 \tag{8}$$

Equation 7 governs flow from the mouth to the end of the lip channel assuming laminar flow and conservation of energy. Equation 8 is derived by assuming perfect boundary separation of the air jet from lips into the mouthpiece cup, conserving momentum, where S_{cup} is the area of bore entrance from cup.

	Meaning	Value
w_j	effective jet width	1.2×10^{-2}
h_r	max. reed displacement	6.0×10^{-4}
k_r	reed stiffness	8.0×10^{6}

(a) Single-reed [Dalmont et al. 2003]

	Meaning	Value
f_l	lip resonance frequency	$60 - 700Hz$
m_l	lip mass	$1.5/\left((2\pi)^2 f_{lip}\right)$
k_l	lip stiffness	$1.5 f_{lip}$
w_c	lip opening width	7.0×10^{-3}
l_c	lip thickness	2.0×10^{-3}
S_{cup}	mouthpiece entrance area	2.3×10^{-4}
$\mathbf{L_0}$	lip resting position	1.0×10^{-3}, -1.0×10^{-4}

(b) Buzzing lips [Adachi and Sato 1996]

	Meaning	Value
h_f	width of flue	1.0×10^{-3}
d_l	flue exit to labium distance	1.0×10^{-2}
y_l	labium vertical offset	3×10^{-4}
α_j	jet speed coefficient	0.4

(c) Air jet [de La Cuadra 2006]

Table 2: Physical parameters for excitation mechanisms. All values are in SI units.

Our modifications: To solve the simultaneous system of equations, we introduce a single time-step delay by employing Eq. 8 and use the existing values of v_e and p_{bore} to determine p_{lip}. This is used to update the lip's position $\mathbf{L}$ per Eq. 6. The particle velocity v_e is then updated via Eq. 7 to calculate U_{ac} and U_{lip} as described previously, yielding the net output flow, $U_{bore} = U_{lip} + U_{ac}$.

Terms corresponding to particle velocity v_e appear as U_{ac}/S in the original paper. We utilize particle velocity directly, which is better behaved numerically when $S \approx 0$. Further, we relax the assumption of flow always going from the player's mouth into the instrument ($v_e > 0$) by substituting $\rho v_e^2 \rightarrow \rho v_e |v_e|$, making sure that the Bernoulli effect respects the direction of flow. We observed in our experiments that this change improves robustness by ensuring that momentary negative flow does not cause instability.

The original paper suggests using a value of $Q = 3$ for the lip damping factor. While this produces convincing sustained tones, we observe that better sounding transients are obtained in our system by reducing the damping, setting $Q = 8$. With this lowered damping, sustained lip oscillations are obtained even if the resonator is removed, corresponding to reality. This ensures that while transitioning between musical notes during performance, the lips keep oscillating and providing energy to the resonator.

5.4 Air jet

Flute and recorder excitation relies on the complex aero-acoustics of an unstable air jet that is directed at a sharp labium (Figure 3c). We use the jet drive model proposed in [Verge 1995], improved in later work [de La Cuadra 2006] with better estimates of some physical parameters. The vertical component of particle velocity near the flue exit, v_{bore} results in deflection of the jet as

$$\eta(t) = \frac{v_{bore}}{v_j} h_f, \tag{9}$$

where h_f is the flue channel width. This captures the resonant cavity's influence on the jet. The particle velocity of the jet at the flue exit is $v_j = \sqrt{2p_{mouth}/\rho}$ where p_{mouth} is pressure in the player mouth. The deflection is advected along the jet, reaching the labium edge after time delay $\tau = d_l/\alpha_j v_j$, where d_l is the distance to the labium edge from the flue exit and $\alpha_j v_j$ is the effective jet propagation speed. Depending on the deflection, and assuming analytic expressions for the jet velocity profile across its cross-section characterized by a width parameter b_j, the jet contributes the following flow into the bore,

$$U_{bore} = -\frac{b_j v_j H}{\Delta_s} \tanh\left(\frac{\eta(t-\tau) - y_l}{b_j}\right), \tag{10}$$

where y_l is the vertical offset of the labium edge with respect to the flue exit center, and H is instrument height. The propagation delay in the feedback, τ, is typically around $1ms$ and is essential to self-sustained oscillations in flue instruments.

