Computational modeling of stiff piano strings using digital waveguides and finite difference

Julien Bensa, Stefan Bilbao, Richard Kronland-Martinet, Julius Smith, Thierry Voinier

Computational Modeling of Stiff Piano Strings Using Digital Waveguides and Finite Differences

Julien Bensa[1], Stefan Bilbao[2], Richard Kronland-Martinet[3], Julius O. Smith III[4], Thierry Voinier[3]

[1]: Laboratoire d'Acoustique Musicale, Université Pierre et Marie Curie, Paris
[2]: Sonic Arts Research Centre, Department of Music, Queen's University, Belfast
[3]: Laboratoire de Mécanique et d'Acoustique, Marseille
[4]: Center for Computer Research in Music and Acoustics (CCRMA), Department of Music, Stanford University

Summary
As is well-known, digital waveguides offer a computationally efficient, and physically motivated means of simulating wave propagation in strings. The method is based on sampling the traveling wave solution to the ideal wave equation and linearly filtering this solution to simulate dispersive effects due to stiffness and frequency-dependent loss; such digital filters may terminate the waveguide or be embedded along its length. For strings of high stiffness, however, dispersion filters can be difficult to design and expensive to implement. In this article, we show how high-quality time-domain terminating filters may be derived from given frequency-domain specifications which depend on the model parameters. Particular attention is paid to the problem of phase approximation, which, in the case of high stiffness, is strongly nonlinear. Finally, in the interest of determining the limits of applicability of digital waveguide techniques, we make a comparison with more conventional finite difference schemes, in terms of computational cost and numerical dispersion, for a set of string stiffness parameters.

1. Introduction

There are several models which are suitable for the simulation of transverse wave propagation on a string. In the case of a piano string, any such model must take into account loss and stiffness in the string itself; these characteristics respectively lead to attenuation and dispersion of wave propagation. From a perceptual point of view, these phenomena determine the decay time and tuning of partial overtones in the sound, and are therefore crucial. In this article, we will discuss two such models which are currently used for string synthesis, namely finite difference (FD) schemes as applied to a model partial differential equation and so-called digital waveguides (DWG).

A finite difference (FD) model of transverse string motion is usually based on the 1D wave equation, accompanied by several perturbation terms which allow for the modeling of the loss and dispersion effects mentioned above. Ruiz [1] was the first to apply such a model in the context of musical sound synthesis of vibrating strings, and the resulting finite difference schemes were able to reproduce measured waveforms with remarkable accuracy [2, 3]. The digital waveguide approach proposed by Smith [4, 5, 6] first solves the ideal wave equation in terms of fixed traveling waveshapes, and then introduces linear time-invariant filters to implement loss and dispersion effects; these traveling-wave filters effectively add loss and dispersion terms to the linear constant-coefficient wave equation being simulated. Due to commutativity of linear time-invariant systems [7], loss and dispersion filters may be *lumped* at the boundaries of the string [8, 9]. We will call these the *terminating filters* for a digital waveguide string model. Due to the weak damping and dispersion in typical stringed musical instruments, perceptually accurate terminating filters can often be designed with very low order, giving the digital waveguide method a considerable computational advantage over finite difference schemes based more explicitly on the wave equation PDE [10]. As a result, waveguide methods are often the first choice for real-time sound synthesis applications [11, 12, 13]. The digital waveguide approach has been applied to piano synthesis previously [14, 15, 16, 17, 9, 18].

We have already presented work which links the FD and DWG techniques [10], using a model slightly different from that of Ruiz; it was shown that it is possible to derive the terminating filter frequency response for a waveguide structure directly in terms of the defining equations of the FD model. In addition, a comparison was made between finite differences and the waveguide model in the case for which the terminating filter was implemented in the frequency domain (via the discrete Fourier transform, or DFT). The resulting comparison was extremely favorable to the waveguide approach, in that there was a complete

lack of numerical dispersion[1], which is characteristic of most finite difference schemes for stiff strings, and which can be highly audible. For real-time synthesis, however, frequency-domain filtering may be out of the question as it introduces latency and is quite costly from a computational point of view. In practice, some time-domain implementation of the terminating filter (of low order) may be necessary in order to approximate the exact frequency response derived from the PDE itself. Because, for stiff strings, the approximation of the ideal phase response requires, in general, at least a moderately high filter order, it is clear that there is more work to be done in comparing difference schemes with a digital waveguide implementation.

Beginning from the PDE proposed in [10], we have developed a finite difference scheme as well as a terminated waveguide, operating purely in the time domain; in this paper we will compare the frequency domain characteristics of the two methods assuming equal computational costs for each. We begin with a brief review of the digital waveguide method applied to string vibration (section 2), followed by a description of our time-domain terminating filter approximation methods in section 3. Section 4 presents an improved finite difference scheme for the model PDE, and relative computational costs are compared in section 5, where the error criterion is considered to be the deviation of phase-velocity curves from the ideal behavior of the model equation itself. Such comparisons are carried out for several different string stiffnesses (i.e., inharmonicities), where, again, the waveguide and finite difference schemes are set to incur identical computational costs.

2. Digital Waveguide String Models

2.1. Basic principles

The digital waveguide model for vibrating strings is based on sampling the traveling-wave solution of the ideal wave equation. The wave equation for the ideal string may be written physically as

$$Ky'' = \epsilon\ddot{y}, \tag{1}$$

where $K \triangleq$ string tension, $y \triangleq y(t,x)$,
$\epsilon \triangleq$ linear mass density, $\dot{y} \triangleq \frac{\partial}{\partial t}y(t,x)$,
$y \triangleq$ string displacement, $y' \triangleq \frac{\partial}{\partial x}y(t,x)$,

and $y(t,x)$ denotes the transverse displacement of the string in one plane at time t and position x. (The symbol "$\triangleq$" means "equal by definition".) Note that Ky'' is the restoring force per unit length for transverse string displacement, and $\epsilon\ddot{y}$ is the transverse acceleration per unit length. In other words, this form of the wave equation merely states Newton's second law "force equals mass times acceleration" on a per-length basis for transverse waves. It was shown by d'Alembert [19] that equation (1) is satisfied by any fixed waveshape traveling at speed $c = \sqrt{K/\epsilon}$ along the string:

$$y(t,x) = y_r(t - x/c) + y_l(t + x/c). \tag{2}$$

Here, y_r and y_l denote the right-going and left-going traveling-wave components, respectively. Note that $\ddot{y}_r = c^2 y_r''$ and $\ddot{y}_l = c^2 y_l''$.

