# REAL-TIME GONG SYNTHESIS

*Stefan Bilbao*
Acoustics and Audio Group
University of Edinburgh
Edinburgh, United Kingdom
sbilbao@ed.ac.uk

*Craig Webb*
Physical Audio Ltd.
London, United Kingdom
craig@physicalaudio.co.uk

*Zehao Wang*
Department of Music
University of California San Diego
La Jolla, California, USA
zehaowang@ucsd.edu

*Michele Ducceschi* [*]
Department of Industrial Engineering
University of Bologna
Bologna, Italy
michele.ducceschi@unibo.it

## ABSTRACT

Physical modeling sound synthesis is notoriously computationally intensive. But recent advances in algorithm efficiency, accompanied by increases in available computing power have brought real-time performance within range for a variety of complex physical models. In this paper, the case of nonlinear plate vibration, used as a simple model for the synthesis of sounds from gongs is considered. Such a model, derived from that of Föppl and von Kármán, includes a strong geometric nonlinearity, leading to a variety of perceptually-salient effects, including pitch glides and crashes. Also discussed here are input excitation and scanned multichannel output. A numerical scheme is presented that mirrors the energetic and dissipative properties of a continuous model, allowing for control over numerical stability. Furthermore, the nonlinearity in the scheme can be solved explicitly, allowing for an efficient solution in real time. The solution relies on a quadratised expression for numerical energy, and is in line with recent work on invariant energy quadratisation and scalar auxiliary variable approaches to simulation. Implementation details, including appropriate perceptually-relevant choices for parameter settings are discussed. Numerical examples are presented, alongside timing results illustrating real-time performance on a typical CPU.

## 1. INTRODUCTION

Physical modeling synthesis has now reached a certain level of maturity. It has become possible to perform audio rate simulations of relatively complex systems in real time. One reason for this follows from the steady increase in available computing power, accompanied by newer tools allowing code acceleration using low-level parallelisation on the CPU [1]. More important, though, have been advances in algorithm efficiency, particularly for systems exhibiting a strong nonlinearity, the subject of this paper.

Perhaps the strongest nonlinear mechanism in any acoustic musical instrument is found in gong-like percussion instruments leading to characteristic pitch glides, crashes and swells. A linear model is grossly insufficient to capture such effects, and the nonlinearity is distributed throughout the entire vibrating structure, normally modelled as a thin plate or shell. Physical models have been available for some time [2, 3], and used for sound synthesis purposes [4], but have been limited to offline use. Computational cost is large, due to the essentially 2D nature of such models, and increased further due to the complexity of the geometric nonlinearity, alongside various practical algorithm design constraints.

The most important of these constraints is the requirement for numerically stable behaviour. Though simple efficient explicit time domain simulation methods, such as Störmer-Verlet integration are available [5], stability is not ensured, and indeed such methods are highly prone to explosive instability, particularly at high amplitudes, exactly at the onset of perceptually salient nonlinear effects. One approach to ensuring numerical stability is through the use of energy-conserving numerical designs, where the solution size is bounded by a numerical invariant (the energy or pseudo-energy). In the present case of nonlinear plate vibration, such methods have been proposed, and allow for designs of so-called linearly implicit character—costly iterative methods such as Newton Raphson are avoided, but potentially large linear systems must be both constructed and solved in the run-time loop [6]—real-time performance is ruled out for such methods. A more recent approach follows from invariant energy quadratisation [7, 8] and scalar auxiliary variable [9, 10] methods applied in geometric numerical integration. In general, these also lead to algorithms with the same linearly-implicit character. However, recent results have allowed for fully explicit numerical solutions through the exploitation of structure in the linear system to be solved, increasing the speed of calculation by an order of magnitude at least [11, 12], while maintaining stable numerical behaviour. This paper is concerned with the range of algorithmic and programming techniques necessary in order to generate gong-like sounds in real time.

A model of nonlinear plate vibration at high amplitudes, based on the dynamic analogue of the model of Föppl and von Kármán, and including effects of loss as well as input excitation and scanned multichannel output, is presented in Section 2. An energy balance is also presented. The basic steps leading to a discrete-time simulation algorithm are presented in Section 3, beginning from the definition of a basic spatial grid and difference operators, and proceeding to a semi-discrete form. This form may then be written directly

[*] Michele Ducceschi has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, Grant agreement No. 950084 - NEMUS.

as an extension of a Hamiltonian system. A fully discrete time numerically stable algorithm accompanied by an energy balance may be constructed, and has the advantage that the nonlinearity is dealt with fully explicitly. Approaches to the numerical solution of the remaining required linear system, involving the biharmonic operator are also outlined. Implementation details, including the simplification of user-supplied instrument design and control parameters, are described in Section 4. Numerical results, including timings illustrating real time performance on a standard CPU, as well as spectrograms of representative outputs are provided in Section 5. Concluding remarks appear in Section 6. Sound examples are available at the companion page [1].

## 2. MODEL

Realistic sound synthesis from a gong-like instrument requires a nonlinear model of plate vibration—necessary in order to capture strong amplitude-dependent effects such as pitch glides, crashes and swells. Linear models (such as, e.g. the thin model due to Kirchhoff [13] or even thick models of Mindlin-Reissner form [14]) are insufficient for this purpose. Simplified nonlinear models such as that of Berger [15] are able to replicate pitch glides, but not the energy cascade to high frequencies characteristic of crashes.

The simplest suitable model is the dynamic analogue of the system of Föppl and von Kármán [16, 17, 18], and describes the high amplitude vibration of thin plates. When accompanied by additional terms emulating loss, and a forcing term, it may be written as the following coupled system of partial differential equations:

$$\begin{aligned}\rho\xi\partial_t^2 w &= -Q\Delta\Delta w - 2\rho\xi\sigma_0\partial_t w + 2\rho\xi\sigma_1\partial_t\Delta w\\ &\quad +\mathcal{L}(w,\Phi) + \delta(\mathbf{r}-\mathbf{r}_\mathrm{i})f \qquad (1a)\\ \frac{2}{E\xi}\Delta\Delta\Phi &= -\mathcal{L}(w,w)\,. \qquad (1b)\end{aligned}$$

Here, $w(\mathbf{r},t)$ and $\Phi(\mathbf{r},t)$ are the transverse plate deflection and Airy stress function, respectively. Both are functions of time $t \geq 0$, and spatial coordinates $\mathbf{r} = (x,y) \in \mathcal{D} \subset \mathbb{R}^2$. In this paper, the rectangular domain $\mathcal{D} = [-L_x/2, L_x/2] \times [-L_y/2, L_y/2]$ will be considered, for plate side lengths $L_x$, $L_y$ in m. See Figure 1. The other material and geometric parameters that define the plate are the density $\rho$, in kg m$^{-3}$, the plate thickness $\xi$, in m, and Young's modulus $E$ in Pa—the flexural rigidity $Q$ is defined as $Q = E\xi^3/12(1-\nu^2)$, where $\nu$ is Poisson's ratio for the plate material. The two parameters $\sigma_0$, in s$^{-1}$ and $\sigma_1$, in m$^2$s$^{-1}$ give two-parameter control over frequency-dependent loss [19]—see also Section 4.1. $\partial_t$ represents partial differentiation with respect to time $t$, and $\Delta$ the two-dimensional Laplacian, where $\Delta = \partial_x^2 + \partial_y^2$, for spatial partial derivatives $\partial_x$ and $\partial_y$ with respect to coordinates $x$ and $y$ respectively. $\Delta\Delta$ is the biharmonic operator.

