S. Willemsen, S. Bilbao, M. Ducceschi, and S. Serafin,“The Dynamic Grid: Time-Varying Parameters
for Musical Instrument Simulations Based on Finite-Difference Time-Domain Schemes”
J. Audio Eng. Soc., vol. 70, no. 9, pp. 650–660, (2022 September).
DOI: https://doi.org/10.17743/jaes.2022.0043

The Dynamic Grid: Time-Varying Parameters for Musical Instrument Simulations Based on Finite-Difference Time-Domain Schemes

SILVIN WILLEMSEN,[1] STEFAN BILBAO,[2] MICHELE DUCCESCHI,[3] AND STEFANIA SERAFIN[1]
(silvinwillemsen@gmail.com) (s.bilbao@ed.ac.uk) (michele.ducceschi@unibo.it) (sts@create.aau.dk)

[1]Multisensory Experience Lab, CREATE, Aalborg University Copenhagen, Denmark
[2]Acoustics and Audio Group, University of Edinburgh, United Kingdom
[3]Department of Industrial Engineering (DIN), University of Bologna, Italy

Several well-established approaches to physical modeling synthesis for musical instruments exist. Finite-difference time-domain methods are known for their generality and flexibility in terms of the systems one can model but are less flexible with regard to smooth parameter variations due to their reliance on a static grid. This paper presents the dynamic grid, a method to smoothly change grid configurations of finite-difference time-domain schemes based on sub-audio–rate time variation of parameters. This allows for extensions of the behavior of physical models beyond the physically possible, broadening the range of expressive possibilities for the musician. The method is applied to the 1D wave equation, the stiff string, and 2D systems, including the 2D wave equation and thin plate. Results show that the method does not introduce noticeable artefacts when changing between grid configurations for systems, including loss.

0 INTRODUCTION

The functioning of nearly any musical instrument can be subdivided into exciter and resonator components [1, 2]. Examples of resonators are the violin (strings and body) and brass instrument bore, which are excited by the bow and lips of the player, respectively. The resonator is often assumed to be linear (exceptions being nonlinear string vibration [3] or shock waves in bores [4]) and time-invariant, whereas the excitation is usually modeled as a lumped nonlinearity and can be controlled by the performer over time. However, real-world cases in which the defining parameters of the resonator are time-varying do exist. A notable example is the trombone, for which the length of the acoustic tube is changed during performance. Furthermore, membrane tension in timpani or “hourglass drums” is varied in performance. See [5, Sec. 12.4] for more examples.

Over the past few decades, much work has been done on emulating real-world musical instruments, specifically resonator components, through various physical modeling techniques. A detailed comparison and summary can be found in [6]. Finite-difference time-domain (FDTD) methods, although not the most efficient, are flexible and generalizable in terms of the systems one can model [7].

Although FDTD methods have been extensively used for sound synthesis purposes, relatively little work has been done on varying the defining parameters of the resonator during performance. Besides needing to handle the difficulties that arise when working with time-varying systems—both the difficulties in the underlying continuous equations and stability issues in numerical implementation—most physical resonators (excluding the exceptions mentioned before) are described by a fixed set of parameters. In other words, properties such as material density and geometry of the instrument are unchangeable in the real world and will thus remain this way in simulation.

In the authors’ view, one of the greatest benefits of musical instrument modeling is extending instrument functionality and design beyond that which is physically possible. Instrument properties that are normally fixed can be made time-varying to greatly extend the range of sound and expression available to the musician. Extensions to the time-varying case of physical modeling synthesis have been presented using different methodologies: modal synthesis

[8] in [9, 10], digital waveguides [11] in [12, 13], and mass-spring systems in [14]. An acoustic tube with time-varying length implemented using FDTD methods is presented in [15] and uses full-grid interpolation to update the system states whenever the length is changed.

This paper presents the dynamic grid, a method to allow for time-varying parameters in real-time simulations of musical instruments based on FDTD methods. The current work generalizes the method presented in [16], in which it is applied to the 1D wave equation, and extends it to more complex systems, such as the stiff string, and 2D systems, including the 2D wave equation and thin plate. Changes in parameter values are assumed to be sub-audio rate (control rate) such that they can be applied to commonly used FDTD schemes. The method appears in part in [5, Chapter 12] and has been used to model the trombone, including time-varying length in [17]. Here, grid points are added along the grid as opposed to [15], in which this only happens at the radiating end.

This paper is structured as follows: SEC. 1 presents the 1D wave equation as a starting point and introduces FDTD methods. SEC. 2 introduces the dynamic grid and its application to the 1D wave equation. SEC. 3 extends the method to the stiff string, after which SEC. 4 presents its application to 2D systems including the 2D wave equation and thin plate. SEC. 5 presents the analysis of the method and its results, and a discussion and concluding remarks appear in SECS. 6 and 7, respectively.

1 1D WAVE EQUATION

A useful starting point for illustrating the dynamic grid is the 1D wave equation, with state variable $q(x, t)$ defined over spatial coordinate $x \in [0, L]$ for some length L (in meters) and time $t \geq 0$ (in seconds):

$$\partial_t^2 q = c^2 \partial_x^2 q. \tag{1}$$

Here, c is the wave speed (in meters per second), and derivatives with respect to t and x are denoted by ∂_t and ∂_x, respectively. In Eq. (1), q can be interpreted as the transverse displacement of an ideal string or acoustic pressure in a cylindrical tube. A basic choice of boundary condition is the Dirichlet condition, which can be interpreted as a "fixed" termination for the ideal string, or an "open" condition or the acoustic tube. The Dirichlet boundary condition is defined as

$$q(0, t) = q(L, t) = 0. \tag{2}$$

1.1 Numerical Methods

To discretize Eq. (1) using FDTD methods, a spatio-temporal grid needs to be defined first. Time $t \geq 0$ can be discretized as $t = nk$ with temporal index $n = 0, 1, 2, \ldots$ and time-step $k = 1/f_\mathrm{s}$ (in seconds) with sample rate f_s (in Hertz). Space $x = [0, L]$ is subdivided into N equal intervals of length h (in m)—also called the grid spacing—according to $x = lh$ with spatial index $l \in \{0, \ldots, N\}$.