Our modifications: The air jet model depends sensitively on the vertical particle velocity in the bore, v_{bore}. It is read from the user-specified cell for excitation input as mentioned in Section 5.1. This cell has to be selected carefully to be on or close to the embouchure hole, for the instrument to function. Turbulent noise is not accounted for by these models but can be quite audible in the jet-driven family of instruments. We add a small white noise term with amplitude 0.5m/s to v_{bore} in Eq. 9. The noise is naturally filtered through the instrument's body, causing it to meld with the musical note being played.

6 GPU Solver

In this section we provide details of the system and GPU implementation.

6.1 Implementation

Finite difference methods in 2D represent fields as two dimensional arrays of values, with update operations requiring access to spatial neighbors for each cell. Graphics processors are designed for fast, parallel two-dimensional texture accesses, thus providing an excellent fit for this data organization. The GPU architecture is tuned for high throughput at visual update rates. Audio simulations operate at orders of magnitude higher rates, making synchronization costs a major hurdle. This limits some implementations to using only a single Streaming Multiprocessor (SM) [Hsu and Pérez 2013], a fraction of the computational power available on today's graphics cards that have more than 10 SMs. We utilize programmable shaders written in the GLSL language and observe real-time performance at update rates of $128,000$Hz by utilizing all available SMs. Shaders have the additional advantage of allowing direct rendering of visualizations from simulation data resident in GPU memory. Although some of the details of our technique are specific to GPUs, we expect it would be easy to adapt to any parallel architecture.

6.2 Data organization

The entire simulation data is stored in a four-channel (RGBA), 32-bit floating point global texture. It is bound to the frame buffer only once, avoiding costly bind operations at every time-step. The texture is divided into four quadrants that store the field for the next, current, and two prior time-steps. The global texture contains additional space for maintaining the state of the excitation mechanism and relevant data as applicable, such as the delay line for the air jet excitation model in Section 5.4. Finally, a single row in the

global texture contains synthesized audio samples. FDTD typically requires data only for one prior time-step. However, second-order IIR filtering involved in modeling visco-thermal wall losses (Section 4.2) necessitates storing fields for two prior steps.

Computation proceeds for each time-step by rendering to the "next" quadrant while reading from the other three, followed by a memory barrier for synchronization and updating pointers circularly so that "next" becomes "current" and so on. Note that this scheme avoids read and write to the same pixel, which can cause performance penalties on GPUs. Figure 4 shows a schematic of simulation data organization in the texture. Rather than storing pressure and velocity fields as separate textures, we pack the data into the texture's four color channels. This greatly improves data locality and cache performance. For each cell, we store the pressure at the cell center p with the velocity v_x to it right and v_y above it on the staggered velocity grid. These values take three of the four texture channels per cell. The fourth channel is utilized for affecting the dynamics of the cell's field, discussed next.

6.3 Boundary handling

The fourth available channel in each pixel is used to store the cell's dynamical state which may be modified by user interaction. A cell can be in one of four states: excitation, wall, air and dynamic. The behavior of each of these kinds of cells is completely characterized by the dynamic geometry field, $\beta(\mathbf{x}, t)$, used in Eq. 4. This field is sampled at cell centers. Excitation cells have $\beta = 0$ with the particle velocity $\mathbf{v}_b$ provided by the currently chosen excitation mechanism, as discussed in Section 5.1. Wall cells use the same value of $\beta = 0$ but with $\mathbf{v}_b$ provided by the lossy boundary condition as described in Section 4.2.2. Air cells are specified by $\beta = 1$. When a user changes a cell between air and wall, the value of β is changed within a single time-step. This generates an impulsive sound providing a useful interaction cue for geometry creation.