The nonlinear operator $\mathcal{L}$, defined, in terms of its operation on two functions $\alpha(x,y)$ and $\beta(x,y)$, as:

$$\mathcal{L}(\alpha,\beta) = \partial_x^2\alpha\partial_y^2\beta + \partial_y^2\alpha\partial_x^2\beta - 2\partial_x\partial_y\alpha\partial_x\partial_y\beta\,. \qquad (2)$$

For simplicity, boundary conditions are chosen to be of simply supported type over the boundary $\partial\mathcal{D}$ of $\mathcal{D}$, so that

$$w = \Delta w = 0 \qquad \Phi = \Delta\Phi = 0 \qquad \text{for} \qquad \mathbf{r} \in \partial\mathcal{D}\,. \qquad (3)$$

Initial conditions are assumed to be zero, so that

$$w(\mathbf{r},0) = \partial_t w|_{\mathbf{r},t=0} = 0\,. \qquad (4)$$

[1]https://physicalaudio.co.uk/modelling-gongs/

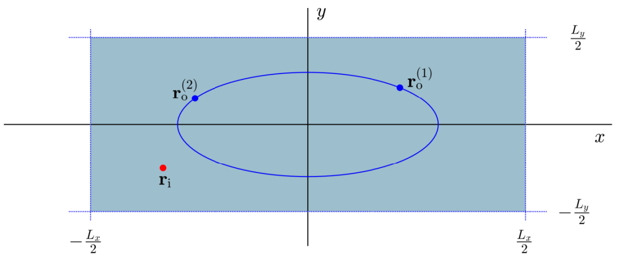

Figure 1: *Plate geometry, with side lengths $L_x$ and $L_y$. The driving location $\mathbf{r}_\mathrm{i}$ is indicated, as well as the two output locations $\mathbf{r}_\mathrm{o}^{(1)}(t)$ and $\mathbf{r}_\mathrm{o}^{(2)}(t)$, drawn from an elliptical trajectory.*

Initial conditions for $\Phi$ do not need to be set independently.

Versions of system (1) have been used in various studies of percussion instruments, and in particular, the cases of circular plates [2] and the generalisation to the case of curved plates or shells [3]. Shells of variable thickness have also been examined in the context of cymbal acoustics [20]. Here, a simplified flat rectangular geometry is chosen, as it reproduces many of the features that are characteristic of gong-like instruments, and also leads to simulation algorithm designs that are very well suited for acceleration.

### 2.1. Input and Output

Also included in system (1) is a point-like excitation term. $f(t)$, in N, is a forcing function applied at location $\mathbf{r}_\mathrm{i} = (x_\mathrm{i}, y_\mathrm{i})$; $\delta$ is a 2-dimensional Dirac delta function. A full model of the interaction between a striking object (such as a mallet) and the plate could be included here, as in earlier models of percussion instruments [21]. Because the interaction time is generally extremely short (on the order of 1-5 ms), a much simpler approach is to model this interaction using an externally supplied excitation function of the form of a short pulse. A suitable candidate is a time-limited raised sinusoidal distribution [19] of the form

$$f(t) = \begin{cases} f_\mathrm{max}\sin^2\left(\frac{\pi(t-t_0)}{T}\right), & t_0 \leq t \leq t_0 + T\\ 0, & \text{otherwise}\end{cases}\,. \qquad (5)$$

See Figure 2. Other excitation functions can be applied—including steady sinusoidal functions, and possibly even audio, in which case the physical model behaves as an effect instead of a synthesizer.

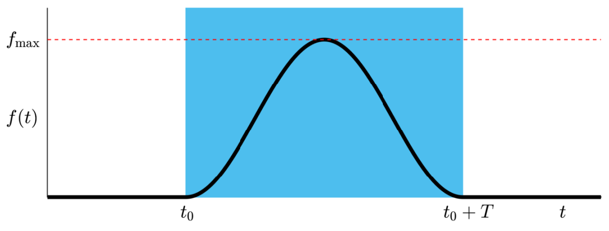

Figure 2: *Excitation function $f(t)$, as defined in* (5).

For output, the picture is slightly different. Output audio signals $w_\mathrm{o}^{(p)}(t)$ can be drawn from the plate at $P$ distinct locations $\mathbf{r}_\mathrm{o}^{(p)}$, $p = 1, \ldots, P$, and defined in terms of plate displacement, as

$$w_\mathrm{o}^{(p)}(t) = w(\mathbf{r}_\mathrm{o}^{(p)}, t)\,. \qquad (6)$$

It can be useful, as an additional effect, to allow the output locations to be time varying. Here, outputs $\mathbf{r}_{\rm o}^{(p)} = (x_{\rm o}^{(p)}, y_{\rm o}^{(p)})$ are drawn from an elliptical distribution as

$$x_{\rm o}^{(p)}(t) = \frac{L_x R}{2}\cos(2\pi f_{\rm o}^{(p)} t + \phi_{\rm o}^{(p)}) \tag{7a}$$

$$y_{\rm o}^{(p)}(t) = \frac{L_y R}{2}\sin(2\pi f_{\rm o}^{(p)} t + \phi_{\rm o}^{(p)}) \,. \tag{7b}$$

Here, $0 \le R < 1$ is a dimensionless parameter controlling the size of the ellipse, $f_{\rm o}^{(p)}$ is a scan frequency (sub audio rate), and $\phi_{\rm o}^{(p)}$ is an initial phase. See Figure 1. If different values are used for each output, with fixed $R$, then a natural phasing effect is obtained, while maintaining uniform normalisation for all channels.

### 2.2. Energy Balance

System (1) satisfies an energy balance of the form

$$\dot{\mathcal{H}} = -\mathcal{Q} + \mathcal{P}\,, \tag{8}$$

where here, a dot indicates ordinary time differentiation. $\mathcal{H}(t)$ is the total stored energy in the plate, $\mathcal{Q}(t)$ is power loss, and $\mathcal{P}(t)$ is input power. These can be defined explicitly [19] as

$$\mathcal{H} = \iint_{\mathcal{D}} \frac{\rho\xi}{2}(\partial_t w)^2 + \frac{Q}{2}(\Delta w)^2 + \frac{1}{2E\xi}(\Delta\Phi)^2 d\mathbf{r} \tag{9a}$$

$$\mathcal{Q} = \iint_{\mathcal{D}} 2\rho\xi\sigma_0(\partial_t w)^2 + 2\rho\xi\sigma_1|\nabla\partial_t w|^2 d\mathbf{r} \tag{9b}$$

$$\mathcal{P} = f\partial_t w|_{\mathbf{r}_{\rm i},t}\,. \tag{9c}$$

Here, $\nabla$ indicates a gradient with respect to $\mathbf{r}$. All are scalar functions. In particular, both $\mathcal{H}$ and $\mathcal{Q}$ are non-negative, meaning that the system is dissipative under zero input conditions (and lossless when $\sigma_0 = \sigma_1 = 0$, meaning that the energy $\mathcal{H}(t)$ is constant).

## 3. FDTD METHODS

### 3.1. Spatial Grid and Difference Operators

As a first step, suppose that the plate surface is discretised with a 2D grid of spacing $h = L_x/N_x$, for some integer $N_x$. Then, set $N_y = \lfloor L_y/h \rfloor$, where $\lfloor\cdot\rfloor$ indicates a flooring operation. Here, for simplicity, the plate side length in the $y$ direction is set to $N_y h \approx L_y$. The semi-discrete grid functions $w_{l,m}(t)$ and $\Phi_{l,m}(t)$, indexed by integers $l$ and $m$ with $1 \le l \le N_x - 1$ and $1 \le m \le N_y - 1$, represent approximations to $w(\mathbf{r},t)$ and $\Phi(\mathbf{r},t)$ at $\mathbf{r} = -1/2(L_x, L_y) + h(l,m)$. Due to the choice of simply supported boundary conditions (3), the grid functions are assumed to take on values of zero at $l = 0$, $l = N_x$, $m = 0$ and $m = N_y$—and thus, such points may be excluded from the algorithm entirely.