To simulate these traveling waves digitally, we may *sample* them at intervals of T seconds. The spatial sampling interval X is best chosen as the distance sound propagates in one temporal sampling interval T, i.e., $X = cT$ meters. With this choice, ideal traveling waves simply shift right or left by one spatial sample for each time sampling period. As a result, a digital implementation of an ideal string requires nothing more than two digital delay lines, one for the left-going wave and one for the right-going wave, as shown in Figure 1. It is important to note that a delay line can be implemented in $\mathcal{O}(1)$ operations per sample. Lossy and/or dispersive strings are then simulated by inserting two terminating filters, one at the output of each delay line (before the tap leading to the output sum) in order to implement the appropriate amount of attenuation and delay at each frequency corresponding to lossy, dispersive strings. The design of these filters will be discussed in the next section.

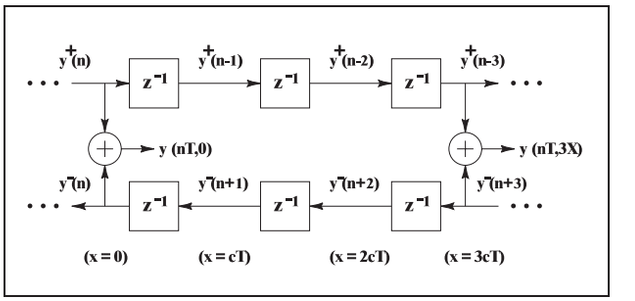

Figure 1. Digital waveguide simulation of a section of ideal vibrating string with physical outputs at $x = 0$ and $x = 3X = 3cT$ (from [20]). (The symbol "z^{-1}" denotes a one-sample delay.)

Sampling the traveling-wave solution of the wave equation equation (2) gives

$$\begin{aligned} y(t_n, x_m) &= y_r(t_n - x_m/c) + y_l(t_n + x_m/c) \\ &= y_r(nT - mX/c) + y_l(nT + mX/c) \\ &= y_r\left[(n-m)T\right] + y_l\left[(n+m)T\right] \\ &\triangleq y^+(n-m) + y^-(n+m), \end{aligned} \tag{3}$$

and Figure 1 shows a signal flow diagram for a section of ideal string simulation. The right-going traveling-wave simulation shifts along the top rail, while the left-going

[1] Note that the *numerical dispersion* associated with finite difference schemes refers to undesired, non-physical signal components that are generated as a result of differentials being replaced by finite differences when converting a PDE to a finite difference scheme. Numerical dispersion can be considered as arising from non-ideal signal interpolation. *Traveling-wave dispersion* associated with stiff strings, on the other hand, is a desired physical phenomenon that we wish to simulate very accurately (to the extent it can be perceived in a piano tone).

wave shifts along the bottom rail in the figure. Physical "transverse displacement" outputs are shown at spatial sample positions $x = 0$ and $x = 3X$. String displacement for any time and position sample may be computed as

$$y(t_n, x_m) = y^+(n-m) + y^-(n+m). \quad (4)$$

When losses and dispersion are included, a terminating filter is necessary, in general, at each delay-line output which is fed to an output tap.

It is a valuable simplification that the traveling-wave signals $y^+(n)$ and $y^-(n)$ are defined as functions of time only—spatial indices as in $y(nT, mX)$ have been eliminated. The spatial sample index persists in the overall simulation, however, as a physical interpretation of the left-right direction in the signal processing diagram.

The digital waveguide simulation for the ideal string is *exact* at the sampling instants, to within the numerical precision of the signal samples. When losses and dispersion are included, the only error (other than simple round-off error) is the difference between the ideal propagation frequency response and that of the terminating filter. To avoid aliasing associated with sampling, the traveling waveshapes must be *bandlimited* to less than half the sampling rate $1/T$. In other words, the spectral content of the signals $y_r(t)$ and $y_l(t)$ in equation (2) may not exceed half the temporal sampling frequency $f_s = 1/T$; equivalently, the highest spatial frequencies in the shapes $y_r(x/c)$ and $y_l(x/c)$ may not exceed half the spatial sampling frequency $1/X$.

2.2. Inputs

Digital waveguide models are typically driven by an external signal at a single spatial point, as if an external excitation force were applied at a single point of the string. However, it is straightforward to excite a spatial region (multiple spatial samples) as well, and this is sometimes done [21, 22]. One reason spatially distributed excitation is rarely used is that, at least for plucked and struck strings, the spatially distributed excitation can be implemented using a half-amplitude pulse of the same shape entering at one sample point to the left and right. That is, a rounded pulse hitting several adjacent samples simultaneously is typically implemented using a rounded pulse entering at one point over time. Since this technique, although well known, appears not to have been carefully described in the prior literature, we include some discussion of it here.

Consider an additive string excitation at time $t = 0$,

$$y(0, x_m) \leftarrow y(0, x_m) + w(x_m),$$

where $w(\cdot)$ is a spatial excitation pulse of width a meters, and assume the pulse is centered spatially about $x = x_c$. In a digital waveguide simulation, half of this pulse may be summed into the right- and left-going delay lines at time 0:

$$y_r(0, x_m) \leftarrow y_r(0, x_m) + \frac{1}{2}w(x_m),$$
$$y_l(0, x_m) \leftarrow y_l(0, x_m) + \frac{1}{2}w(x_m),$$

where, for notational clarity, we have defined

$$y_r(t_n, x_m) \triangleq y_r(t_n - x_m/c) \triangleq y^+(n-m),$$
$$y_l(t_n, x_m) \triangleq y_r(t_n + x_m/c) \triangleq y^-(n+m).$$

For simplicity, consider a string initially at rest, i.e., $y(t_n, x_m) = \dot{y}(t_n, x_m) = 0$ for all x_m, and $t_n < 0$. Then the initial string displacement is $y(0, x_m) = w(x_m)$ and the subsequent traveling-wave state becomes

$$\begin{aligned} y(t_n, x_m) &= y_r(t_n - x_m/c) + y_l(t_n + x_m/c) \\ &= \frac{1}{2}w(x_m - ct_n) + \frac{1}{2}w(x_m + ct_n). \end{aligned}$$

That is, a half-amplitude excitation pulse propagates away to the left and right at speed c. To implement this excitation using a time-domain pulse driving one point at $x_m = x_c$, we write instead

$$y_r(t_n - x_c/c) \leftarrow y_r(t_n - x_c/c) + \frac{1}{2}w(x_c - ct_n)$$
$$y_l(t_n + x_c/c) \leftarrow y_l(t_n + x_c/c) + \frac{1}{2}w(x_c + ct_n).$$

These single-point excitation pulses are centered about time $t_n = 0$, and the copy entering the left-going path goes in *flipped* (time reversed) relative to the one entering the right-going path. For $t_n > a/(2c)$, i.e., after the entire pulse has been injected into the string, we have the subsequent traveling-wave state

$$\begin{aligned} y(t_n, x_m) &= y_r(t_n - x_m/c) + y_l(t_n + x_m/c) \\ &= \frac{1}{2}w(x_m - ct_n) + \frac{1}{2}w(x_m + ct_n). \end{aligned}$$

as before. An analogous technique exists for additive *velocity* excitations, corresponding to "striking" the string as opposed to "plucking" it [22]. Thus, when additively exciting a traveling-wave simulation, instantaneous spatial excitations may be converted to single-point excitations over time. By superposition, the technique extends also to sequences of spatially distributed excitations over time.