For geometry that needs to change smoothly over time, such as tone holes, the user draws the cells as dynamic. Dynamic cells can be grouped so they utilize a single value of β that varies smoothly over time between 0 and 1. We set this transition interval to 10ms over which β changes linearly. Each dynamic cell thus stores the ID of its group as its current state. Based on the ID, the value of β is looked up for the current cell at each time-step. Grouping allows easy modeling of dynamic geometric elements that change together, such as changing multiple valve states, or a mute being introduced. Moving geometry can also be approximated. One can place cells with different group IDs in a row and stagger their transitions, such as the slide whistle example discussed in Section 7.

6.4 Simulation update

At each time-step, pressure at each cell center is updated via Eq. 4a using the velocities stored with the current cell and its neighbors. The velocity update per Eq. 4b poses a problem, requiring *updated* pressures for the neighbors above and to the right. The standard approach is to update the global pressure field and then use it to update the global velocity field. This introduces a costly synchronization barrier between pressure and velocity updates. Further, one performs a scan on the entire field data twice, increasing memory bandwidth requirements. Our scheme avoids these issues and is much faster in practice: the current cell computes updated pressures for the required neighbors as temporary values, which are used to update its particle velocity. The required data for a cell is shown in dark colors in Figure 4, with the containing pixels marked by dotted lines. The resulting access pattern is still quite local to the cell.

An additional issue is that the value of the dynamic geometry field, β, is defined at cell centers and velocity updates require its value on cell edges. We tie-break between the cells adjoining the edge by taking the minimum value of β for the two cells sharing the edge. This ensures that boundary conditions are properly enforced at the edges of a wall or excitation cell. Updated pressure from the user-specified listener cell is appended to the audio buffer. To reduce read-back bandwidth requirements, only every other audio sample is stored in the audio buffer, which has a size of 2048 samples. When the buffer is full, the data is read back from the GPU. The resulting audio at a sampling rate of 64kHz is sent directly to the audio device for playback.

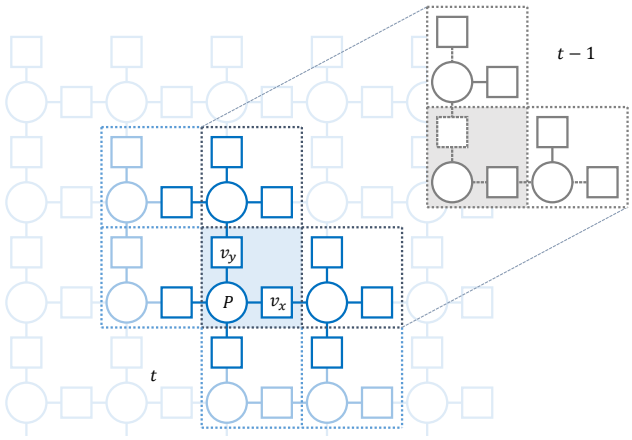


Figure 4: Data organization. Pressure and velocity (p, v_x, v_y) are packed into three color channels of a 32-bit floating point RGBA texture. Pixels are shown as dotted squares. History for two prior time-steps is stored similarly, one of which is shown in gray. The fourth channel (not shown) is used to store interactive information, such as parameters for dynamic geometry.

7 Results

Our system runs in real-time on an NVIDIA GeForce GTX Titan Black graphics card at an update rate of 128,000Hz. Please refer to the accompanying video which demonstrates all the effects discussed is this section. The simulation cell size was 3.83mm in a rectangular domain of dimensions 0.84m×.42m (220 × 110 pixels). The cell size and domain dimensions allow real-time execution on a region large enough to contain typical wind instruments. When designing instruments with curved geometry, the stair-cased approximation results in numerical scattering error. However, it is limited because all audible wavelengths are significantly larger than the cell size. The pitch of the instrument is nevertheless affected and in our informal tests, reasonable convergence is achieved in the range of 1mm cell size. With advancing GPU computational power, one can expect such simulations to be real-time in the near future.

7.1 Virtual Instruments

We have designed simplified 2D analogues of real instruments with our system, as shown in Figure 1.