Using these definitions, the continuous state can be approximated as $q(x, t) \approxeq q_l^n$, in which q_l^n is a grid function that describes the state of the system over $N + 1$ grid points. Furthermore, continuous-time derivatives that appear in Eq. (1) are approximated as

$$\begin{aligned}\delta_{tt} q_l^n &= \frac{1}{k^2}\left(q_l^{n+1} - 2q_l^n + q_l^{n-1}\right) \approxeq \partial_t^2 q,\\ \delta_{xx} q_l^n &= \frac{1}{h^2}\left(q_{l+1}^n - 2q_l^n + q_{l-1}^n\right) \approxeq \partial_x^2 q.\end{aligned} \tag{3}$$

Eq. (1) can then be discretized to the following FDTD scheme:

$$\delta_{tt} q_l^n = c^2 \delta_{xx} q_l^n, \tag{4}$$

which, using Eq. (3), can be expanded to the following updated equation:

$$q_l^{n+1} = 2q_l^n - q_l^{n-1} + \lambda^2\left(q_{l+1}^n - 2q_l^n + q_{l-1}^n\right). \tag{5}$$

Here,

$$\lambda = \frac{ck}{h} \tag{6}$$

is referred to as the Courant number [18] and is related to stability and simulation quality, as will be described in SEC. 1.3.

From Eq. (5), the update at the domain endpoints ($l = 0$, and $l = N$) appears to require access to grid points outside the defined domain, i.e., q_{-1}^n and q_{N+1}^n. But, using the Dirichlet boundary conditions in Eq. (2), one has

$$q_0^n = q_N^n = 0. \tag{7}$$

In practice, the computational domain is reduced to $l = \{1, \ldots, N-1\}$.

1.2 Matrix Form

Both for compact implementation of FDTD schemes and for the application of the dynamic grid to more complex systems later on, it is useful to write an updated equation [such as Eq. (5)] in matrix form. The general matrix form of an updated equation is defined as

$$\mathbf{A}\mathbf{q}^{n+1} = \mathbf{B}\mathbf{q}^n + \mathbf{C}\mathbf{q}^{n-1}, \tag{8}$$

where the definitions of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ depend on the system at hand. Furthermore, $\mathbf{q}^n$ contains the state of the system at time index n.

Using Dirichlet boundary conditions, the state q_l^n in SEC. 1.1 can be represented as an $(N-1) \times 1$ column vector $\mathbf{q}^n = [q_1^n, \ldots, q_{N-1}^n]^T$, in which T denotes the transpose operation. Notice that the boundaries (q_0^n and q_N^n) are not included in the state vector because they are 0 at all times. With reference to Eq. (8), Eq. (4) can then be written in matrix form using the following definitions for the matrices:

$$\mathbf{A} = \mathbf{I}_{N-1}, \quad \mathbf{B} = 2\mathbf{I}_{N-1} + \lambda^2\mathbf{D}_{xx}, \quad \text{and} \quad \mathbf{C} = -\mathbf{I}_{N-1}, \tag{9}$$

where the $(N-1) \times (N-1)$ matrix

$$\mathbf{D}_{xx} = \begin{bmatrix} \ddots & \ddots & & & \mathbf{0} \\ \ddots & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & 1 & -2 & \ddots \\ \mathbf{0} & & & \ddots & \ddots \end{bmatrix}, \tag{10}$$

and $\mathbf{I}_{N-1}$ is the identity matrix of the same size.

1.3 Stability and Simulation Quality

In order to ensure stability for the scheme in Eq. (4), the Courant number in Eq. (6) needs to satisfy the Courant-Friedrichs-Lewy condition [18]:

$$\lambda \leq 1. \tag{11}$$

If $\lambda = 1$, Eq. (4) provides an approximation to Eq. (1) with the highest possible simulation quality for this scheme [7]. In the case that $\lambda < 1$, numerical error is introduced, which decreases the quality of the simulation shown by a decrease in bandwidth and dispersive effects. It is important to note that the smaller λ is, the lower the simulation quality.

Usually, Eq. (11) is rewritten in terms of the grid spacing such that $h \geq ck$, which is implemented as

$$h := ck, \quad N := \left\lfloor \frac{L}{h} \right\rfloor, \quad h := \frac{L}{N}, \quad \text{and} \quad \lambda := \frac{ck}{h}. \tag{12}$$

This ensures that λ is as close to 1 as possible while still satisfying Eq. (11). Here, $\lfloor \cdot \rfloor$ denotes the flooring operation that is necessary to ensure an integer number of intervals. Eq. (12) shows that h needs to be recalculated based on integer N, after which, λ is calculated from this. If L/h is not an integer, this means that $\lambda < 1$, yielding numerical dispersion (though usually small if λ is near 1).

2 THE DYNAMIC GRID

As stated in SEC. 0, the goal of this work is to introduce time-varying parameters into FDTD-based simulations. In audio applications, the sample rate f_s, and thus the time step k, are rarely varied [7] and therefore assumed fixed in this work. One can therefore observe from the 1D wave equation in Eq. (1) that the wave speed c and system length L are the only parameters that can be made time-varying; the other variables, such as h, λ, and N are derived from these physical parameters.

Parameter changes cause several computational issues to emerge. First, a change in the wave speed c will alter λ through Eq. (12), which causes issues regarding stability and simulation quality as detailed in SEC. 1.3. Secondly, and more importantly, changing c (or L) changes the number of intervals N according to Eq. (12) and thus the number of grid points describing the state of the system. Apart from *how* and *where* to add or remove grid points based on the now-dynamic wave speed, this needs to happen smoothly in order to prevent audible artefacts.

This work proposes a nonuniform grid with a *fractional* number of intervals $\mathcal{N} = L/h$ such that $N = \lfloor \mathcal{N} \rfloor$. This removes the need for the flooring operation in Eq. (12) (and therefore the recalculation of h), and allows $\lambda = 1$ at all times. Furthermore, $\mathcal{N}$ potentially allows for smooth transitions between grid configurations.