For a given grid function $u_{l,m}(t)$, forward and backward spatial differences in the $x$ and $y$ directions, approximating $\partial_x$ and $\partial_y$, are defined (suppressing time dependence) as

$$D_x^{\pm}u_{l,m} = \pm\frac{u_{l\pm1,m} - u_{l,m}}{h} \qquad D_y^{\pm}u_{l,m} = \pm\frac{u_{l,m\pm1} - u_{l,m}}{h}\,. \tag{10}$$

Second derivative approximations follow directly as:

$$D_{xx} = D_x^+ D_x^- \qquad\qquad D_{yy} = D_y^+ D_y^-\,. \tag{11}$$

The Laplacian and biharmonic may then be approximated as

$$D_{\triangle} = D_{xx} + D_{yy} \qquad D_{\triangle\triangle} = D_{\triangle}D_{\triangle}\,. \tag{12}$$

Finally, four approximations $D_{xy}^{ab}$ to $\partial_x\partial_y$ may be defined as

$$D_{xy}^{ab} = D_x^a D_y^b \qquad \text{for} \qquad a,b \in \{+,-\}\,. \tag{13}$$

### 3.2. Semi-discrete Form

Before moving directly to a semi-discrete form, it is useful to recast the $(N_x - 1)\times(N_y - 1)$ grid functions $w_{l,m}(t)$ and $\Phi_{l,m}(t)$ as $N \times 1$ column vectors $\mathbf{w}(t)$ and $\mathbf{\Phi}(t)$, through concatenation of consecutive columns of $w_{l,m}(t)$ and $\Phi_{l,m}(t)$. Here, $N = (N_x - 1)(N_y - 1)$ is the total number of grid points in either grid function. The linear operators $D_{xx}$, $D_{yy}$, $D_{xy}^{ab}$, $D_{\triangle}$ and $D_{\triangle\triangle}$ can thus be represented as sparse $N \times N$ matrices $\mathbf{D}_{xx}$, $\mathbf{D}_{yy}$, $\mathbf{D}_{xy}^{ab}$, $\mathbf{D}_{\triangle}$ and $\mathbf{D}_{\triangle\triangle}$, respectively. Simply supported boundary conditions are assumed directly encoded into these matrices [19]. Also necessary is an approximation to the Dirac delta function which selects the excitation location in (1). This may be represented as an $N \times 1$ column vector $\frac{1}{h^2}\mathbf{j}$. Many approximations to the delta function over a grid are available [22]; for simplicity, excitation is assumed to occur directly at a grid point, so that $\mathbf{j}$ is all zero except for a single value of 1 at the excitation location.

A semi-discrete form of (1) may be written directly as

$$\rho\xi\ddot{\mathbf{w}} = -Q\mathbf{D}_{\triangle\triangle}\mathbf{w} - 2\rho\xi\sigma_0\dot{\mathbf{w}} + 2\rho\xi\sigma_1\mathbf{D}_{\triangle}\dot{\mathbf{w}} + \ell(\mathbf{w},\mathbf{\Phi}) + \frac{1}{h^2}\mathbf{j}f \tag{14a}$$

$$\frac{2}{E\xi}\mathbf{D}_{\triangle\triangle}\mathbf{\Phi} = -\ell(\mathbf{w},\mathbf{w})\,. \tag{14b}$$

Here, a discrete counterpart to the nonlinear operator $\mathcal{L}$, as defined in (2), may be written in terms of its action on two $N \times 1$ vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, as

$$\ell(\boldsymbol{\alpha},\boldsymbol{\beta}) = \mathbf{D}_{xx}\boldsymbol{\alpha}\odot\mathbf{D}_{yy}\boldsymbol{\beta} + \mathbf{D}_{yy}\boldsymbol{\alpha}\odot\mathbf{D}_{xx}\boldsymbol{\beta} - \tfrac{1}{2}\sum_{a,b\in\{+,-\}}\mathbf{D}_{xy}^{ab}\boldsymbol{\alpha}\odot\mathbf{D}_{xy}^{ab}\boldsymbol{\beta} \tag{15}$$

where $\odot$ denotes element-wise multiplication of two vectors. The approximation $\ell$ to $\mathcal{L}$ is bilinear and possesses various important symmetry properties, including that of triple self-adjointness [19] inherited from the continuous operator $\mathcal{L}$—the reader is referred to the literature for further discussion [23].

### 3.3. Energy Balance and Potential Energy Quadratisation

For analysis purposes, it is useful to rewrite the system of ordinary differential equations (14) in terms of displacement $\mathbf{w}$ and momentum $\mathbf{p}$, as:

$$\dot{\mathbf{w}} = \nabla_{\mathbf{p}}H \qquad\qquad \dot{\mathbf{p}} = -\nabla_{\mathbf{w}}H - \mathbf{R}\mathbf{p} + \mathbf{j}f\,. \tag{16}$$

Here, $H(t)$ is the total system energy, defined as

$$H = \tfrac{1}{2M}\mathbf{p}^T\mathbf{p} + V_0 + V' \tag{17}$$

and $\nabla_{\mathbf{p}}$ and $\nabla_{\mathbf{w}}$ represent gradients with respect to $\mathbf{p}$ and $\mathbf{w}$, respectively. The first term in $H$ represents kinetic energy, where $M = \rho\xi h^2$ is the mass per grid point, in kg. $V_0$ and $V'$ represent contributions to the potential energy due to linear and nonlinear effects, respectively, and are defined by

$$V_0 = \tfrac{1}{2}\mathbf{w}^T\mathbf{K}_0\mathbf{w} \qquad V' = \tfrac{1}{2}\mathbf{\Phi}^T\mathbf{K}'\mathbf{\Phi}\,, \tag{18}$$

where

$$\mathbf{K}_0 = Qh^2\mathbf{D}_{\Delta\Delta} > \mathbf{0} \qquad \mathbf{K}' = \tfrac{h^2}{E\xi}\mathbf{D}_{\Delta\Delta} > \mathbf{0}\,. \tag{19}$$

The positive definiteness conditions above hold under simply supported boundary conditions. The plate nonlinearity intervenes through the relationship (14b) between $\mathbf{\Phi}$ and $\mathbf{w}$.

The equations (16) are an extension of Hamilton's equations, including loss and a forcing term. Loss is encoded here through the matrix $\mathbf{R}$, defined as

$$\mathbf{R} = \underbrace{2\sigma_0\mathbf{I}_N}_{\mathbf{R}_0} + \underbrace{-2\sigma_1\mathbf{D}_\Delta}_{\mathbf{R}_1} \geq \mathbf{0}\,, \tag{20}$$

where $\mathbf{I}_N$ is the $N\times N$ identity matrix. Under unforced conditions, with $f=0$, the system is dissipative, so that

$$\dot{H} = -\tfrac{1}{M}\mathbf{p}^T\mathbf{R}\mathbf{p} \leq 0\,. \tag{21}$$

$H(t)$ is the semi-discrete counterpart of the total plate energy, as defined in (9a). When $\sigma_0=\sigma_1=0$, the system is lossless.

$V'$, as defined in (18), is non-negative. Scalar auxiliary variable methods follow from the definition of a new variable $\psi$, as

$$V' = \tfrac{1}{2}\psi^2\,. \tag{22}$$

Notice here that only the contribution $V'$ to the energy due to nonlinear effects has been quadratised here—other possibilities are avalable [12]. System (16) can then be rewritten, using this definition, as well as the quadratic dependence of $H$ on $\mathbf{p}$, as

$$\dot{\mathbf{w}} = \tfrac{1}{M}\mathbf{p} \qquad \dot{\mathbf{p}} = -\mathbf{K}_0\mathbf{w} - \psi\mathbf{g} - \mathbf{R}\mathbf{p} + \mathbf{j}f\,, \tag{23}$$

where

$$\mathbf{g} \triangleq \nabla_{\mathbf{w}}\psi\,. \tag{24}$$

Furthermore, using the chain rule,

$$\dot{\psi} = (\nabla_{\mathbf{w}}\psi)^T\dot{\mathbf{w}} = \mathbf{g}^T\dot{\mathbf{w}}\,. \tag{25}$$

Given that $H$ is quadratic in $\mathbf{p}$, a second order form of (23) follows immediately as

$$M\ddot{\mathbf{w}} = -\mathbf{K}_0\mathbf{w} - \psi\mathbf{g} - M\mathbf{R}\dot{\mathbf{w}} + \mathbf{j}f\,. \tag{26}$$

This equation, alongside (25), describing the time evolution of the scalar auxiliary variable $\psi$, and the nonlinear relationship (14b), forms a complete system describing the vibration of the plate.