A **Simple Trumpet** tuned to a chromatic scale is shown in Figure 1(a). Our system lets the user create arbitrary geometric arrangements, such as the valve system in this case. A key was associated to control the valves in a coordinated way to change the effective sound path. There is little work on modeling such dynamic brass valve systems [Harrison and Chick 2014]. The flaring bell makes the sound more mellow, qualitatively similar to reality. High amplitude non-linearity (Section 4.3) does yield the expected bright, rich sound of brass instruments in our simulations. With-

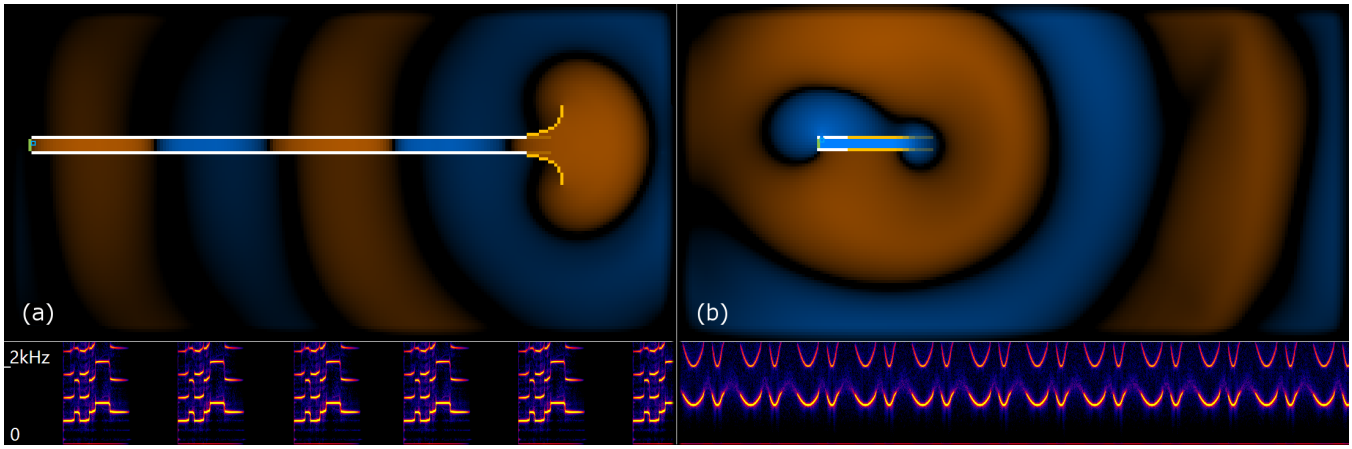


Figure 5: (a) A bugle call on a trumpet-like instrument produces natural-sounding transients between notes, seen in the spectrogram at the bottom. (b) A model akin to a slide whistle, with dynamically-changing length, produces artifact-free audio with continuously-varying pitch.

out such modeling the instruments lack this characteristic timbre. We also clearly observe elliptical lip motion in our simulations, as expected from the brass excitation model (Section 5.3).

A **2D clarinet** tuned to a pentatonic scale is depicted in Figure 1(b). When the register key is opened (bottom), the pitch shifts an octave higher and a large fraction of the sound is observed to radiate from the key. As shown in the accompanying demo, the model clearly exhibits odd harmonics. These behaviors are qualitatively as expected [Fletcher and Rossing 1998].

A simple **diatonic flute** is shown in Figure 1(c). The small embouchure hole on the left top of the instrument is sufficient to turn it into a resonator open at both ends, clearly exhibiting a dipole radiation pattern and even harmonics. These are heard as distinctly flute-like sounds. We also interfaced a digital wind controller to provide breath pressure and tone hole state, resulting in a significantly more nuanced performance.

7.2 Dynamic geometry

Our technique produces smooth, natural-sounding transients between musical notes, enabled by our time-varying PML formulation. An example is shown in Figure 5(a) of a bugle call, with the continuously varying pitch clearly visible in the spectrogram. We use a value for $10ms$ as the transition time for all dynamic geometry. As a stress test, in Figure 5(b) we show a model akin to a slide whistle. The length varies smoothly causing continuous change in the pitch as seen in the spectrogram at the bottom.