2.1 Proposed Method

In the following, the location of a grid point q_l at time index n will be written as $x^n_{q_l}$ (in meters from the left boundary). Moreover, the following time-varying parameters are indicated with a superscript n: L^n, c^n, h^n, λ^n, $\mathcal{N}^n$ and N^n. As a starting point, the original system q^n_l, with $l = \{0, \ldots, N^n\}$ is split into two subsystems: $v^n_{l_v}$ with $l_v = \{0, \ldots, M^n_v\}$ and $w^n_{l_w}$ with $l_w = \{0, \ldots, M^n_w\}$:

$$\begin{aligned} \delta_{tt} v^n_{l_v} &= (c^n)^2 \delta_{xx} v^n_{l_v}, \\ \delta_{tt} w^n_{l_w} &= (c^n)^2 \delta_{xx} w^n_{l_w}, \end{aligned} \tag{13}$$

which have $M^n_v + 1$ grid points and $M^n_w + 1$ grid points, respectively. (It is important to note that the superscripts n in M^n_v and M^n_w are unaffected by the δ_{tt} operator after expansion.) Here, $0 < M^n_v < N^n$ and $M^n_w = N^n - M^n_v$ which causes Eq. (13) to contain one more grid point than the original system in Eq. (4). Both systems are placed on the same domain x with their locations defined as

$$x^n_{v_{l_v}} = l_v h^n, \quad x^n_{w_{l_w}} = L^n - (M^n_w - l_w)h^n. \tag{14}$$

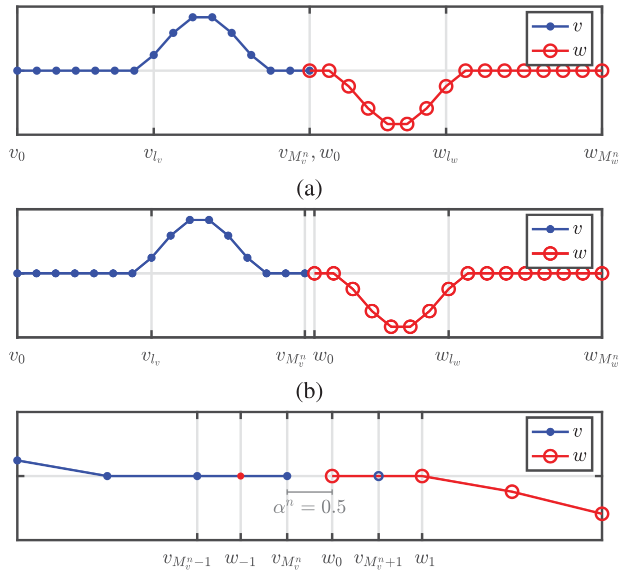
See Fig. 1(a). Notice that although the grid points are allowed to move, sub-audio–rate parameter changes are assumed such that $x^{n+1}_{v_{l_v}} \approx x^n_{v_{l_v}} \approx x^{n-1}_{v_{l_v}}$ and $x^{n+1}_{w_{l_w}} \approx x^n_{w_{l_w}} \approx x^{n-1}_{w_{l_w}}$ which allows δ_{tt} in Eq. (13) to still be applied to $v^n_{l_v}$ and $w^n_{l_w}$ as an approximation to ∂^2_t. To transfer the state of a grid point to its next location, zero-order interpolation (or rounding) is implicitly used.

With reference to Eq. (14), the grid locations $l_v = 0$ and $l_w = M^n_w$ are referred to as the *outer boundaries* and fixed to be at the limits of the system domain, i.e., $x^n_{v_0} = 0$ and $x^n_{w_{M^n_w}} = L^n, \forall n$. Furthermore, Dirichlet boundary conditions are imposed on the outer boundaries according to Eq. (7). The grid locations $l_v = M^n_v$ and $l_w = 0$ are referred to as the *inner boundaries*, at which Eq. (13) (after expansion) requires a definition for the *virtual grid points* $v^n_{M^n_v+1}$ and w^n_{-1}. If $\mathcal{N}^n = N^n$, i.e., $\mathcal{N}^n$ is an integer, the inner boundaries overlap, and the following condition must be satisfied:

$$v^n_{M^n_v} = w^n_0, \quad \text{if } x^n_{v_{M^n_v}} = x^n_{w_0}. \tag{15}$$

This acts as a continuity constraint at the inner boundaries. As shown in [16], setting the virtual grid points needed to calculate $v^n_{M^n_v}$ and w^n_0 according to

$$v^n_{M^n_v+1} = w^n_1 \quad \text{and} \quad w^n_{-1} = v^n_{M^n_v-1}, \tag{16}$$



(a)

(b)

(c)

Fig. 1. Illustration of the dynamic grid. System q_l^n in Eq. (4) is split into subsystems $v_{l_v}^n$ and $w_{l_w}^n$ [Eq. (13)]. The x-axis shows the location of the respective grid points with "x^n" omitted for brevity. (a) $\mathcal{N}^n = 30$. As $\mathcal{N}^n = N^n$, the inner boundaries overlap. (b) $\mathcal{N}^n = 30.5$. The wave speed c^n and thus h^n are decreased, $\mathcal{N}^n \neq N^n$, and the inner boundaries no longer overlap. (c) Zoomed-in view of (b). All grid points (including virtual grid points) used in Eq. (17) are shown. The distance between the inner boundaries is expressed using α^n in Eq. (19). (Image adapted from [16].)

causes Eq. (13) to exhibit identical behavior to the original system in Eq. (4).