For further details regarding digital waveguide modeling of vibrating strings, see, e.g., [15, 20, 13].

2.3. Relating the Digital Waveguide Model to PDE Parameters

As mentioned in section 1, it was shown in previous work [10] that the digital waveguide parameters may be explicitly related to the parameters of a PDE model. We extend those results here to a time domain implementation of the DWG terminating filter (section 3) and also present an improved finite difference scheme based on the PDE (section 3).

In [10] we presented a model of string vibration, second order in time, which models frequency-dependent loss via a mixed time-space derivative term. It was then quite easy to obtain explicit formulas for dispersion and loss

curves, allowing the identification of the PDE with a digital waveguide model. The proposed PDE model is

$$\frac{\partial^2 y}{\partial t^2} = c^2 \frac{\partial^2 y}{\partial x^2} - \kappa^2 \frac{\partial^4 y}{\partial x^4} - 2b_1 \frac{\partial y}{\partial t} + 2b_2 \frac{\partial^3 y}{\partial x^2 \partial t}. \quad (5)$$

The first term on the right-hand side of the equation gives rise to wave-like motion with speed c. The second term introduces dispersion, or frequency-dependent wave velocity, and is parameterized by a stiffness coefficient κ, which is related to physical properties of the string by $\kappa^2 = ESr_g^2/\epsilon$, where E is Young's modulus, S is the cross-sectional area of the string, and r_g is its radius of gyration (half the cross-section radius for a circular string) [23]. (As in equation (1), ϵ is density in mass per unit length.) The third and fourth terms allow for loss (b_1 and b_2 are the loss parameters). A complete model should include a hammer excitation force term, as has been done in, e.g., [24, 25, 2, 26, 27]. In terms of the inharmonicity factor B and string length L, the terminating filter for a DWG model is expressed as

$$|F(\omega)| \simeq \exp\left(-D\left[b_1 + \frac{b_2\pi^2\xi}{2BL^2}\right]\right) \quad (6a)$$

$$\arg(F(\omega)) \simeq \omega D - \pi\sqrt{\frac{\xi}{2B}} \quad (6b)$$

where D, the minimum phase delay is given by

$$D = L/c$$

and

$$\xi = -1 + \sqrt{1 + 4B\omega^2/\omega_0^2}.$$

The inharmonicity factor B is related to the string stiffness factor κ by [10]

$$B = \kappa^2 \frac{\omega_0^2}{c^4}$$

where ω_0 denotes the fundamental frequency of string vibration in radians per second when there is no stiffness (ideal string case).

Expressions (6a–6b) serve to link the PDE model to the lumped terminating filters of the digital waveguide, under the approximations

$$b_1 b_2 \ll c^2 \qquad b_2^2 \ll \kappa^2 \qquad b_1^2 \ll \omega^2$$

which are justifiable for realistic piano strings given that $b_1 \simeq 1$, $b_2 \simeq 10^{-4}$, $c \simeq 200$, $\kappa \simeq 1$ and $\omega \simeq 400$. We note that the modulus and phase of the terminating filter are parameterized by the inharmonicity coefficient B and are nonlinear as a function of the frequency, as expected.

2.4. Dispersion Filter Design

In the context of a digital waveguide string model, dispersion associated with stiff strings can be supplied by an *allpass filter* in the basic feedback loop. A modification of the method in [28] was suggested for designing allpass filters having a phase delay corresponding to the delay profile needed for a stiff string simulation [4, pp. 60,172]. In [2, 3], piano strings were modeled using finite-difference techniques. An update on this approach appears in [10]. In [29], high quality stiff-string sounds were demonstrated using high-order allpass filters in a digital waveguide model. In [17], this work was extended by applying a least-squares allpass-design method [30] and a spectral Bark-warping technique [31] to the problem of calibrating an allpass filter of arbitrary order to recorded piano strings. They were able to correctly tune the first several tens of partials for any natural piano string with a total allpass order of 20 or less. Additionally, minimization of the L^∞ norm [32] has been used to calibrate an allpass-filter cascade [16, 33].

A cost-effective synthesis result is obtained using an allpass filter to correctly tune only the lowest-frequency partial overtones, where the number of partials correctly tuned is significantly less than the total number of partials present [17]. We have used a such approach, as presented in the next section.

3. A Time-Domain Digital Waveguide Implementation

In order to obtain a time-domain implementation of the digital waveguide terminating filter, as given by the relations (6a) and (6b) in the frequency domain, we decompose it into four different filters: one finite impulse response filter (FIR), having a linear phase and taking into account wave attenuation; one allpass filter (or a cascade of allpass filters) taking into account the dispersion of waves; one pure delay, allowing for gross tuning of the whole loop; and one allpass filter of low order, which simulates fractional delay [34]. We then try to find the optimal set of digital filter coefficients of lowest order, accurately fitting the expressions of the modulus and phase of the terminating filter in the frequency domain. The pure delay and the allpass filter simulating the fractional delay do not cause any particular difficulties. We have therefore only considered the modulus and phase approximations in this section.

3.1. Modulus Approximation of the Terminating Filter

The modulus of the terminating filter is simulated using a linear phase FIR filter. In order to reduce the cost of real-time computation, the number of filter coefficients must be minimized. Optimal filter coefficients were obtained using the `firls` function from *Matlab*. For a given filter structure, of a predetermined number of coefficients, this function determines the coefficients which yield an amplitude response closest to that of the original (in the mean square sense, using the L^2 norm). In order to increase precision at low frequencies, a weighting function proportional to $1/f$ has been applied. The choice of the number of filter coefficients depends on the note played; we have used 17 coefficients for the first strings of the piano and 9 coefficients for the remainder. The `firls` function requires an

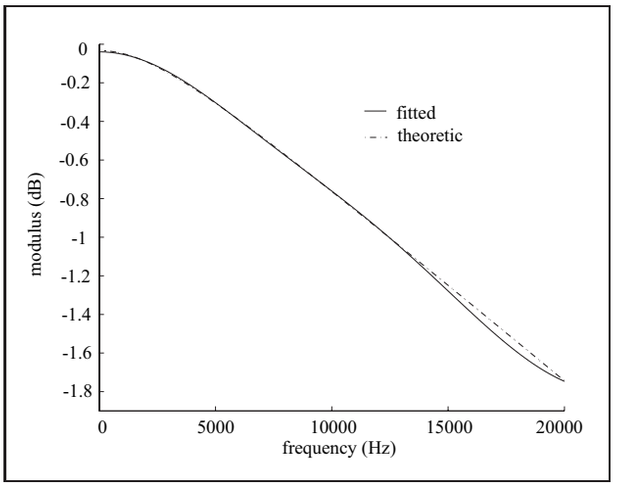

Figure 2. Theoretical modulus of the terminating filter (dotted line) and fitted (solid line) using an eighth-order FIR filter, at C2.

odd number of coefficients, in order to enforce amplitude zero at the Nyquist limit. Figure 2 shows the theoretical amplitude of the terminating filter and the approximation using an eighth-order FIR filter, for a C2 string.