7.3 Experimental designs

Our system enables intuitive experimentation with virtual instruments by direct shape design. Figure 6(a) shows a brass instrument that supports multiple simultaneous paths which would be difficult to create in reality. Very complex valve systems can be created, as shown in Figure 6(b), to make a maze-like instrument which routes sounds through a large number of possible paths with lengths tuned to musical notes. Figure 6(c) depicts a hybrid instrument containing both a valve and a tone hole. Please consult the supplemental video to listen to their sounds.

7.4 Comparisons

We have performed comparisons of our simulations with recorded musical tones from a $B\flat$ trumpet and a $B\flat$ clarinet. The players were presented with videos of our virtual instruments and tried to best reproduce the music. For this purpose, they were instructed not to re-articulate notes of the melody or to adjust their embouchure. For sustained (held) note comparisons, we also show results from the Synthesis Tool Kit (STK), a well-regarded implementation of real-time digital waveguides. We tuned the parameters for the STKClarinet object as: {reed stiffness $= 80$, noise level $= 32$, vibrato $= 0$, aftertouch $= 128$} and for the STKBrass as: {amplitude $= 0.5$, lip tension $= 64$, slide length $= 32$, mod frequency $= 8$, mod wheel $= 4$, aftertouch $= 128$}. The tests show that the musical timbre of our simulations is believably similar to real instruments, and our simulations sound less synthetic than the STK.

We also performed comparisons with recordings for simple performances on the clarinet and trumpet models. While for the clarinet the player was able to match notes, the notes do not quite match on the trumpet. Nevertheless, the results do show that the perceived transients between notes and overall instrument timbre in our simulations sound natural, similar to recordings. The brass sounds are substantially brighter in our simulations, perhaps because the listener was placed close-by in front of the bell during simulation, while the live recording was done to the side and further away.

8 Conclusion and future directions

We have presented the first technique for real-time wave simulation of 2D wind instruments for all audible frequencies. Our interactive system promotes experimentation with novel virtual instrument designs. The user can directly manipulate shape to instantly affect synthesized sound, providing a more natural medium for experimenting with virtual instruments than prior 1D techniques requiring digital signal processing expertise. Generated sounds are naturally constrained to be physically plausible. Field and excitation visualizations provide insight into instrument operation. In future work, we wish to undertake more detailed investigation into natural interfaces to these virtual instruments, both for design and performance. Our technique could also form the basis for accurate, interactive 3D simulations in the future. A unified excitation model that subsumes the various separate models for reed, lips and air jet would also be a fruitful direction, to let the user explore a wider variety of virtual instrument timbres.

Acknowledgments

Thanks to Paul Hembree and Kyle Rowan for providing the trumpet and clarinet recordings.