Consider a decrease in wave speed c^n, which yields a decrease in h^n. This causes all grid points to move toward their respective outer boundary according to Eq. (14) [see Fig. 1(b)]. As the inner boundaries no longer overlap, i.e., $\mathcal{N}^n \neq N^n$, Eq. (16) can no longer be used, and alternative definitions for the virtual grid points need to be found. To this end, quadratic Lagrange interpolation can be used:

$$\begin{aligned} v_{M_v^n+1}^n &= \mathcal{I}^n v_{M_v^n}^n + w_0^n - \mathcal{I}^n w_1^n, \\ w_{-1}^n &= -\mathcal{I}^n v_{M_v^n-1}^n + v_{M_v^n}^n + \mathcal{I}^n w_0^n, \end{aligned} \tag{17}$$

where

$$\mathcal{I}^n = \frac{\alpha^n - 1}{\alpha^n + 1}, \tag{18}$$

and

$$\alpha^n = \mathcal{N}^n - N^n \tag{19}$$

is the fractional part of $\mathcal{N}^n$. (Notice that $0 \leq \alpha^n < 1$.) Also see Fig. 1(c). For results of experiments with other interpolators, see [5, Sec. 12.3].

2.2 Matrix Form

Similar to Sec. 1.2, one can define a general update equation of a system including the dynamic grid as

$$\boldsymbol{\mathcal{A}}^n \boldsymbol{u}^{n+1} = \boldsymbol{\mathcal{B}}^n \boldsymbol{u}^n + \boldsymbol{\mathcal{C}}^n \boldsymbol{u}^{n-1}, \tag{20}$$

where the definitions of $\boldsymbol{\mathcal{A}}^n$, $\boldsymbol{\mathcal{B}}^n$ and $\boldsymbol{\mathcal{C}}^n$ depend on the system at hand. Furthermore, $\mathcal{M}^n \times 1$ vector

$$\boldsymbol{u}^n = \left[(\mathbf{v}^n)^T, (\mathbf{w}^n)^T\right]^T, \tag{21}$$

with $\mathcal{M}^n = M_v^n + M_w^n$, includes the state of the subsystems in Eq. (13). Here, $\mathbf{v}^n = [v_1^n, v_2^n, \ldots, v_{M_v^n}^n]^T$ and $\mathbf{w}^n = [w_0^n, w_1^n, \ldots, w_{M_w^n-1}^n]^T$ which are $M_v^n \times 1$ and $M_w^n \times 1$, respectively. The 1D wave equation including the dynamic grid can then be written in matrix form as Eq. (20) with

$$\begin{aligned} \boldsymbol{\mathcal{A}}^n = \mathbf{I}_{\mathcal{M}^n}, \quad \boldsymbol{\mathcal{B}}^n &= 2\mathbf{I}_{\mathcal{M}^n} + (\lambda^n)^2 \boldsymbol{\mathcal{D}}_{xx}^n, \\ \text{and} \quad \boldsymbol{\mathcal{C}}^n &= -\mathbf{I}_{\mathcal{M}^n}, \end{aligned} \tag{22}$$

where $\lambda^n = c^n k / h^n$ (which can always be set to 1 in the case of the 1D wave equation, yielding optimal simulation quality at all times), and $\boldsymbol{\mathcal{D}}_{xx}^n$ is an adapted version of $\mathbf{D}_{xx}$ in Eq. (10) to include the quadratic interpolation presented in Eq. (17),

$$\boldsymbol{\mathcal{D}}_{xx}^n = \left[\begin{array}{ccc|ccc} \ddots & \ddots & & & & \mathbf{0} \\ \ddots & -2 & 1 & & & \\ & 1 & \mathcal{I}^n - 2 & 1 & -\mathcal{I}^n & \\ \hline & -\mathcal{I}^n & 1 & \mathcal{I}^n - 2 & 1 & \\ & & & 1 & -2 & \ddots \\ \mathbf{0} & & & & \ddots & \ddots \end{array}\right], \tag{23}$$

and is of size $\mathcal{M}^n \times \mathcal{M}^n$. In the extreme case that one of the systems has only one moving grid point (e.g., if $\mathbf{w}^n = [w_0^n]$ the lower-right quadrant in Eq. (23) will only have one entry (being $\mathcal{I}^n - 2$), and the lower-left and top-right quadrants will only have one row and one column, respectively.

2.3 Adding and Removing Grid Points

If c^n is decreased such that $N^n > N^{n-1}$, a grid point is added to the system. One can add points to either $\mathbf{v}$ or $\mathbf{w}$ or to both in an alternating fashion as in [16], but here, only changes in $\mathbf{v}$ are considered. A grid point can be added to $\mathbf{v}$ using the following operations:

$$\left.\begin{aligned} \mathbf{v}^n &:= [(\mathbf{v}^n)^T, I_3^n \mathbf{z}^n]^T, \\ \mathbf{v}^{n-1} &:= [(\mathbf{v}^{n-1})^T, I_3^n \mathbf{z}^{n-1}]^T, \end{aligned}\right. \quad \text{if } N^n > N^{n-1}, \tag{24}$$

where

$$\begin{aligned} \mathbf{z}^n &= [v_{M_v^{n-1}-1}^n, v_{M_v^{n-1}}^n, w_0^n, w_1^n]^T, \quad \text{and} \\ \mathbf{z}^{n-1} &= [v_{M_v^{n-1}-1}^{n-1}, v_{M_v^{n-1}}^{n-1}, w_0^{n-1}, w_1^{n-1}]^T. \end{aligned} \tag{25}$$

Notice that I_3^n is used for adding a grid point to both $\mathbf{v}^n$ and $\mathbf{v}^{n-1}$ and that M_v^{n-1} is used as an index for both $v_{l_v}^n$ and $v_{l_v}^{n-1}$. Furthermore, cubic Lagrangian interpolator

$$I_3^n = \left[-\frac{\alpha^n(\alpha^n+1)}{(\alpha^n+2)(\alpha^n+3)} \; \frac{2\alpha^n}{\alpha^n+2} \; \frac{2}{\alpha^n+2} \; - \frac{2\alpha^n}{(\alpha^n+3)(\alpha^n+2)}\right], \tag{26}$$

where α^n is as defined in Eq. (19). Notice that, because $\alpha^n \gtrsim 0$ the moment a grid point is added (because of sub-audio–rate parameter variations), $I_3^n \approx [0, 0, 1, 0]$ and the state of the added grid point is almost fully determined by the state of the inner boundary w_0^n.