## 3.4. Fully Discrete Form

A discrete update preserving an energy balance may be arrived at directly, generalising results in [12]. First, beginning from system (23) and (25), define the time series $\mathbf{w}^n$ and $\mathbf{g}^n$, representing approximations to $\mathbf{w}(t)$ and $\mathbf{g}(t)$ at times $t=nk$, for a given time step $k$ in s, and for integer $n$, and $\mathbf{p}^{n+1/2}$, $\psi^{n+1/2}$, interleaved approximations to $\mathbf{p}(t)$ and $\psi(t)$ at times $t=(n+1/2)k$. Consider the following time-interleaved scheme, resulting from basic centered differences and averaging of (23) and (25):

$$\begin{aligned}
\mathbf{w}^{n+1} &= \mathbf{w}^n + \tfrac{k}{M}\mathbf{p}^{n+\frac{1}{2}} && (27a)\\
\mathbf{p}^{n+\frac{1}{2}} &= \mathbf{p}^{n-\frac{1}{2}} - k\mathbf{K}_0\mathbf{w}^n - \frac{k}{2}\left(\psi^{n+\frac{1}{2}} + \psi^{n-\frac{1}{2}}\right)\mathbf{g}^n && (27b)\\
&\quad -\frac{k}{2}\mathbf{R}_0\left(\mathbf{p}^{n+\frac{1}{2}} + \mathbf{p}^{n-\frac{1}{2}}\right) - k\mathbf{R}_1\mathbf{p}^{n-\frac{1}{2}} + k\mathbf{j}f^n &&\\
\psi^{n+\frac{1}{2}} &= \psi^{n-\frac{1}{2}} + \tfrac{1}{2}\left(\mathbf{g}^n\right)^T\left(\mathbf{w}^{n+1} - \mathbf{w}^{n-1}\right)\,. && (27c)
\end{aligned}$$

Notice that in (27b), the linear and nonlinear parts of the plate dynamics have been approximated using different integration rules; and, in order to arrive at an explicit update, as will be seen shortly, the loss terms have been approximated separately, according to the decomposition in (20). $f^n$ is an approximation to $f(t)$ at $t=nk$; the calculation of $\mathbf{g}^n$ will be returned to in Section 3.6.

The updates (27a) and (27b) can be consolidated into a single two-step update in $\mathbf{w}^n$

$$\mathbf{A}^n\mathbf{w}^{n+1} = \mathbf{U}\mathbf{w}^n - \mathbf{C}^n\mathbf{w}^{n-1} + \tfrac{k^2}{M}\left(f^n\mathbf{j} - \psi^{n-\frac{1}{2}}\mathbf{g}^n\right) \tag{28}$$

and depends upon the three matrices $\mathbf{A}^n$, $\mathbf{U}$ and $\mathbf{C}^n$ defined by

$$\begin{aligned}
\mathbf{A}^n &= \mathbf{I}_N + \tfrac{k}{2}\mathbf{R}_0 + \tfrac{k^2}{4M}\mathbf{g}^n\left(\mathbf{g}^n\right)^T && (29a)\\
\mathbf{U} &= 2\mathbf{I}_N - \tfrac{k^2}{M}\mathbf{K}_0 - k\mathbf{R}_1 && (29b)\\
\mathbf{C}^n &= \mathbf{I}_N - \tfrac{k}{2}\mathbf{R}_0 - k\mathbf{R}_1 - \tfrac{k^2}{4M}\mathbf{g}^n\left(\mathbf{g}^n\right)^T\,. && (29c)
\end{aligned}$$

This update is apparently implicit, requiring the inversion of $\mathbf{A}^n$ at each time step. However, $\mathbf{A}^n$ is of the form of a multiple of the identity plus a rank-one perturbation, or $\mathbf{A}^n = d\mathbf{I}_N + \boldsymbol{a}^n(\boldsymbol{a}^n)^T$, where $d = 1+k\sigma_0$ and $\boldsymbol{a}^n = \frac{k}{2\sqrt{M}}\mathbf{g}^n$. A closed-form inverse is available through the Sherman-Morrison formula [24] as

$$\left(\mathbf{A}^n\right)^{-1} = d^{-1}\left(\mathbf{I}_N - \frac{\boldsymbol{a}^n(\boldsymbol{a}^n)^T}{d+(\boldsymbol{a}^n)^T\boldsymbol{a}^n}\right)\,. \tag{30}$$

Thus the linear system solution required in (28) can be performed in $O(N)$ operations, and the update can be viewed as effectively explicit.

## 3.5. Discrete Energy Balance and Stability Condition

In the lossless and source-free case, the scheme (27) is lossless to machine precision, as demonstrated recently [12]. When loss and sources are present, an energy balance of the following form holds:

$$\frac{1}{k}\left(\mathfrak{h}^{n+\frac{1}{2}} - \mathfrak{h}^{n-\frac{1}{2}}\right) = -\mathfrak{q}^n + \mathfrak{p}^n\,, \tag{31}$$

where

$$\begin{aligned}
\mathfrak{h}^{n+\frac{1}{2}} &= \tfrac{1}{2M}|\mathbf{p}^{n+\frac{1}{2}}|^2 + \tfrac{1}{2}\mathbf{w}^{n+1}\mathbf{K}_0\mathbf{w}^n + \tfrac{1}{2}\left(\psi^{n+\frac{1}{2}}\right)^2 && (32a)\\
&\quad -\tfrac{k}{4M}(\mathbf{p}^{n+\frac{1}{2}})^T\mathbf{R}_1\mathbf{p}^{n+\frac{1}{2}} &&\\
\mathfrak{q}^n &= \tfrac{1}{4M}(\mathbf{p}^{n+\frac{1}{2}}+\mathbf{p}^{n-\frac{1}{2}})^T(\mathbf{R}_0+k\mathbf{R}_1)(\mathbf{p}^{n+\frac{1}{2}}+\mathbf{p}^{n-\frac{1}{2}}) && (32b)\\
\mathfrak{p}^n &= \tfrac{1}{2M}(\mathbf{p}^{n+\frac{1}{2}}+\mathbf{p}^{n-\frac{1}{2}})^T\mathbf{j}f^n\,. && (32c)
\end{aligned}$$

This energy balance mirrors that of the continuous system, from (9); the major difference is that the expression $\mathfrak{h}^{n+1/2}$ for the stored energy is only non-negative under the condition

$$h \geq 2\sqrt{k}\sqrt{\sigma_1 + \sqrt{\sigma_1^2 + Q/\rho\xi}}\,, \tag{33}$$

which is the numerical stability condition for scheme (28). This is the same condition that follows from an analysis of numerical stability for the plate under linear conditions (i.e., the Kirchhoff plate) [19]. In practice, $h$ will be set as close to possible above this lower bound. Another difference, with respect to the energy for the continuous system, from (9a), is the appearance of additional stored energy above due to the loss term—this is the result of using a non-centered (backward) difference for the frequency-dependent loss term, allowing an explicit update for the plate.