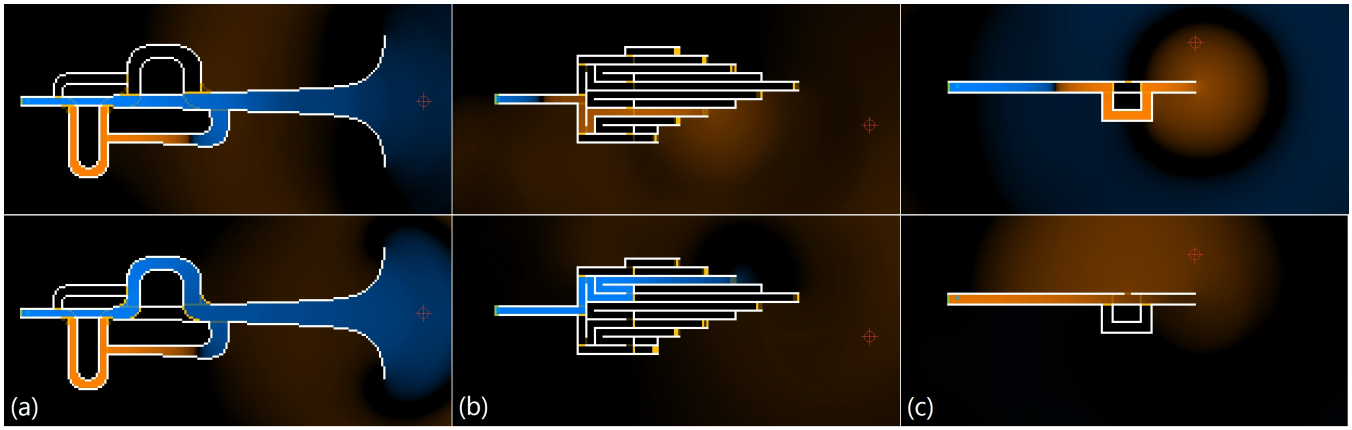


Figure 6: Virtual instrument experiments. (a) A trumpet-like instrument with a valve system supporting multiple simultaneously active sound paths (b) A sound maze controlled by a complex valve system (c) A hybrid instrument containing both a valve and a tone hole.

References

ABEL, J., SMYTH, T., AND SMITH, J. O. 2003. A simple, accurate wall loss filter for acoustic tubes. *DAFX 2003 Proceedings*, 53–57.

ADACHI, S., AND SATO, M. 1996. Trumpet sound simulation using a two-dimensional lip vibration model. *The Journal of the Acoustical Society of America 99*, 2 (Feb.), 1200–1209.

BADER, R. 2013. Damping in turbulent Navier-Stokes finite element model simulations of wind instruments. *The Journal of the Acoustical Society of America 134*, 5 (Nov.), 4219.

BAINES, A. 1967. *Woodwind Instruments and Their History*, third ed. Faber & Faber, London.

BILBAO, S., AND CHICK, J. 2013. Finite difference time domain simulation for the brass instrument bore. *The Journal of the Acoustical Society of America 134*, 5 (Nov.), 3860–3871.

BILBAO, S., HAMILTON, B., TORIN, A., WEBB, C., GRAHAM, P., GRAY, A., KAVOUSSANAKIS, K., AND PERRY, J. 2013. Large scale physical modeling sound synthesis. In *Proc. Stockholm Music Acoustics Conference*. 593–600.

BILBAO, S. 2009. *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics*, 1 ed. Wiley, Dec.

BOSSART, R., JOLY, N., AND BRUNEAU, M. 2003. Hybrid numerical and analytical solutions for acoustic boundary problems in thermo-viscous fluids. *Journal of Sound and Vibration 263*, 1 (May), 69–84.

CAMPBELL, M. 2004. Brass Instruments As We Know Them Today. *Acta Acustica united with Acustica 90*, 600–610.

CHADWICK, J. N., AND JAMES, D. L. 2011. Animating Fire with Sound. *ACM Trans. Graph. 30*, 4 (July).

CHADWICK, J. N., AN, S. S., AND JAMES, D. L. 2009. Harmonic shells: a practical nonlinear sound model for near-rigid thin shells. *ACM Trans. Graph. 28* (Dec.).

CHADWICK, J. N., ZHENG, C., AND JAMES, D. L. 2012. Precomputed Acceleration Noise for Improved Rigid-body Sound. *ACM Trans. Graph. 31*, 4 (July).

CHANDAK, A., LAUTERBACH, C., TAYLOR, M., REN, Z., AND MANOCHA, D. 2008. AD-Frustum: Adaptive Frustum Tracing for Interactive Sound Propagation. *IEEE Transactions on Visualization and Computer Graphics 14*, 6, 1707–1722.

COOK, P. R. 2002. *Real Sound Synthesis for Interactive Applications (Book & CD-ROM)*, 1st ed. AK Peters, Ltd.

COOPER, C. M., AND ABEL, J. S. 2010. Digital simulation of "brassiness" and amplitude-dependent propagation speed in wind instruments. In *Proc. 13th Int. Conf. on Digital Audio Effects (DAFx-10)*, 1–6.

COPLEY, D. C., AND STRONG, W. J. 1996. A stroboscopic study of lip vibrations in a trombone. *The Journal of the Acoustical Society of America 99*, 2 (Feb.), 1219–1226.

DALMONT, J.-P., GILBERT, J., AND OLLIVIER, S. 2003. Nonlinear characteristics of single-reed instruments: Quasistatic volume flow and reed opening measurements. *The Journal of the Acoustical Society of America 114*, 4 (Oct.), 2253–2262.