Removing grid points happens through simple deletion. If c^n is increased such that $N^n < N^{n-1}$, a point is removed from $\mathbf{v}$ according to

$$\begin{aligned}\mathbf{v}^n &:= [v_1^n, \ldots, v^n_{M_v^{n-1}-1}]^T,\\ \mathbf{v}^{n-1} &:= [v_1^{n-1}, \ldots, v^{n-1}_{M_v^{n-1}-1}]^T,\end{aligned} \quad \text{if } N^n < N^{n-1}. \tag{27}$$

It must be noted that for lossless systems, removing grid points could cause audible artefacts. This will be discussed in SEC. 5.2. Furthermore, the limit on changing grid configurations is the addition or removal of one grid point per sample, but this limit needs to be much lower to retain the sub-audio–rate assumption mentioned before.

Notice that although a proof of convergence of the proposed method is left for future work, numerical experiments in SEC. 5.3 demonstrate convergence.

3 DAMPED STIFF STRING

Using the matrix in Eq. (23) as a starting point, one can extend the dynamic grid method to more complex systems. A commonly used 1D model is the damped stiff string (see, e.g., [19]), which is described by the following partial differential equation (PDE) [20]:

$$\partial_t^2 q = c^2 \partial_x^2 q - \kappa^2 \partial_x^4 q - 2\sigma_0 \partial_t q + 2\sigma_1 \partial_t \partial_x^2 q, \tag{28}$$

with stiffness coefficient κ (in m^2/s) and frequency-independent and frequency-dependent damping coefficients σ_0 (in s^{-1}) and σ_1 (in m^2/s).

Using

$$\begin{aligned}\delta_{t\cdot} q_l^n &= \frac{1}{2k}\left(q_l^{n+1} - q_l^{n-1}\right) \approxeq \partial_t q, \quad \text{and}\\ \delta_{t-} q_l^n &= \frac{1}{k}\left(q_l^n - q_l^{n-1}\right) \approxeq \partial_t q,\end{aligned} \tag{29}$$

Eq. (28) can be discretized using the following FDTD scheme:

$$\delta_{tt} q_l^n = c^2 \delta_{xx} q_l^n - \kappa^2 \delta_{xxxx} q_l^n - 2\sigma_0 \delta_{t\cdot} q_l^n + 2\sigma_1 \delta_{t-} \delta_{xx} q_l^n, \tag{30}$$

where $\delta_{xxxx} = \delta_{xx}\delta_{xx}$. The first-order temporal operators for the damping terms are chosen like this to keep the system explicit. Note that the stability condition is now $h \geq \sqrt{\frac{1}{2}\left(c^2k^2 + 4\sigma_1 k + \sqrt{(c^2k^2 + 4\sigma_1 k)^2 + 16\kappa^2 k^2}\right)}$ (in the time-invariant case). Eq. (30) can be written in matrix form using Eq. (8) with

$$\begin{aligned}\mathbf{A} &= (1 + \sigma_0 k)\mathbf{I}_{N-1},\\ \mathbf{B} &= 2\mathbf{I}_{N-1} + \lambda^2 \mathbf{D}_{xx} - \mu^2 \mathbf{D}_{xxxx} + \frac{2\sigma_1 k}{h^2}\mathbf{D}_{xx}, \text{ and}\\ \mathbf{C} &= -(1 - \sigma_0 k)\mathbf{I}_{N-1} - \frac{2\sigma_1 k}{h^2}\mathbf{D}_{xx},\end{aligned} \tag{31}$$

where $\mu = \kappa k/h^2$ and (for simply supported boundary conditions)

$$\mathbf{D}_{xxxx} = \mathbf{D}_{xx}\mathbf{D}_{xx}, \tag{32}$$

with $\mathbf{D}_{xx}$ as defined in Eq. (10).

To apply the dynamic grid to the stiff string, the definition of $\mathbf{D}_{xx}$ in Eqs. (31) and (32) can simply be replaced by the $\boldsymbol{\mathcal{D}}_{xx}^n$ from Eq. (23). With reference to the general form in Eq. (20), the following matrices can then be applied to the alternative vector $\boldsymbol{u}$ from Eq. (21) to yield the dynamic stiff string:

$$\begin{aligned}\boldsymbol{\mathcal{A}}^n &= (1 + \sigma_0^n k)\mathbf{I}_{\mathcal{M}^n},\\ \boldsymbol{\mathcal{B}}^n &= 2\mathbf{I}_{\mathcal{M}^n} + (\lambda^n)^2 \boldsymbol{\mathcal{D}}_{xx}^n - (\mu^n)^2 \boldsymbol{\mathcal{D}}_{xxxx}^n + \frac{2\sigma_1^n k}{(h^n)^2}\boldsymbol{\mathcal{D}}_{xx}^n,\\ \boldsymbol{\mathcal{C}}^n &= -(1 - \sigma_0^n k)\mathbf{I}_{\mathcal{M}^n} - \frac{2\sigma_1^n k}{(h^n)^2}\boldsymbol{\mathcal{D}}_{xx}^n,\end{aligned} \tag{33}$$

where $\mu^n = \kappa^n k/(h^n)^2$ and (for simply supported boundary conditions)

$$\boldsymbol{\mathcal{D}}_{xxxx}^n = \boldsymbol{\mathcal{D}}_{xx}^n \boldsymbol{\mathcal{D}}_{xx}^n. \tag{34}$$

Notice that κ^n, σ_0^n, and σ_1^n are now allowed to be time-varying. Furthermore, h^n can be obtained by calculating the stability condition with equality. Sound examples and an implementation of the dynamic stiff string can be found via [21]. One example shows a string being made more inharmonic by increasing κ^n but decreasing c^n to retain the same pitch. Another varies L^n, c^n, κ^n, σ_0^n, and σ_1^n such that the timbral qualities of the string vary between a regular string, its octave, a vibraphone, and a marimba.