## 3.6. Linear System Solution: The Biharmonic Operator

The scheme (28), as written, appears to be fully explicit, once Sherman-Morrison inversion is employed in order to solve the linear system involving $\mathbf{A}^n$. A hidden aspect here, though, is the determination of $\mathbf{g}^n$, approximating $\mathbf{g}(t)$ as defined in (24) at time $t = nk$. Regardless of the form of this approximation, the scheme (28) will be dissipative under zero-input conditions—note that $\mathbf{g}^n$ does not appear explicitly in the energetic expressions in (32). It must, however, be chosen to be consistent with the definition of $\mathbf{g}(t)$ in order to lead to a convergent algorithm. Note, from the definition of $\mathbf{g}$, that one may furthermore write, using $V' = \frac{1}{2}\psi^2$,

$$\mathbf{g} = \nabla_{\mathbf{w}}\psi = \frac{1}{\sqrt{2V'}}\nabla_{\mathbf{w}}V' \,. \tag{34}$$

From the definition of $V'$ in terms of $\mathbf{\Phi}$, from (18), and also the definition of $\Phi$ in terms of $\mathbf{w}$, from (14b), one may ultimately arrive at the following form [12] for $\mathbf{g}^n$:

$$\mathbf{g}^n = -\frac{h^2}{\sqrt{2(V')^n}}\ell(\mathbf{w}^n, \mathbf{\Phi}^n)\,, \tag{35}$$

where, in discrete time,

$$(V')^n = \frac{1}{2}(\mathbf{\Phi}^n)^T\mathbf{K}'\mathbf{\Phi}^n \quad \text{and} \quad \mathbf{D}_{\Delta\Delta}\mathbf{\Phi}^n = -\frac{EH}{2}\ell(\mathbf{w}^n, \mathbf{w}^n)\,, \tag{36}$$

and where the bilinear operator $\ell$ is as defined in (15).

Most of the operations above required in order to form $\mathbf{g}^n$ are simple. The exception is the linear system involving the biharmonic operator $\mathbf{D}_{\Delta\Delta}$ required in (36) in order to determine $\mathbf{\Phi}^n$ from $\mathbf{w}^n$. This is the remaining computational bottleneck, and is inherent to all numerical solutions to the Föppl-von Kármán equations. Using the fact that, under simply supported conditions, the biharmonic may be separated into a product of two Laplacians as $\mathbf{D}_{\Delta\Delta} = \mathbf{D}_\Delta\mathbf{D}_\Delta$, the linear system to solved at each time step is:

$$\mathbf{D}_\Delta\mathbf{D}_\Delta\mathbf{y} = \mathbf{c} \tag{37}$$

for a known $N \times 1$ vector $\mathbf{c}$, yielding an $N \times 1$ vector $\mathbf{y}$.

Standard linear system solvers (such as, e.g., those relying on Cholesky or LU factorisation) are far out of real time for reasonable plate sizes. See Section 5.1. Here, a very recently developed solver [25] that exploits the structure of $\mathbf{D}_\Delta$ is employed, leading to an efficient variant of the Thomas algorithm [26]. It is closely related to the method proposed by Buzbee [27]. To this end, note that the scaled $N \times N$ Laplacian operator $\tilde{\mathbf{D}}_\Delta = h^2\mathbf{D}_\Delta$ may be written explicitly as the Toeplitz-block-Toeplitz form:

$$\tilde{\mathbf{D}}_\Delta = \begin{bmatrix} \mathbf{T} & \mathbf{I} & & & \bullet \\ \mathbf{I} & \mathbf{T} & \mathbf{I} & & \\ & \ddots & \ddots & \ddots & \\ & & \mathbf{I} & \mathbf{T} & \mathbf{I} \\ \bullet & & & \mathbf{I} & \mathbf{T} \end{bmatrix} \quad \mathbf{T} = \begin{bmatrix} -4 & 1 & & & \bullet \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 1 \\ \bullet & & & 1 & -4 \end{bmatrix} . \tag{38}$$

Here, block sizes are $(N_y - 1) \times (N_y - 1)$. $\mathbf{I}$ is an identity matrix, and $\mathbf{T}$ a Toeplitz matrix as given above. Zero entries are indicated by $\bullet$, and $\tilde{\mathbf{D}}_\Delta$ is extremely sparse. The matrix $\mathbf{T}$ has the closed form eigendecomposition $\mathbf{T} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}$ where

$$[\mathbf{S}]_{\beta\gamma} = \sqrt{\frac{2}{N_y}}\sin(\beta\gamma\pi/N_y) \quad [\mathbf{\Lambda}]_{\beta\beta} = 2\cos(\beta\pi/N_y) - 4 \tag{39}$$

for $1 \leq \beta, \gamma \leq N_y - 1$. Note that $\mathbf{S} = \mathbf{S}^T = \mathbf{S}^{-1}$ is an orthogonal $(N_y - 1) \times (N_y - 1)$ matrix. $\mathbf{\Lambda}$ is diagonal, and contains the eigenvalues of $\tilde{\mathbf{D}}_\Delta$.

Now, form the $N \times N$ matrix $\mathbf{Q}$ as a Kronecker product $\mathbf{Q} = \mathbf{I}_{N_x - 1} \otimes \mathbf{S}$. Using the orthogonality of $\mathbf{S}$, one has

$$\tilde{\mathbf{D}}_\Delta = \mathbf{Q}\mathbf{\Xi}\mathbf{Q} \qquad \mathbf{\Xi} = \begin{bmatrix} \mathbf{\Lambda} & \mathbf{I} & & & \bullet \\ \mathbf{I} & \mathbf{\Lambda} & \mathbf{I} & & \\ & \ddots & \ddots & \ddots & \\ & & \mathbf{I} & \mathbf{\Lambda} & \mathbf{I} \\ \bullet & & & \mathbf{I} & \mathbf{\Lambda} \end{bmatrix} . \tag{40}$$

Given that $\mathbf{Q}$ inherits orthogonality from $\mathbf{S}$, one may then write the solution to (37) as

$$\mathbf{y} = h^4\mathbf{Q}\mathbf{\Xi}^{-1}\mathbf{\Xi}^{-1}\mathbf{Q}\mathbf{c}\,. \tag{41}$$

Note here that $\mathbf{Q}$ is sparse, with $N_x - 1$ blocks of size $(N_y - 1) \times (N_y - 1)$. Also, $\mathbf{\Xi}$ is block tridiagonal, with diagonal blocks, and thus the Thomas algorithm may be applied directly (twice, here).

Other approaches to Toeplitz-block-Toeplitz linear system solution are available, and were tested during the course of this work. These include extensions of the Levinson-Durbin algorithm due to Wax and Kailath [28]. One may also note that the matrix $\mathbf{S}$ corresponds to a discrete sine transform (DST); fft-based methods were tried, but were not efficient, due to the small size of $N_y$, and exhibited great variation depending on the factorisation of $N_y$. The method presented above performed best in all tests.

## 3.7. Output and Interpolation

Moving outputs are assumed drawn at locations $(\mathbf{r}_\mathrm{o}^{(p)})^n$ sampled from the elliptical trajectories $\mathbf{r}_\mathrm{o}^{(p)}(t)$, $p = 1, \ldots, P$ as defined in (7), and at times $t = nk$. Interpolation is a necessity in this setting, in order to avoid numerical artefacts (“zipper noise").

Suppose, at a given time instant, the coordinates of one of the trajectories takes on the value $\boldsymbol{\eta} = (\mathbf{r}_\mathrm{o}^{(p)})^n$. One may write

$$\boldsymbol{\eta} = h(l_\mathrm{o}, m_\mathrm{o}) + h(\zeta_x, \zeta_y) \tag{42}$$

uniquely for integer grid indeces $(l_\mathrm{o}, m_\mathrm{o})$ and fractional addresses $(\zeta_x, \zeta_y)$, where $0 \leq \zeta_x, \zeta_y < 1$. Assuming an interpolation width of $2J$ points, one may form an interpolant $w_\mathrm{o}$ from the two-dimensional grid function $w_{l,m}$ as:

$$w_\mathrm{o} = \sum_{\nu_x=-J+1}^{J}\sum_{\nu_y=-J+1}^{J} b_x^{\nu_x}(\zeta_x)b_y^{\nu_y}(\zeta_y)w_{l_\mathrm{o}+\nu_x, m_\mathrm{o}+\nu_x}\,. \tag{43}$$

Here, the interpolant is assumed separable, so that $b_x^{\nu_x}(\zeta_x)$ and $b_y^{\nu_y}(\zeta_y)$, $-J+1 \leq \nu_x, \nu_y \leq J$ are one-dimensional interpolation coefficients. In this work, approximations are assumed to be of Lagrange type, with $J = 2$.