DE LA CUADRA, P. 2006. *The sound of oscillating air jets: Physics, modeling and simulation in flute-like instruments*. PhD thesis, Stanford University.

FABRE, B., GILBERT, J., HIRSCHBERG, A., AND PELORSON, X. 2012. Aeroacoustics of Musical Instruments. *Annual Review of Fluid Mechanics 44*, 1, 1–25.

FLETCHER, N. H., AND ROSSING, T. 1998. *The Physics of Musical Instruments*, 2nd ed. corr. 5th printing ed. Springer, Dec.

FUNKHOUSER, T., CARLBOM, I., ELKO, G., PINGALI, G., SONDHI, M., AND WEST, J. 1998. A Beam Tracing Approach to Acoustic Modeling for Interactive Virtual Environments. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '98, 21–32.

GEDNEY, S. D. 1996. An anisotropic perfectly matched layer-absorbing medium for the truncation of FDTD lattices. *Antennas and Propagation, IEEE Transactions on 44*, 12 (Dec.), 1630–1639.

GIORDANO, N. 2014. Simulation studies of a recorder in three dimensions. *The Journal of the Acoustical Society of America 135*, 2 (Feb.), 906–916.

HAMILTON, B., AND WEBB, C. J. 2013. Room acoustics modelling using GPU-accelerated finite difference and finite volume methods on a face-centered cubic grid. *Proc. Digital Audio Effects (DAFx), Maynooth, Ireland*.

HARRISON, R., AND CHICK, J. 2014. A Single Valve Brass Instrument Model using Finite-Difference Time-Domain Methods. In *International Symposium on Musical Acoustics*.

HELMHOLTZ, H. 1885. *On the Sensations of Tone*, second ed. Longmans, Berlin.

HSU, B., AND PÉREZ, M. S. 2013. Realtime GPU Audio. *Queue 11*, 4 (Apr.).

HUOPANIEMI, J., KARJALAINEN, M., VAELIMAEKI, V., AND HUOTILAINEN, T. 1994. Virtual instruments in virtual rooms - a real-time binaural room simulation environment for physical models of musical instruments. In *Proceedings of the International Computer Music Conference*, 455.

JAMES, D. L., BARBIC, J., AND PAI, D. K. 2006. Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. *ACM Transactions on Graphics 25*, 3 (July), 987–995.

KEEFE, D. H. 1984. Acoustical wave propagation in cylindrical ducts: Transmission line parameter approximations for isothermal and nonisothermal boundary conditions. *The Journal of the Acoustical Society of America 75*, 1, 58–62.

LANGLOIS, T. R., AN, S. S., JIN, K. K., AND JAMES, D. L. 2014. Eigenmode Compression for Modal Sound Models. *ACM Trans. Graph. 33*, 4 (July).

LEFEBVRE, A., AND SCAVONE, G. P. 2012. Characterization of woodwind instrument toneholes with the finite element method. *The Journal of the Acoustical Society of America 131*, 4 (Apr.), 3153–3163.

MACALUSO, C. A., AND DALMONT, J.-P. 2011. Trumpet with near-perfect harmonicity: Design and acoustic results. *The Journal of the Acoustical Society of America 129*, 1 (Jan.), 404–414.

MCINTYRE, M. E., SCHUMACHER, R. T., AND WOODHOUSE, J. 1983. On the oscillations of musical instruments. *The Journal of the Acoustical Society of America 74*, 5 (Nov.), 1325–1345.

MEHRA, R., RAGHUVANSHI, N., ANTANI, L., CHANDAK, A., CURTIS, S., AND MANOCHA, D. 2013. Wave-based Sound Propagation in Large Open Scenes Using an Equivalent Source Formulation. *ACM Trans. Graph. 32*, 2 (Apr.).

MYERS, A., PYLE, R. W., GILBERT, J., CAMPBELL, D. M., CHICK, J. P., AND LOGIE, S. 2012. Effects of nonlinear sound propagation on the characteristic timbres of brass instruments. *The Journal of the Acoustical Society of America 131*, 1 (Jan.), 678–688.