Since the phase is linear, the impulse response of the FIR filter we use is symmetric and it is therefore possible to reduce the number of operations. Considering the transfer function

$$H(z) = \sum_{k=0}^{N} a_k z^{-k}, \tag{7}$$

then for N even, we can write $a_k = a_{N-k}$ with $k \neq N/2$ so that

$$H(z) = \sum_{k=0}^{\frac{N}{2}-1} a_k \left(z^{-k} + z^{-(N-k)} \right) + a_{\frac{N}{2}} z^{-\frac{N}{2}}. \tag{8}$$

The number of MACs (multiplication accumulation operations) is then reduced from $N+1$ for equation (7) to $N/2+1$ for equation (8). For $z = \exp(i\omega)$, equation (8) can be written as

$$H(\mathrm{e}^{\mathrm{i}\omega}) = \mathrm{e}^{-\mathrm{i}\omega\frac{N}{2}} \left(a_{\frac{N}{2}} + \sum_{k=0}^{\frac{N}{2}-1} 2a_k \cos\left(\omega\left(k - \frac{N}{2}\right)\right)\right).$$

We have used this formulation in order to implement the FIR filter in section 5. Notice that this filter creates a phase delay τ_{FIR} of $N/2$ samples. It is necessary to account for this delay when calculating the global filter.

3.2. Phase Approximation of the Terminating Filter

Thick piano strings cause dispersion when waves propagate, leading to inharmonic partial overtones. In terms of waveguides, this inharmonicity is accounted for in the phase (necessarily nonlinear) of the terminating filter. In order to approximate the theoretical phase of the terminating filter in the time domain, we have used a cascade of allpass filters. The allpass filter allows the approximation of a given phase response, without any modification of the magnitude response (which has already been dealt with by the FIR filter). In real-time simulations we need to limit the number of filter coefficients. Although it is possible to cheaply simulate strings of weak inharmonicity (such as those of guitars, violin and cellos), it is much more difficult, in the case of the piano strings, to obtain the exact filter phase for a given number of coefficients. Using a cascade of allpass filters allows a simplification of the optimization process. The original phase is divided by the number of allpass sections in the cascade. The phase behavior is thus more linear, allowing the optimization procedure to work more efficiently with a reduced number of coefficients.

In order to find the coefficients of the allpass filter that best approximates the phase, we use an optimization procedure proposed by M. Lang [32] and recently discussed in [16, 17, 33]. Calling k_{AP} the number of allpass coefficients in cascade, and given an Nth-order all-pass filter whose transfer function is

$$H(z) = \frac{z^{-N} P(z^{-1})}{P(z)}, \tag{9}$$

with $P(z) = \sum_{k=0}^{N} a_k z^{-k}$ and $a_0 = 1$. The phase response is given by:

$$\begin{aligned} \phi_{AP} &= \arg(H(\mathrm{e}^{\mathrm{i}\omega})) \\ &= -N\omega + 2\ \arctan\ \frac{\sum_{k=0}^{N} a_k \sin(k\omega)}{\sum_{k=0}^{N} a_k \cos(k\omega)}. \end{aligned} \tag{10}$$

As proposed by Lang, we minimize (using the Remez exchange algorithm) the L^∞-norm of the weighted phase error between the phase ϕ_F of the original terminating filter and the phase ϕ_{AP} of the allpass filter

$$\min_{a_k, \tau_0} \| W(\omega) \left(\phi_F / k_{AP} - \tau_0 \omega - \phi_{AP} \right) \|_\infty$$

$W(\omega) = 1/\omega$ is the weighting function, which allows us to emphasize the importance of good modeling at low frequencies. $\tau_0\omega$ is the pure delay extracted from ϕ_F/k_{AP} in order to fit the phase of the allpass filter. Figure 3 shows the theoretical phase ϕ_F and the fit $k_{AP}(\phi_{AP} + \tau_0\omega)$ at C2, using three third-order allpass filters.

The phase is approximated well only for low frequencies; for middle-range and high frequencies, the inharmonicity of the synthesized sound will be much weaker than that of the original signal. To get an improved approximation, we could use more allpass filters or increase the order of individual sections (see section 5). For real-time applications, however, we have to use a small number of low-order filters.

We also need to take into account the delay introduced by the filters in cascade. For a cascade of k_{AP} allpass filters, the delay introduced at fundamental frequency ω_0 is given by $\tau_{AP} = k_{AP}\Phi_{AP}(\omega_0)/\omega_0$. Using equation (10),

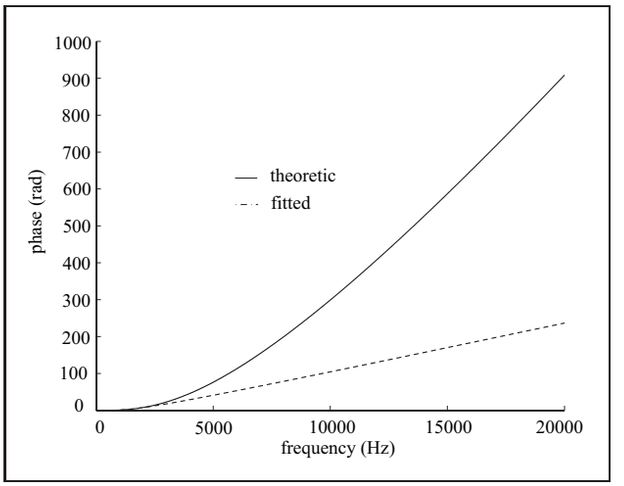

Figure 3. Theoretical phase of the terminating filter (solid line) and fitted (dotted line) using a cascade of four IIR filters of third order at C2.

we have

$$\tau_{AP} = -k_{AP}N + \frac{2}{\omega_0} \arctan \frac{\sum_{k=0}^{N} a_k \sin(k\omega_0)}{\sum_{k=0}^{N} a_k \cos(k\omega_0)}.$$

As in the case of the linear-phase FIR filter, it is possible to minimize the number of operations for the allpass filter. Equation (9) can be written as

$$H(z) = \frac{Y(z)}{X(z)} = \frac{z^{-N} + \sum_{k=1}^{n} a_k z^{k-N}}{1 + \sum_{k=1}^{N} a_k z^{-k}},$$

and the corresponding time equation becomes

$$y(n) = x(n-N) + \sum_{k=1}^{N} a_k \left(x(n-N+k) - y(n-k)\right).$$

Using this formulation, it is necessary to perform N MACs instead of $2N$. We have used this formulation in section 5.