When the grid function $w_{l,m}$ is reconstituted as an $N \times 1$ column vector $\mathbf{w}$, then this linear operation may be expressed as an inner product

$$w_\mathrm{o} = \mathbf{b}^T\mathbf{w}\,, \tag{44}$$

where the $N \times 1$ vector $\mathbf{b}$ incorporates the interpolation coefficients $b_x$ and $b_y$ The extension to the case of $P$ time-varying output trajectories is straightforward, and may be represented as

$$\mathbf{w}_\mathrm{o}^n = (\mathbf{B}^n)^T\mathbf{w}^n\,. \tag{45}$$

Here, $\mathbf{w}_\mathrm{o}^n$ is a $P \times 1$ column vector of output signals, and $\mathbf{B}^n = [\mathbf{b}^{n,(1)} \ldots \mathbf{b}^{n,(P)}]$ is an $N \times P$ matrix of interpolation coefficients.

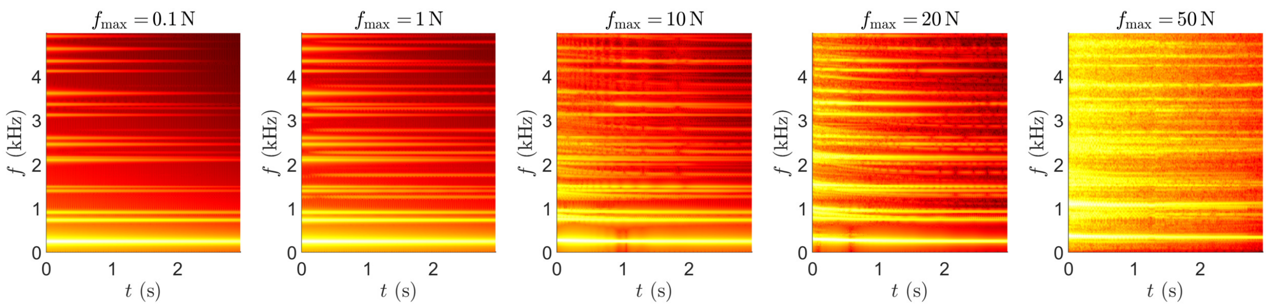

Figure 3: *Spectrograms of output from a small plate model, at different excitation amplitudes, as indicated.*

## 4. IMPLEMENTATION

### 4.1. Parameter Sets

Complete parameter sets for the gong, assuming a struck excitation, and $P$ outputs over an elliptical trajectory, are as follows:

$$\mathcal{O}_{\text{plate}} = \{E, \rho, \xi, \nu, L_x, L_y, \sigma_0, \sigma_1\} \tag{46a}$$
$$\mathcal{O}_{\text{I}} = \{t_0, T, f_{\max}\} \tag{46b}$$
$$\mathcal{O}_{\text{O}} = \{R, f_o^{(1)}, \ldots, f_o^{(P)}, \phi_o^{(1)}, \ldots, \phi_o^{(P)}\}\,. \tag{46c}$$

The sample rate $F_s$ must also be supplied. For the given system, this set of parameters is redundant, in terms of the space of possible sound outputs. One approach to reducing the number of defining parameters in $\mathcal{O}_{\text{plate}}$ is to non-dimensionalise the system (1) with respect to $w$ and $\Phi$, and to spatially scale the domain to unit area. Equally, and perhaps more intuitively, one could fix the plate material and thickness, leaving only the plate dimensions variable[2]. Furthermore, it is useful to introduce the equivalent parameters

$$A = L_x L_y \qquad\qquad \alpha = L_y/L_x \tag{47}$$

which are the plate surface area in m$^2$ and dimensionless aspect ratio respectively. $A$ is useful, as it will scale directly with computational cost, independently of $\alpha$. Furthermore, it is more intuitive to set the loss parameters in terms of perceptually-relevant decay times $T_{60,0}$ and $T_{60,c}$ at 0 Hz and $f_c$ Hz, respectively as

$$\sigma_0 = \frac{6\ln(10)}{T_{60,0}} \qquad \sigma_1 = \frac{6\ln(10)\sqrt{Q/\rho\xi}}{2\pi f_c}\left(\frac{1}{T_{60,c}} - \frac{1}{T_{60,0}}\right), \tag{48}$$

with $T_{60,0} \geq T_{60,c}$. A reduced parameter set $\mathcal{O}'_{\text{plate}}$ results:

$$\mathcal{O}'_{\text{plate}} = \{A, \alpha, T_{60,0}, T_{60,c}\}\,. \tag{49}$$

For the reduced set, $A$ and $\alpha$ must remain fixed over the course of a simulation. It is possible to vary $T_{60,0}$ and $T_{60,c}$, though in this case one must back off slightly from the stability condition given in (33) to accommodate the range of such variations.

### 4.2. Real-time Implementation

Prototyping was carried out in Matlab, using a sparse vector/matrix representation following directly from the form of the scheme in (28). Such a representation is optimal in terms of performance as well as readability and debugging in Matlab. (The additional biharmonic linear system solution in (36) was carried out using a generic Cholesky factorisation computed outside the runtime loop.)

In order to achieve real-time performance as in, e.g., an audio plug-in format, highly-optimised single-threaded C++ code is required. Previous testing has shown that a direct “matrix-unrolled" approach is up to $10\times$ faster than using a sparse matrix representation directly in C++ [29]. There are two main reasons why unrolling the sparse matrix form and applying stencil operations directly is more efficient. First, the vector/matrix representation, while sparse, is highly redundant with many repeated values, and can be reduced to a very small number of stencil coefficients. Second, in unrolled form the compiler is much more likely to be able to vectorize the code using a suitable optimisation level, and even if not it is relatively simple to manually apply vector intrinsics to the update. Sparse matrix data structures are more complicated in this respect due to the irregular data patterns used in typical reduced memory formats such as CSR (Compressed Spare Row).

For the gong algorithm described here, there are 12 different array operations that are required at each time-step, not including the core element of the linear system solution, as described in Section 3.6. By unrolling the sparse matrix representation and using -O3 optimisation level in Clang, the compiler was able to fully vectorize each array operation with AVX vectors without having to write any manual intrinsics at all. The solver element, however, did require manual application of intrinsics in order to achieve maximum efficiency.

## 5. NUMERICAL RESULTS

### 5.1. Timings

For numerical tests, three machines were used: (1) *LinuxLap*: a Linux laptop with an Intel 12th Gen quad-core i7-1260P CPU; (2) *WinPC*: a Windows desktop with an AMD Ryzen 7 8-core 5800X CPU; and (3) *MBA*: MacBook Air with an Intel 10th Gen quad-core i5 CPU; All tests were written in C++ and compiled with -O3 and -mavx2 flags.

Computation times for the complete plate simulation algorithm, for various choices of plate area and aspect ratio on different machines are shown in Table 1. These indicate that computation time tracks the total surface area $A$ reasonably closely, regardless of the aspect ratio, which is as expected. They indicate faster than real time performance for these plate areas, which are small, but definitely within the realm of musical gongs.