NORELAND, D., KERGOMARD, J., LALOE, F., VERGEZ, C., AND GUILLEMAIN, P. 2013. The logical clarinet: numerical optimization of the geometry of woodwind instruments. *Acta Acustica united with Acustica 99*, 615–628.

O'BRIEN, J. F., SHEN, C., AND GATCHALIAN, C. M. 2002. Synthesizing sounds from rigid-body simulations. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM, New York, NY, USA, 175–181.

RAGHUVANSHI, N., AND SNYDER, J. 2014. Parametric Wave Field Coding for Precomputed Sound Propagation. *ACM Trans. Graph. 33*, 4 (July).

REN, Z., MEHRA, R., COPOSKY, J., AND LIN, M. C. 2012. Tabletop Ensemble: Touch-enabled Virtual Percussion Instruments. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D '12, 7–14.

RICARDO, A., SCAVONE, G. P., AND VAN WALSTIJN, M. 2007. Numerical simulations of fluid-structure interactions in single-reed mouthpieces. *The Journal of the Acoustical Society of America 122*, 3 (Sept.).

ROSSING, T. D., MOORE, F. R., AND WHEELER, P. A. 2001. *The Science of Sound, 3rd Edition*, 3rd ed. Addison-Wesley, Dec.

SAVIOJA, L., MANOCHA, D., AND LIN, M. 2010. Use of GPUs in room acoustic modeling and auralization. In *Proc. Int. Symp. Room Acoustics*.

SAVIOJA, L. 2010. Real-Time 3D Finite-Difference Time-Domain Simulation of Mid-Frequency Room Acoustics. In *13th International Conference on Digital Audio Effects (DAFx-10)*.

SCAVONE, G. P., AND SMITH, J. O. 1996. Digital waveguide modeling of woodwind toneholes. *The Journal of the Acoustical Society of America 100*, 4 (Oct.), 2812.

SCHISSLER, C., MEHRA, R., AND MANOCHA, D. 2014. High-order Diffraction and Diffuse Reflections for Interactive Sound Propagation in Large Environments. *ACM Trans. Graph. 33*, 4 (July).

SMITH, J. O. 1986. Efficient simulation of the reed-bore and bow-string mechanisms. In *Proc. Int. Computer Music Conf.*, 275–280.

SMITH, J. O. 1996. Physical Modeling Synthesis Update. *Computer Music Journal 20*, 2, 44–56.

SMITH, J. O. 2004. Virtual Acoustic Musical Instruments: Review and Update. *Journal of New Music Research 33*, 3, 283–304.

SMITH, J. O. 2010. *Physical Audio Signal Processing*. http://ccrma.stanford.edu/~jos/pasp/ (online book, accessed Jan 2014).

TSINGOS, N., JIANG, W., AND WILLIAMS, I. 2011. Using Programmable Graphics Hardware for Acoustics and Audio Rendering. *J. Audio Eng. Soc 59*, 9, 628–646.

VÄLIMÄKI, V., PAKARINEN, J., ERKUT, C., AND KARJALAINEN, M. 2006. Discrete-time modelling of musical instruments. *Reports on Progress in Physics 69*, 1 (Jan.), 1–78.

VAN WALSTIJN, M., AND CAMPBELL, M. 2003. Discrete-time modeling of woodwind instrument bores using wave variables. *The Journal of the Acoustical Society of America 113*, 1 (Jan.), 575–585.

VAN WALSTIJN, M. 2007. Wave-based Simulation of Wind Instrument Resonators. *Signal Processing Magazine, IEEE 24*, 2 (Mar.), 21–31.

VERGE, M.-P. 1995. *Aeroacoustics of confined jets: with applications to the physical modeling of recorder-like instruments*. PhD thesis, Technische Universiteit Eindhoven.

WEBB, C. J. 2014. *Parallel computation techniques for virtual acoustics and physical modelling synthesis*. PhD thesis, Acoustics and Audio Group, University of Edinburgh.

ZHENG, C., AND JAMES, D. L. 2009. Harmonic Fluids. In *ACM SIGGRAPH 2009 Papers*, ACM, New York, NY, USA, SIGGRAPH '09, 1–12.