4 2D SYSTEMS

This section extends the framework presented in SEC. 2 to higher-dimensional systems. A rectangular 2D system can be described by a state variable $q = q(x, y, t)$ defined over $(x, y) \in [0, L_x] \times [0, L_y]$ with side-lengths L_x and L_y (both in meters). Discretizing q will result in grid function $q_{l,m}^n$ in which $l \in \{0, \ldots, N_x\}$ and $m \in \{0, \ldots, N_y\}$ and the number of intervals in the x-direction and y-direction are $N_x = \lfloor L_x/h \rfloor$ and $N_y = \lfloor L_y/h \rfloor$, respectively. Notice that the same grid spacing h is used in both the x-direction and y-direction, which is typical for isotropic, homogeneous systems.

Higher-dimensional systems can be written in matrix form by stacking or "flattening" the state. For Dirichlet or simply supported boundary conditions, the following $(N_x - 1)(N_y - 1) \times 1$ vector can be used to describe the state:

$$\begin{aligned}\mathbf{q}^n &= [(\boldsymbol{q}_1^n)^T, \ldots, (\boldsymbol{q}_{N_x-1}^n)^T]^T \quad \text{with}\\ \boldsymbol{q}_l^n &= [q_{l,1}^n, \ldots, q_{l,N_y-1}^n]^T.\end{aligned} \tag{35}$$

4.1 2D Dynamic Grid

Rather than splitting the original system into two subsystems, one can split it into four. Consider grid function u_{i,l_i,m_i}^n, where $i \in \{1, \ldots, 4\}$ is the index of the subsystem (also see Fig. 2).

The subsystems are subdivided into $M_{x,i}^n$ intervals in the x-direction and $M_{y,i}^n$ intervals in the y-direction, and the spatial indices have the following ranges: $l_i \in \{0, \ldots M_{x,i}^n\}$ and $m_i \in \{0, \ldots M_{y,i}^n\}$. Subsystems placed next to each other in the x-direction need to have the same number of points in the y-direction and vice versa. In other words, the following

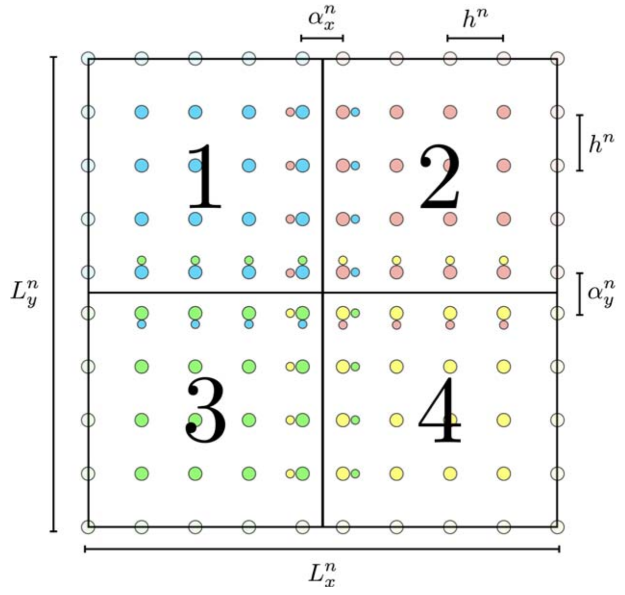


Fig. 2. Applying the dynamic grid to a 2D system. Virtual grid points are denoted as smaller circles, and boundary points are not included in the calculation (for Dirichlet/simply supported boundary conditions).

constraints apply to the number of intervals per subsystem:

$$M_{x,1}^n = M_{x,3}^n,\ \ M_{x,2}^n = M_{x,4}^n,\ \ M_{y,1}^n = M_{y,2}^n,\ \ M_{y,3}^n = M_{y,4}^n. \quad (36)$$

Furthermore, $0 < M_{x,1}^n < N_x^n$ and $M_{x,2}^n = N_x^n - M_{x,1}^n$, and $0 < M_{y,1}^n < N_y^n$ and $M_{y,3}^n = N_y^n - M_{y,1}^n$.

The grid points of each of the subsystems are positioned in the x-y plane as follows:

$$\begin{aligned}\left(x_{u_{1,l_1,m_1}}^n, y_{u_{1,l_1,m_1}}^n\right) &= \left(l_1 h^n, m_1 h^n\right),\\ \left(x_{u_{2,l_2,m_2}}^n, y_{u_{2,l_2,m_2}}^n\right) &= \left(L_x^n - (M_{x,2}^n - l_2)h^n, m_2 h^n\right),\\ \left(x_{u_{3,l_3,m_3}}^n, y_{u_{3,l_3,m_3}}^n\right) &= \left(l_3 h^n, L_y^n - (M_{y,3}^n - m_3)h^n\right),\\ \left(x_{u_{4,l_4,m_4}}^n, y_{u_{4,l_4,m_4}}^n\right) &= \Big(L_x^n - (M_{x,4}^n - l_4)h^n,\\ &\qquad L_y^n - (M_{y,4}^n - m_4)h^n\Big).\end{aligned} \quad (37)$$

Notice that, like in the 1D case in Eq. (14), the boundary points are fixed at the ends of the domain. If Dirichlet or simply supported boundary conditions are used, one can exclude the boundary points in the calculation, and the ranges for the spatial indices become

$$\begin{aligned} l_1 &= \{1, M_{x,1}^n\}, & m_1 &= \{1, M_{y,1}^n\},\\ l_2 &= \{0, M_{x,2}^n - 1\}, & m_2 &= \{1, M_{y,2}^n\},\\ l_3 &= \{1, M_{x,3}^n\}, & m_3 &= \{0, M_{y,3}^n - 1\},\\ l_4 &= \{0, M_{x,4}^n - 1\}, & m_4 &= \{0, M_{y,4}^n - 1\}.\end{aligned} \quad (38)$$

Finally, one can define the inner boundaries that connect u_1 and u_2 as the first vertical inner boundary and those connecting u_3 and u_4 as the second vertical inner boundary. The same can be done for the first and second horizontal inner boundaries, which are those connecting u_1 and u_3 and connecting u_2 and u_4, respectively.