Table 2 shows the comparison between computation times for the biharmonic solver presented in Section 3.6 and those for heavily-

[2]In this article, the material is taken to be steel, with $E = 2\times10^{11}$ Pa, $\rho = 7850$ kg m$^{-3}$, and $\nu = 0.3$, and the plate thickness is $\xi = 0.5$ mm.

| $A$ (m$^2$) | $\alpha$ | $N_x$ | $N_y$ | LinuxLap | WinPC | MBA |
|---|---|---|---|---|---|---|
| 0.06 | 1.24 | 25 | 31 | 0.484 | 0.635 | 0.908 |
| 0.06 | 0.80 | 31 | 25 | 0.569 | 0.649 | 0.993 |
| 0.05 | 1.38 | 21 | 29 | 0.365 | 0.441 | 0.636 |
| 0.05 | 0.72 | 29 | 21 | 0.393 | 0.510 | 0.695 |
| 0.05 | 2.06 | 17 | 35 | 0.278 | 0.360 | 0.761 |
| 0.05 | 1.00 | 25 | 25 | 0.393 | 0.476 | 0.815 |
| 0.05 | 3.46 | 13 | 45 | 0.280 | 0.355 | 0.664 |
| 0.04 | 1.32 | 19 | 25 | 0.295 | 0.397 | 0.562 |
| 0.04 | 0.76 | 25 | 19 | 0.296 | 0.421 | 0.653 |
| 0.03 | 1.24 | 17 | 21 | 0.150 | 0.183 | 0.264 |
| 0.03 | 2.08 | 13 | 27 | 0.150 | 0.190 | 0.287 |

Table 1: Timings, in s, to compute 1 s output for plates of different areas $A$ and aspect ratios $\alpha$. Grid sizes $N_x$ and $N_y$ are as indicated.

| | $14 \times 14$ | $16 \times 20$ | $23 \times 17$ | $25 \times 25$ |
|---|---|---|---|---|
| Our solver | **0.054** | **0.135** | **0.118** | **0.279** |
| LU | 0.375 | 0.652 | 0.865 | 1.735 |
| Cholesky | 0.254 | 0.476 | 0.531 | 0.978 |

Table 2: Comparison between computation times, in s, for the biharmonic solver presented in Section 3.6 and alternative solvers from Eigen on WinPC, for typical grid sizes $N_x \times N_y$.

used generic linear system solvers like based on LU and Cholesky decompositions. Here we have used realisations of these solvers from Eigen[30], a well-known high-level C++ library for linear algebra.

Table 3 shows the percentage split between the biharmonic solver element of the timeloop code and the remaining sections. The solver is by far the most significant element, taking up to 76% of the computation time at each time-step.

### 5.2. Sound Output

It is useful to examine the effect of the plate nonlinearity through spectrograms of sound output. The sample rate is chosen as $F_s = 44.1$ kHz, and the material parameters and thickness are fixed as in the footnote on page 6. Reduced plate parameters are chosen as $\alpha = 1.4$, $T_{60,0} = 20$ s and $T_{60,c} = 10$ s, with $f_c = 1$ kHz. The excitation is of the form of (5), with $t_0 = 0$ s. Spectrograms are calculated using a window size of 2048 points, with a hop size of 128 points and Hann windowing applied.

Consider first a very small plate with $A = 0.01$ m$^2$, and the effect of increased excitation amplitude $f_{\max}$, where all other parameters are held constant. Output is drawn at the fixed location $\mathbf{r}_o = (L_x/5, 0)$. In this case, the excitation duration is $T = 2$ ms. See Figure 3. Under low-amplitude excitation amplitude, the linear behaviour of the plate is recovered, and distinct constant

| Plate size | Biharmonic Solver | Remaining |
|---|---|---|
| $19 \times 25$ $(0.04, 1.2)$ | $72.3\%$ | $27.7\%$ |
| $25 \times 31$ $(0.06, 1.2)$ | $75.3\%$ | $24.7\%$ |
| $19 \times 35$ $(0.05, 1.8)$ | $72.4\%$ | $27.6\%$ |
| $29 \times 21$ $(0.05, 0.7)$ | $76.2\%$ | $23.8\%$ |

Table 3: Comparison of computation time for the solver vs remaining elements of the optimised C++ code at each time-step.

modal frequencies are observed. At higher amplitudes, effects of pitch glides are observed—these glides are not uniform across all partials, however, as they would be if a simpler model of nonlinear plate vibration (e.g. that of Berger [15]) were used. At very high amplitudes, the partials themselves are replaced by wideband noise—the crash.

As a further illustration, consider now the case of a larger plate, with $A = 0.16$ m$^2$, again under increasing excitation amplitude, and now with excitation duration $T = 4$ ms. See Figure 4. In this case, the characteristic “swell" of a gong-like instrument may be observed—a slow migration of energy to the high frequency range over the first several hundred milliseconds.

As a final example, consider a comparison between spectrograms of sound output for a small plate with $A = 0.01$ m$^2$, and with an excitation amplitude $f_{\max} = 20$ N and duration $T = 4$ ms, in the presence of time-varying monophonic output, drawn from an ellipse with $R = 0.4$, and with a scan frequency of 1 Hz. See Figure 5. Easily visible are complex modulations of the individual frequency components, characteristic of that which occurs in an instrument that may be free to exhibit rigid-body oscillation relative to the listener.

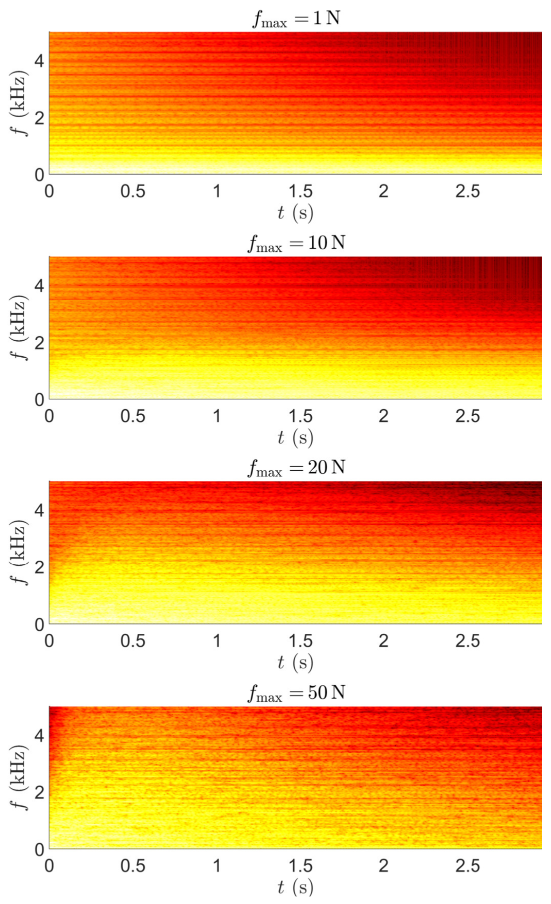

Figure 4: *Spectrograms of output from a large plate model, at different excitation amplitudes, as indicated.*

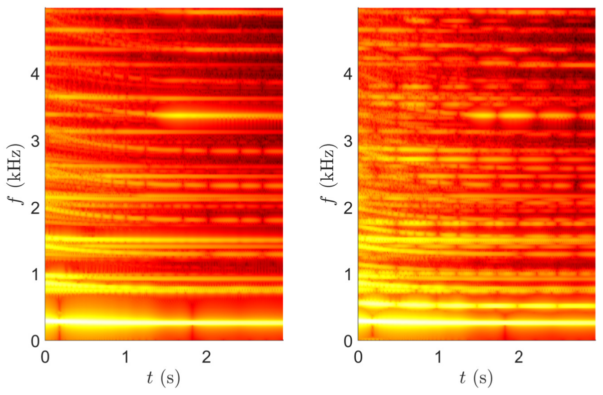

Figure 5: *Spectrograms of output from a small plate model, with a static output location (left), and a time-varying output location (right).*

## 6. CONCLUDING REMARKS

In this paper, it has been shown that it is possible to implement computationally-intensive physical models in real time—in this case the Föppl-von Kármán model of high amplitude plate vibration that is necessary in order to emulate gongs. Due to the complexity of the system, achieving real-time performance requires optimisation at multiple levels. First: the design of a numerically stable explicit integrator, as presented here, has been the key to breaking the real-time barrier. The computational advantage here hinges on the exploitation of matrix structure (in this case, rank-1 perturbation of the identity—a general property of SAV designs for Hamiltonian or near-Hamiltonian systems [12]) But even when this bottleneck has been removed, there remains the problem of linear system solution (of the biharmonic operator) in the run-time loop. Fast solution has been approached by exploiting a different type of matrix structure (block Toeplitz in this case). This type of acceleration is much more targeted at the particular case of the Föppl-von Kármán system. Further acceleration depends on the use of low-level parallelisation tools. The larger lesson here is that for physical modeling synthesis from any reasonably complex system, a silver bullet is likely not available—rather, in order to achieve good performance, great attention must be paid to the specifics of the system at hand.