4. A Finite Difference Scheme

We now return to the PDE model discussed in section 2.3. The numerical solution of this equation may also be approached using finite difference schemes [35, 2].

An explicit finite difference scheme corresponding to equation (5) follows immediately; it can be written as

$$\begin{aligned} u_i^{n+1} = {} & B_0 u_i^n + B_1 \left(u_{i+1}^n + u_{i-1}^n\right) + B_2 \left(u_{i+2}^n + u_{i-2}^n\right) \\ & + A_0 u_i^{n-1} + A_1 \left(u_{i-1}^{n-1} + u_{i+1}^{n-1}\right), \end{aligned} \tag{11}$$

where u_i^n is a grid function, representing the approximate solution at time $t = nT$ and at location $x = iX$. Here, T is the time step, and X is the spacing between adjacent grid points. The scheme coefficients are given by

$$B_0 = \frac{2(1-\lambda^2) - 6(\mu^2 - r\lambda^2) - 4b_2\mu/\kappa}{1 + b_1 T},$$

$$B_1 = \frac{\lambda^2 + 4(\mu^2 - r\lambda^2) + 2b_2\mu/\kappa}{1 + b_1 T},$$

$$B_2 = -\frac{\mu^2 - r\lambda^2}{1 + b_1 T},$$

$$A_0 = \frac{b_1 T + 4b_2\mu/\kappa - 1}{1 + b_1 T},$$

$$A_1 = -\frac{2b_2\mu/\kappa}{1 + b_1 T},$$

where we have introduced the extra parameters

$$\lambda = \frac{cT}{X} \qquad \mu = \frac{\kappa T}{X^2}.$$

The scheme reduces to the scheme given in [10] when $r = 0$.

Scheme (11) is an explicit two-step method, with a stencil width of five points. It is similar to the scheme presented in [10], except for the additional term of coefficient r, which is to be set to a constant independent of X or T. This parameter allows us some freedom when attempting to optimize phase velocity curves; parameterized methods of this type, applied to the wave equation, have been discussed in a recent publication [36].

Stability analysis can be carried out, as is usual, through the use of spectral, or von Neumann techniques [35, 37, 38], In the case $r \geq 0$, a sufficient condition for stability can be shown to be

$$\lambda^2(1 - 4r) + 4\mu^2 + 4b_2\mu/\kappa \leq 1.$$

This leads to a lower bound on X, the grid spacing, in terms of the time step T, given by

$$\begin{aligned} X^2 \geq {} & \Big[c^2T^2(1-4r) + 4b_2T \\ & + \sqrt{(c^2T^2(1-4r) + 4b_2T)^2 + 16\kappa^2T^2}\Big]/2. \end{aligned}$$

For pinned boundary conditions (i.e., $u = \partial^2/u\partial x^2 = 0$ at both ends), difference scheme (11) must be specialized near the boundaries. Considering the left end termination at $x = 0$, and assuming that the grid variable u_0^n is aligned with the boundary, we clearly may set $u_0^n = 0$ for all n. In order to update u_1^n, however, the following boundary scheme may be used:

$$\begin{aligned} u_1^{n+1} = {} & B_0 u_1^n + B_1 u_2^n + B_2 \left(u_3^n - u_1^n\right) \\ & + A_0 u_1^{n-1} + A_1 u_2^{n-1} \end{aligned}$$

This satisfies the second boundary condition mentioned above. A similar scheme may be applied at the right end. The computational complexity in this case is 105 MACs per sample.

5. Accuracy of the Two Numerical Implementations

In this section, we make a comparison of the accuracy of the finite difference and digital waveguide implementations of the stiff string vibration model. For this reason,

we compare implementations which require an identical number of MACs. Since the systems are Linear and Time Invariant (LTI), we compare the methods by looking at the error in the loss and the phase-velocity curves with respect to the ideal curves. This corresponds to exciting the model using a Dirac delta function on the string, thereby exciting all frequencies equally and simultaneously. The system is then entirely described by the loss and the phase-velocity curves in both cases. We have chosen to examine the tone C2 since it contains a large number of partials, ensuring the inharmonicity to be of perceptual importance [39]. The number of MACs per sample has been set to 105, which is realizable in real time. For the DWG model, this allows an eighth order FIR loss filter and twenty IIR dispersion filters of 5th order. We vary the string stiffness in order to make comparisons for different values of the inharmonicity factor B (we refer the reader to [10] regarding the relationship between B and κ). The following four values of the inharmonicity factor B : $B1$ = 1e-4, $B2$ = 1e-3, $B3$ = 1e-2, $B4$ = 1e-1 have been used. The first two values are typical of piano strings while the third and fourth approach bar characteristics. By changing the inharmonicity factor, both the phase velocity and the damping are altered, since they both depend on the string stiffness. For perceptual reasons, we have considered only the bandwidth 20 Hz–10 kHz and a "reduced band" containing the first 40 partials of the C2 tone with inharmonicity $B1$. This band corresponds to the range 65 Hz–3 kHz. This band is of particular importance in that the piano sounds can easily be altered [39].

5.1. Loss Comparison

Figure 4 shows the losses as a function of frequency for, respectively, a weak piano string inharmonicity factor ($B1$ = 1e-4) and a very strong one ($B3$ = 1e-2). Expressing an exponential decay as $\exp(-t/\tau)$, we define the "loss" as $1/\tau$ (the inverse of the time constant of decay). In the case of weak inharmonicity, the DWG model allows a nearly perfect fit to the ideal curve over the entire frequency range. The error resulting from the use of the FD scheme increases rapidly with the frequency. For stronger inharmonicity, the DWG scheme shows discrepancies in the reduced frequency band. Nevertheless, as the error is only near 0.25s^{-1}, the perceptual quality of the synthetic sound should remain high.

5.2. Phase Velocity Comparison

For weak inharmonicity, the DWG model again yields a nearly perfect fit of the phase-velocity curve to the ideal curve (Figure 5, top). The FD scheme leads to a good approximation of the phase velocity in the reduced frequency band although a small error is visible at 3 kHz which grows rapidly at higher frequencies. For the higher values $B2$, $B3$ and $B4$ of the inharmonicity, the DWG scheme shows visible deviations in the reduced frequency band, while the FD scheme seems to behave the same as it does in the weak inharmonicity case (Figure 5, bottom and Figure 6).