Similar to Eq. (17) in the 1D case, virtual grid points at the first vertical and horizontal inner boundary are then calculated through

$$\begin{aligned} u_{1,M_{x,1}^n+1,m_1}^n &= \mathcal{I}_x^n u_{1,M_{x,1}^n,m_1}^n + u_{2,0,m_2}^n - \mathcal{I}_x^n u_{2,1,m_2}^n,\\ u_{2,-1,m_2}^n &= -\mathcal{I}_x^n u_{1,M_{x,1}^n-1,m_1}^n + u_{1,M_{x,1}^n,m_1}^n + \mathcal{I}_x^n u_{2,0,m_2}^n,\\ u_{1,l_1,M_{y,1}^n+1}^n &= \mathcal{I}_y^n u_{1,l_1,M_{y,1}^n}^n + u_{3,l_3,0}^n - \mathcal{I}_y^n u_{3,l_3,1}^n,\\ u_{3,l_3,-1}^n &= -\mathcal{I}_y^n u_{1,l_1,M_{y,1}^n-1}^n + u_{1,l_1,M_{y,1}^n}^n + \mathcal{I}_y^n u_{3,l_2,0}^n, \end{aligned} \quad (39)$$

respectively, and can be applied in the same manner to the second vertical and horizontal inner boundaries. Here, similar to Eq. (18) in the 1D case,

$$\mathcal{I}_x^n = \frac{\alpha_x^n - 1}{\alpha_x^n + 1}, \quad \text{and} \quad \mathcal{I}_y^n = \frac{\alpha_y^n - 1}{\alpha_y^n + 1}, \quad (40)$$

with

$$\alpha_x^n = \mathcal{N}_x^n - N_x^n, \quad \text{and} \quad \alpha_y^n = \mathcal{N}_y^n - N_y^n, \quad (41)$$

and fractional number of intervals in the x-direction and y-direction are $\mathcal{N}_x^n = L_x^n/h^n$ and $\mathcal{N}_y^n = L_y^n/h^n$, respectively.

4.1.1 Matrix Form

Similar to Eq. (35), one can stack the total state of the system into a single column vector. The order in which the subsystems are stacked, in terms of "column of subsystem no. #," is: 1, 3, ..., 1, 3, 2, 4, ..., 2, 4. If Dirichlet or simply supported boundary conditions are used, the total state can then be described as the following $\mathcal{M}_2^n \times 1$ column vector:

$$\boldsymbol{u}^n = [(\boldsymbol{v}^n)^T, (\boldsymbol{w}^n)^T]^T, \quad (42)$$

with $\mathcal{M}_2^n = (M_{x,1}^n + M_{x,2}^n)(M_{y,1}^n + M_{y,3}^n)$. Here,

$$\begin{aligned}\boldsymbol{v}^n &= [(\mathbf{v}_1^n)^T, \ldots, (\mathbf{v}_{M_{x,1}^n}^n)^T]^T, \quad \text{and}\\ \boldsymbol{w}^n &= [(\mathbf{w}_0^n)^T, \ldots, (\mathbf{w}_{M_{x,2}^n-1}^n)^T]^T,\end{aligned} \quad (43)$$

contain the states of subsystems 1 and 3 and subsystems 2 and 4, respectively:

$$\mathbf{v}_{j_1^n}^n = [(\mathbf{u}_{1,j_1^n}^n)^T, (\mathbf{u}_{3,j_1^n}^n)^T]^T, \ \ \mathbf{w}_{j_2^n}^n = [(\mathbf{u}_{2,j_2^n}^n)^T, (\mathbf{u}_{4,j_2^n}^n)^T]^T, \quad (44)$$

with $j_1^n = \{1, \ldots, M_{x,1}^n\}$ and $j_2^n = \{0, \ldots, M_{x,2}^n - 1\}$. Finally,

$$\begin{aligned}\mathbf{u}_{1,l_1}^n &= [u_{1,l_1,1}^n, \ldots, u_{1,l_1,M_{y,1}^n}^n]^T,\\ \mathbf{u}_{2,l_2}^n &= [u_{2,l_2,1}^n, \ldots, u_{2,l_2,M_{y,2}^n}^n]^T,\\ \mathbf{u}_{3,l_3}^n &= [u_{3,l_3,0}^n, \ldots, u_{3,l_3,M_{y,3}^n-1}^n]^T, \quad \text{and}\\ \mathbf{u}_{4,l_4}^n &= [u_{4,l_4,0}^n, \ldots, u_{1,l_4,M_{y,4}^n-1}^n]^T,\end{aligned} \quad (45)$$

where the ranges for l_i are specified in Eq. (38).

4.1.2 Adding and Removing Grid Points

Addition and removal of grid points happens in a similar fashion as described in SEC. 2.3, the difference being that an entire row or column of grid points is affected rather than a single grid point.

Considering only alterations in the left subsystems, a column can be added by carrying out the following operations on $\boldsymbol{v}$ in Eq. (43)

$$\begin{aligned}\boldsymbol{v}^n &:= [(\boldsymbol{v}^n)^T, \mathbf{Z}^n(I_3^n)^T]^T, \\ \boldsymbol{v}^{n-1} &:= [(\boldsymbol{v}^{n-1})^T, \mathbf{Z}^{n-1}(I_3^n)^T]^T,\end{aligned} \quad \text{if } N_x^n > N_x^{n-1}, \tag{46}$$

where

$$\begin{aligned}\mathbf{Z}^n &= [(\mathbf{v}^n_{M_{x,1}^{n-1}-1})^T, (\mathbf{v}^n_{M_{x,1}^{n-1}})^T, (\mathbf{w}_0^n)^T, (\mathbf{w}_1^n)^T] \quad \text{and} \\ \mathbf{Z}^{n-1} &= [(\mathbf{v}^{n-1}_{M_{x,1}^{n-1}-1})^T, (\mathbf{v}^{n-1}_{M_{x,1}^{n-1}})^T, (\mathbf{w}_0^{n-1})^T, (\mathbf{w}_1^{n-1})^T]\end{aligned} \tag{47}$$

contain the states of the vertical inner boundaries and their first neighbors in the x-direction.