Many simplifications to more realistic models of percussion instruments have been made in order to arrive at a real-time implementation. Among these are: a) the restriction to a rectangular geometry with simply-supported conditions; b) the assumption of a flat plate rather than a curved shell, which is more usual; c) the assumption of a uniform thickness; and d) the consolidation of effects of loss due to various mechanisms (radiation, viscothermal) to a basic two-parameter loss model. Including any of these effects would have no impact on the explicit integration method, provided the system may still be written in terms of the extension of a Hamiltonian system. On the other hand, the block-Toeplitz solver is highly dependent on restrictions a) to c). Coping with restriction d) would necessarily increase the temporal order of the scheme as a whole, leading to a larger footprint in terms of both memory usage (not a major concern here) as well as computational cost.

## 7. REFERENCES

[1] N Firasta, M. Buxton, P. Jinbo, K. Nasri, and S. Kuo, “Intel AVX: New frontiers in performance improvements and energy efficiency,” *Intel White Paper*, 2008.

[2] C. Touzé, O. Thomas, and A. Chaigne, “Asymmetric non-linear forced vibrations of free-edge circular plates, part I: theory,” *J. Sound Vib.*, vol. 258, no. 4, pp. 649–676, 2002.

[3] O. Thomas, C. Touzé, and A. Chaigne, “Non-linear vibrations of free-edge thin spherical shells: Modal interaction rules and 1:1:2 internal resonance,” *Int. J. Solids Structures*, vol. 42, pp. 3339–3373, 2005.

[4] S. Bilbao, “Sound synthesis for nonlinear plates,” in *Proc. 8th Int. Conf. Digital Audio Effects*, Madrid, Spain, Sept. 2005, pp. 243–248.

[5] E. Hairer, C. Lubich, and G. Wanner, “Geometric numerical integration illustrated by the Störmer–Verlet method,” *Acta Numerica*, vol. 12, pp. 399–450, 2003.

[6] S. Bilbao, “A family of conservative finite difference schemes for the dynamical von Kármán plate equations,” *Num. Meth. Partial Diff. Eq.*, vol. 24, no. 1, pp. 193–216, 2008.

[7] X. Yang, J. Zhao, and Q. Wang, “Numerical approximations for the molecular beam epitaxial growth model based on the invariant energy quadratization method,” *J. Comp. Phys.*, vol. 333, pp. 104–127, 2017.

[8] J. Zhao, “A revisit of the energy quadratization method with a relaxation technique,” *Appl. Math. Lett.*, vol. 120, pp. 107331, 2021.

[9] Z. Liu and X. Li, “The exponential scalar auxiliary variable (e-sav) approach for phase field models and its explicit computing,” *SIAM J, Sci. Comp.*, vol. 42, no. 3, pp. B630–B655, 2020.

[10] M. Jiang, Z. Zhang, and J. Zhao, “Improving the accuracy and consistency of the scalar auxiliary variable (SAV) method with relaxation,” *J. Comp. Phys.*, vol. 456, pp. 110954, 2022.

[11] S. Bilbao and M. Ducceschi, “Fast explicit algorithms for Hamiltonian numerical integration,” in *Proc. Eur. Nonlinear Dynamics Conf.*, Lyon, France, July 2022.

[12] S. Bilbao, M. Ducceschi, and F. Zama, “Explicit exactly energy-conserving methods for Hamiltonian systems,” *J. Comp. Phys.*, vol. 427, pp. 111697, 2023.

[13] K. Graff, *Wave Motion in Elastic Solids*, Dover, New York, New York, 1975.

[14] R. Mindlin, “Influence of rotatory inertia and shear on flexural motions of isotropic elastic plates,” *J. Appl. Mech.*, vol. 18, pp. 31–38, 1951.

[15] H. Berger, “A new approach to the analysis of large deflections of plates,” *J. Appl. Math.*, vol. 22, pp. 465–472, 1955.

[16] A. Föppl, *Vorlesungen über technische Mechanik*, Druck und Verlag von B.G. Teubner, Leipzig, 1907.

[17] T. von Kármán, “Festigkeitsprobleme im maschinenbau,” *Encyklopädie der Mathematischen Wissenschaften*, vol. 4, no. 4, pp. 311–385, 1910.

[18] A. Nayfeh and D. Mook, *Nonlinear Oscillations*, John Wiley and Sons, New York, New York, 1979.

[19] S. Bilbao, *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics*, John Wiley and Sons, Chichester, UK, 2009.

[20] Q. Nguyen and C. Touzé, “Nonlinear vibrations of thin plates with variable thickness: Application to sound synthesis of cymbals,” *J. Acoust. Soc. Am.*, vol. 145, pp. 977–988, 2019.

[21] L. Rhaouti, A. Chaigne, and P. Joly, “Time-domain modeling and numerical simulation of a kettledrum,” *J. Acoust. Soc. Am.*, vol. 105, no. 6, pp. 3545–3562, 1999.

[22] B. Hosseini, N. Nigam, and J. Stockie, “On regularizations of the Dirac delta distribution,” *J. Comp. Phys.*, vol. 305, pp. 423–447, 2016.

[23] O. Thomas and S. Bilbao, “Geometrically nonlinear flexural vibrations of plates: In-plane boundary conditions and some symmetry properties,” *J. Sound Vib.*, vol. 315, no. 3, pp. 569–590, 2008.

[24] J. Sherman and W. Morrison, “Adjustment of an inverse matrix corresponding to a change in one element of a given matrix,” *Ann. Math. Stat.*, vol. 21, no. 1, pp. 124–127, 1950.

[25] Z. Wang and M. Puckette, “A fast algorithm for the inversion of the biharmonic in plate dynamics applications,” in *Proceedings of the 5th Stockholm Music Acoustic Conference*, Stockholm, Sweden, June 2023.

[26] L. Thomas, “Elliptic problems in linear differential equations over a network: Watson scientific computing laboratory,” Tech. Rep., Columbia Univ., 1949.

[27] B. Buzbee, G. Golub, and C. Nielson, “On direct methods for solving Poisson’s equations,” *SIAM J. Num. Anal.*, vol. 7, no. 4, pp. 627–656, 1970.

[28] M. Wax and T. Kailath, “Efficient inversion of Toeplitz-block Toeplitz matrix,” *IEEE Trans. Acoust. Speech Sig. Proces.*, vol. 31, no. 5, pp. 1218–1221, 1983.

[29] C. Webb and S. Bilbao, “On the limits of real-time physical modelling synthesis with a modular environment,” in *Proc. 18th Int. Conf. Digital Audio Effects*, Trondheim, Norway, Sept. 2015, pp. 65–72.

[30] G. Guennebaud and B. Jacob et al., “Eigen v3,” http://eigen.tuxfamily.org, 2010.