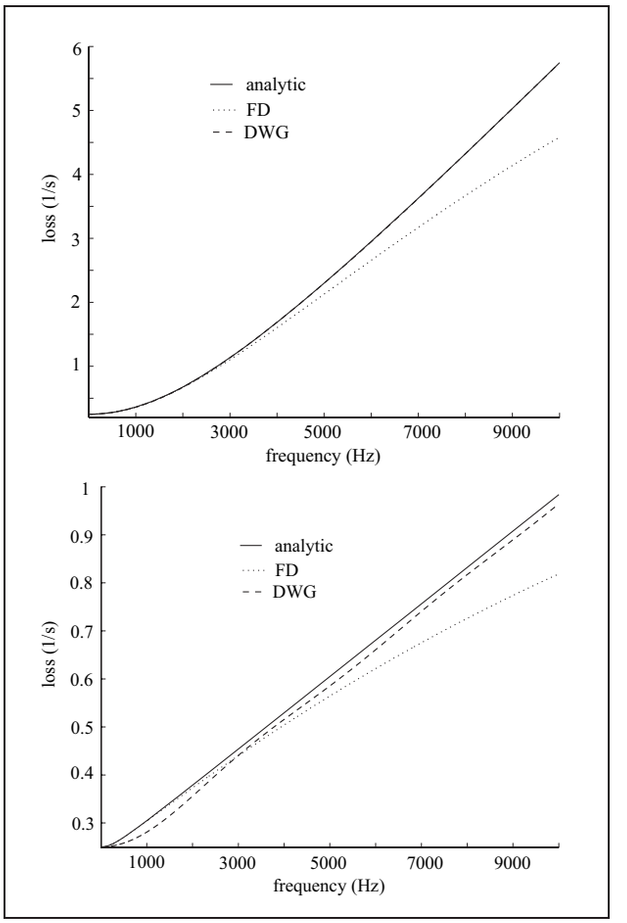

Figure 4. Loss curves in the analytic case (solid line), for the FD scheme (dotted line) and the DWG (dash-dotted), for inharmonicities B1 (top) and B3 (bottom).

These results are perhaps more easily viewed in plots of the phase-velocity error for each value of the parameter B (see Figures 7 and 8).

It is easy to see that the FD scheme leads to a significant error in the phase velocity at high frequencies (above 3 kHz), regardless of the value of the inharmonicity factor. The DWG scheme shows an oscillating error value for high stiffness, and its error is larger than that of the FD scheme below ≈ 1.4 kHz for B3 and ≈ 3 kHz for B4. These results are highly dependent upon the weighting function used in the allpass filter design for the DWG terminating filter. Between these frequencies up to 10 kHz, the FD scheme has the larger error, and it grows more systematically with increasing frequency. The audibility of these errors at high frequencies (which takes the form of mistuned high-frequency partials) is not great; in fact, the high-frequency error may not be perceivable at all in a practical piano synthesis system.

5.3. Choice of Numerical Implementation

While the numerical experiments presented above do not conclusively distinguish the characteristics of the FD scheme and the DWG, some tentative conclusions can be ventured. These conclusions are, of course, dependent on

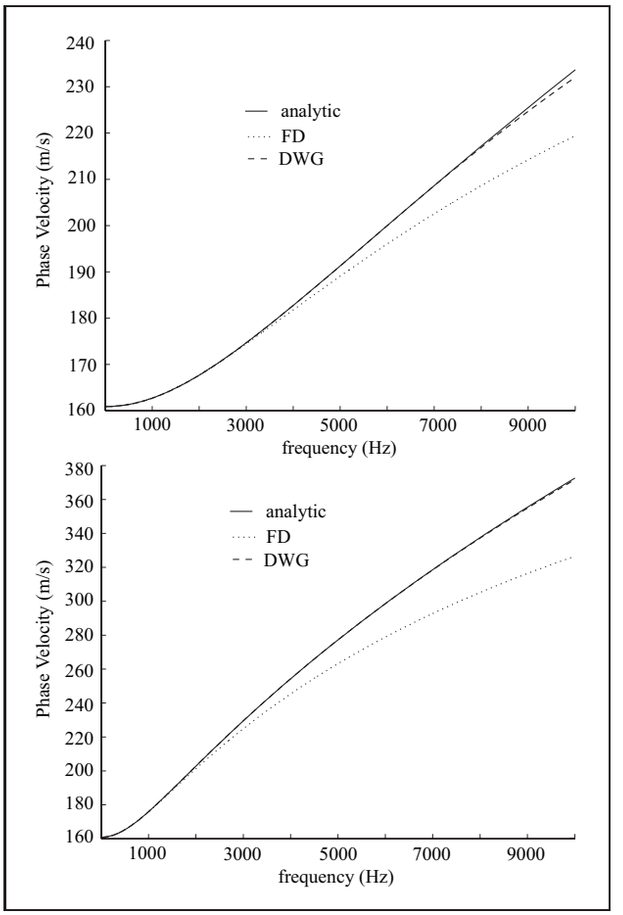

Figure 5. Phase-velocity curves in the analytic case (solid line), for the FD scheme (dotted line) and the DWG (dash-dotted line), for inharmonicity factors B1 (top) and B2 (bottom).

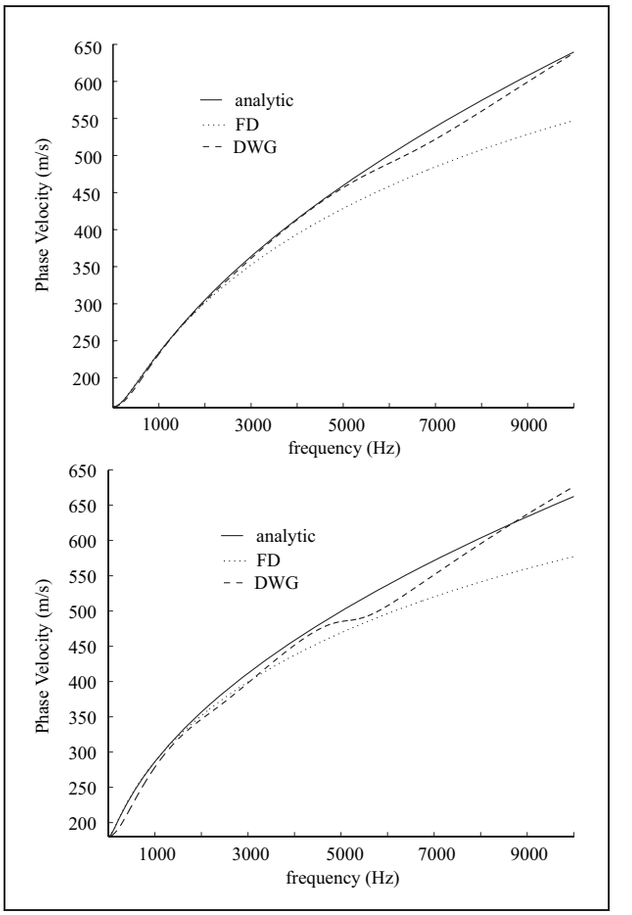

Figure 6. Phase-velocity curves in the analytic case (solid line), for the FD scheme (dotted line) and the DWG (dash-dotted line), for inharmonicity factors B3 (top) and B4 (bottom).

current implementation techniques. In addition, ongoing work is aimed at the improvement of phase velocity approximation by digital filters.

- For weak inharmonicity (corresponding to typical piano string stiffness), the DWG scheme gives improved accuracy with respect to the FD scheme at equal computational cost (i.e., number of MACs) for both the damping and the phase velocity.
- For inharmonicity larger than approximately 0.01, the FD scheme gives better results. More precisely, the error given by this scheme is smaller in the critical reduced band.

As far as sound synthesis is concerned, DWG is more efficient for piano strings while FD schemes should be used for stiff bars. It is also important to note the intrinsic difference between the methods, i.e., the FD scheme yields an approximation to the entire state of the string, whereas the DWG is an approximation at a single (or perhaps a few) chosen locations. For full state recovery, the digital waveguide complexity approaches that of a more conventional finite difference scheme.

6. Conclusions

This paper has reviewed digital waveguide methods and finite difference methods for the computational simulation of vibrating piano strings. For strings found in typical musical instruments, the digital waveguide approach was found to be more efficient than conventional finite difference methods. However, when the stiffness is very high, or when the simulation must produce many outputs, conventional finite difference schemes remain competitive.

For future work, the design of DWG dispersion filters remains an open area. The goal is to minimize computational cost while maintaining perceptually ideal performance. While the audibility of stiff-string inharmonicity has been measured [39], the audibility of various types of approximation by different dispersion-filter design strategies has not yet been carefully investigated. In particular, one could seek perceptually optimal weightings in the least-squares filter-design procedure used in this paper.

References

[1] P. M. Ruiz: A technique for simulating the vibrations of strings with a digital computer. Ph.D. thesis, University of Illinois, 1970.

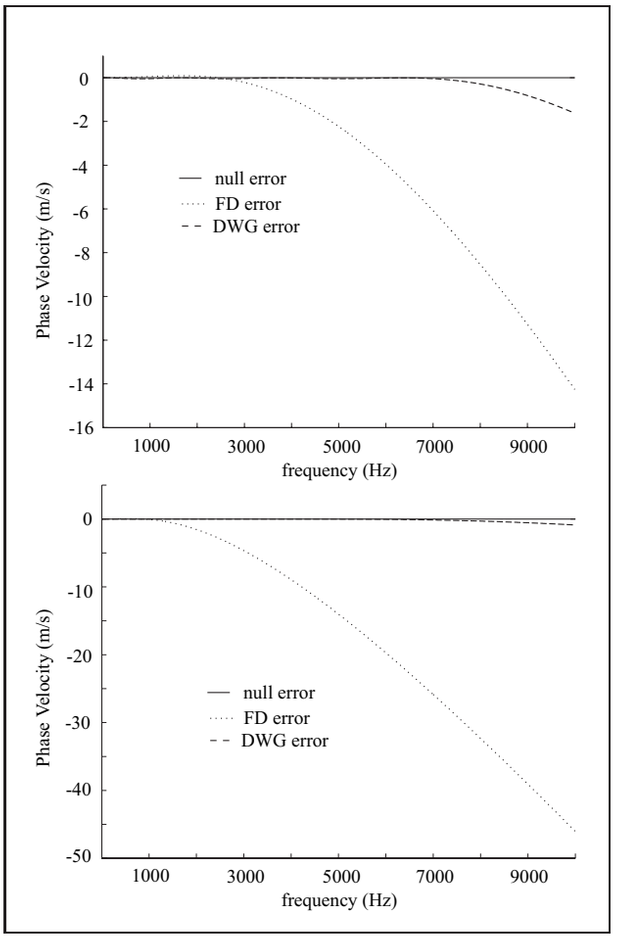

Figure 7. Phase-velocity error curves for the FD scheme (dotted line) and the DWG (dash-dotted line), for inharmonicities B1 (top) and B2 (bottom).

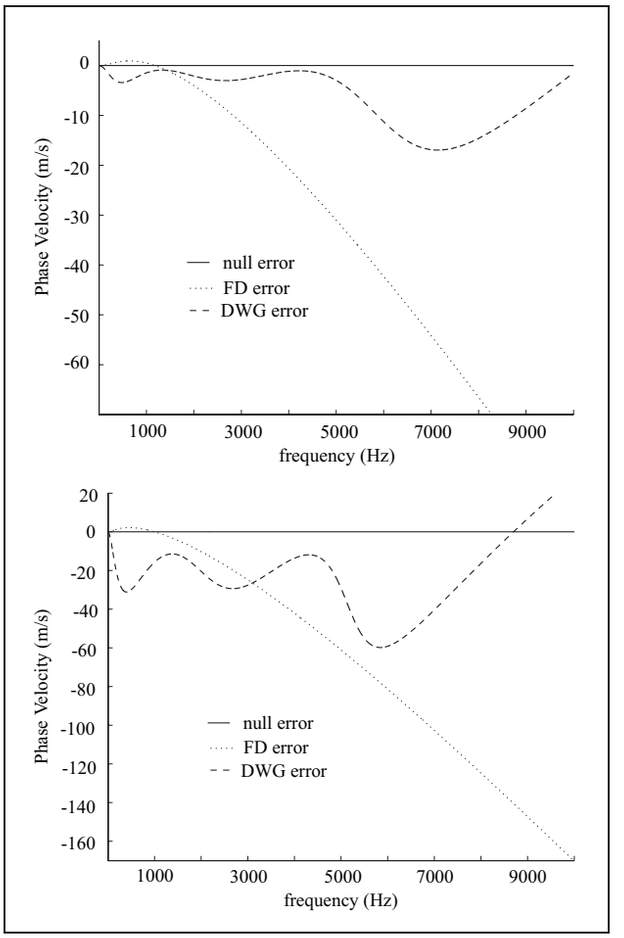

Figure 8. Phase-velocity error curves for the FD scheme (dotted line) and the DWG (dash-dotted line), for inharmonicities B3 (top) and B4 (bottom).

[2] A. Chaigne, A. Askenfelt: Numerical simulations of struck strings. I. A physical model for a struck string using finite difference methods. J. Acoust. Soc. Amer. **95** (1994) 1112–8.

[3] A. Chaigne, A. Askenfelt: Numerical simulations of struck strings. II. Comparisons with measurements and systematic exploration of some hammer-string parameters. J. Acoust. Soc. Amer. **95** (1994) 1631–40.

[4] J. O. Smith: Techniques for digital filter design and system identification with application to the violin. Ph.D. thesis, Elec. Eng. Dept., Stanford University (CCRMA), CCRMA Technical Report STAN-M-14, URL: http://-ccrma.stanford.edu/STANM/stanm/, June 1983.

[5] D. A. Jaffe, J. O. Smith: Extensions of the Karplus-Strong plucked-string algorithm. Computer Music Journal **7** (1983) 56–69.

[6] J. O. Smith: Physical modeling using digital waveguides. Computer Music Journal **16** (1992) 74–91. Special issue: Physical Modeling of Musical Instruments, Part I. URL: http://ccrma.stanford.edu/ jos/pmudw/.

[7] J. O. Smith: Introduction to digital filters. URL: http://-ccrma.stanford.edu/ jos/filters/, May 2004.

[8] J. O. Smith: Efficient synthesis of stringed musical instruments. Proc. Int. Computer Music Conf., Tokyo, Computer Music Association, 1993, 64–71. URL: http://ccrma.-stanford.edu/ jos/cs/.

[9] J. O. Smith, S. A. Van Duyne: Commuted piano synthesis. Proc. Int. Computer Music Conf., Banff, Computer Music Association, 1995, 319–326.

[10] J. Bensa, S. Bilbao, R. Kronland-Martinet, J. O. Smith: The simulation of piano string vibration: from physical model to finite difference schemes and digital waveguides. J. Acoust. Soc. Amer. **114** (2003) 1095–1107.

[11] R. Rideout: Yamaha VL1 virtual acoustic synthesizer. Keyboard Magazine **20** (1994) 104–118.

[12] J. O. Smith: Physical modeling synthesis update. Computer Music Journal **20** (1996) 44–56. URL: http://ccrma.-stanford.edu/ jos/pmupd/.

[13] J. O. Smith: Selected tutorials, papers, programming examples, (some) sound samples, and related links. Digital Waveguide Synthesis Home Page, http://ccrma.stanford.-edu/ jos/wg.html, 2003.

[14] F. Avanzini, B. Bank, G. Borin, G. De Poli, D. Rocchesso: Musical instrument modeling: the case of the piano. Proc. of the worshop on current research directions in computer music, MOSART Reaserach training network, 2001, 124–125.

[15] B. Bank, F. Avanzini, G. Borin, G. De Poli, F. Fontana, D. Rocchesso: Physically informed signal processing methods

for piano sound synthesis: A research overview. EURASIP Journal on Applied Signal Processing **10** (2003).

[16] J. Bensa: Analysis and synthesis of piano sounds using physical and signal models. Ph.D. thesis, Université de la Méditeranée. Available online at URL: http://www.lma.cnrs-mrs.fr/∼bensa, 2003.

[17] D. Rocchesso, F. Scalcon: Accurate dispersion simulation for piano strings. Proc. Nordic Acoust. Meeting (NAM'96), Helsinki, Finland, June 12-14 1996.

[18] S. A. Van Duyne, J. O. Smith: Developments for the commuted piano. Proc. Int. Computer Music Conf., Banff, Computer Music Association, 1995, 335–343.

[19] J. l. d'Alembert: Investigation of the curve formed by a vibrating string, 1747. – In: Acoustics: Historical and Philosophical Development. R. B. Lindsay (ed.). Dowden, Hutchinson and Ross, Stroudsburg, 1973, 119–123.

[20] J. O. Smith: Physical audio signal processing: Digital waveguide modeling of musical instruments and audio effects. URL: http://ccrma.stanford.edu/ jos/pasp/, August 2004.

[21] R. Pitteroff, J. Woodhouse: Mechanics of the contact area between a violin bow and a string, Part II: Simulating the bowed string **84** (1998) 744–757.

[22] J. O. Smith: On the equivalence of digital waveguide and finite difference time domain schemes. URL: http://arxiv.org/abs/physics/0407032/, July 21, 2004.

[23] P. M. Morse: Vibration and sound. American Institute of Physics, for the Acoust. Soc. Amer., URL: http://-asa.aip.org/publications.html, 1948, 1st edition 1936, last author's edition 1948, ASA edition 1981.

[24] X. Boutillon: Model for piano hammers: Experimental determination and digital simulation. J. Acoust. Soc. Amer. **83** (1988) 746–754.

[25] H. Suzuki: Model analysis of a hammer-string interaction. J. Acoust. Soc. Amer. **82** (1987) 1145–1151.

[26] A. Stulov: Hysteretic model of the grand piano hammer felt. J. Acoust. Soc. Amer. **97** (1995) 2577–2585.

[27] N. Giordano, J. P. Millis: Hysteretic behavior of piano hammers. Proc. Int. Symposium of Musical Acoustics, 2001, 237–240.

[28] B. Yegnanarayana: Design of recursive group-delay filters by autoregressive modeling. IEEE Trans. Acoustics, Speech, Signal Processing **30** (1982) 632–637.

[29] A. Paladin, D. Rocchesso: A dispersive resonator in real-time on mars workstation. Proc. Int. Computer Music Conf., San Jose, Computer Music Association, 1992, 146–149.

[30] M. Lang, T. I. Laakso: Simple and robust method for the design of allpass filters using least-squares phase error criterion. IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications **41** (1994) 40–48.

[31] J. O. Smith, J. S. Abel: Bark and ERB bilinear transforms. IEEE Trans. Acoust., Speech, Signal Processing (1999) 697–708. Matlab code for the main figures are available online at URL: http://ccrma.stanford.edu/ jos/bbt/.

[32] M. Lang: Allpass filter design and applications. IEEE Transactions on Signal Processing **46** (1998) 2505–2514.

[33] S. Serafin, J. O. Smith: Impact of string stiffness on digital waveguide models of bowed strings. Catgut Acoustical Society Journal **4** (2001) 49–52.

[34] T. I. Laakso, V. Välimäki, M. Karjalainen: Splitting the unit delay. IEEE Signal Processing Magazine (1996) 30–60.

[35] J. Strikwerda: Finite difference schemes and partial differential equations. Wadsworth and Brooks/Cole Advanced Books and Software, Pacific Grove, Calif., 1989.

[36] S. Bilbao: Stability of parameterized finite difference schemes for the wave equation. Numerical Methods for Partial Differential Equations (2004) in press.

[37] B. Gustaffson, H.-O. Kreiss, J. Oliger: Time dependent problems and difference methods. John Wiley and Sons, New York, 1986.

[38] R. Vichnevetsky, J. Bowles: Fourier analysis of numerical approximations of hyperbolic equations. SIAM, Philadelphia, 1982.

[39] H. Järveläinen, V. Välimäki, M. Karjalainen: Audibility of inharmonicity in strings instruments sounds, and implications to digital sound synthesis. Proc. Int. Computer Music Conf., Beijing, Computer Music Association, 1999, 359–362.