Considering only alterations in the top subsystems, a row can be added by carrying out the following operations on $\mathbf{u}_1$ and $\mathbf{u}_2$ (at n and $n-1$) in Eq. (45):

$$\begin{aligned}\mathbf{u}_{1,l_1}^n &:= [(\mathbf{u}_{1,l_1}^n)^T, I_3^n\mathbf{z}_{1,l_1}^n]^T, \\ \mathbf{u}_{2,l_2}^n &:= [(\mathbf{u}_{2,l_2}^n)^T, I_3^n\mathbf{z}_{2,l_2}^n]^T,\end{aligned} \quad \text{if } N_y^n > N_y^{n-1}, \tag{48}$$

where

$$\begin{aligned}\mathbf{z}_{1,l_1}^n &= [u^n_{1,l_1,M_{y,1}^{n-1}-1}, u^n_{1,l_1,M_{y,1}^{n-1}}, u^n_{3,l_1,0}, u^n_{3,l_1,1}]^T, \\ \mathbf{z}_{2,l_2}^n &= [u^n_{2,l_2,M_{y,2}^{n-1}-1}, u^n_{2,l_2,M_{y,2}^{n-1}}, u^n_{4,l_2,0}, u^n_{4,l_2,1}]^T,\end{aligned} \tag{49}$$

contain the horizontal inner boundaries and their first neighbors in the y-direction [the ranges for l_1 and l_2 are given in Eq. (38)].

Removing grid points also happens in a similar fashion to the 1D case. Removing a column from the system happens according to

$$\begin{aligned}\boldsymbol{v}^n &:= [(\mathbf{v}_1^n)^T, \ldots, (\mathbf{v}^n_{M_{x,1-1}^{n-1}})^T]^T, \\ \boldsymbol{v}^{n-1} &:= [(\mathbf{v}_1^{n-1})^T, \ldots, (\mathbf{v}^{n-1}_{M_{x,1-1}^{n-1}})^T]^T,\end{aligned} \quad \text{if } N_x^n < N_x^{n-1}, \tag{50}$$

and removing a row from the system happens (at n and $n-1$) according to

$$\begin{aligned}\mathbf{u}_{1,l_1}^n &:= [u^n_{1,l_1,1}, \ldots, u^n_{1,l_1,M_{y,1}^{n-1}-1}]^T, \\ \mathbf{u}_{2,l_2}^n &:= [u^n_{2,l_2,1}, \ldots, u^n_{2,l_2,M_{y,1}^{n-1}-1}]^T,\end{aligned} \quad \text{if } N_y^n < N_y^{n-1}. \tag{51}$$

4.2 2D Wave Equation

The PDE of the 2D wave equation is defined as

$$\partial_t^2 q = c^2\Delta q, \tag{52}$$

with wave speed c (in meters per second) and Laplacian $\Delta = \partial_x^2 + \partial_y^2$. Eq. (52) can be discretized to the following FDTD scheme:

$$\delta_{tt}q_{l,m}^n = c^2\delta_\Delta q_{l,m}^n\ , \tag{53}$$

where $\delta_\Delta = \delta_{xx} + \delta_{yy}$ is the discrete Laplacian, and

$$\begin{aligned}\delta_{xx}q_{l,m}^n &= \frac{1}{h^2}\left(q_{l+1,m}^n - 2q_{l,m}^n + q_{l-1,m}^n\right) \approxeq \partial_x^2 q, \\ \delta_{yy}q_{l,m}^n &= \frac{1}{h^2}\left(q_{l,m+1}^n - 2q_{l,m}^n + q_{l,m-1}^n\right) \approxeq \partial_y^2 q.\end{aligned} \tag{54}$$

Finally, the stability condition is $h \geq \sqrt{2}ck$.

4.2.1 Matrix Form

Again assuming Dirichlet boundary conditions, one can define scaled (by h^2) matrix forms of δ_{xx} and δ_{yy} similar to Eq. (10), i.e., $(N_x-1)\times(N_x-1)$ matrix $\mathbf{D}_{xx}$ and $(N_y-1)\times(N_y-1)$ matrix $\mathbf{D}_{yy}$. These can be used to obtain a matrix form of the discrete Laplacian by performing a Kronecker sum [22, 23]:

$$\mathbf{D}_\Delta = \mathbf{D}_{yy} \oplus \mathbf{D}_{xx}, \tag{55}$$

which is of size $(N_x-1)(N_y-1)\times(N_x-1)(N_y-1)$. Using the same-sized identity matrix $\mathbf{I} = \mathbf{I}_{(N_x-1)(N_y-1)}$, one can use the general matrix form in Eq. (8), with $\mathbf{q}^n$ as defined in Eq. (35), and define the matrices as follows:

$$\mathbf{A} = \mathbf{I}, \quad \mathbf{B} = 2\mathbf{I} + \lambda^2\mathbf{D}_\Delta, \quad \text{and} \quad \mathbf{C} = -\mathbf{I}, \tag{56}$$

where, as in the 1D case, $\lambda = ck/h$.

To apply the dynamic grid to the 2D wave equation, one can use the general matrix form in Eq. (20) with state vector $\boldsymbol{u}^n$ from Eq. (42) and

$$\boldsymbol{\mathcal{A}}^n = \mathbf{I}_{\mathcal{M}_2^n},\ \boldsymbol{\mathcal{B}}^n = 2\mathbf{I}_{\mathcal{M}_2^n} + (\lambda^n)^2\boldsymbol{\mathcal{D}}_\Delta^n,\ \boldsymbol{\mathcal{C}}^n = -\mathbf{I}_{\mathcal{M}_2^n}. \tag{57}$$

Here, $\mathcal{M}_2^n \times \mathcal{M}_2^n$ matrix

$$\boldsymbol{\mathcal{D}}_\Delta^n = \boldsymbol{\mathcal{D}}_{yy}^n \oplus \boldsymbol{\mathcal{D}}_{xx}^n, \tag{58}$$

where $\boldsymbol{\mathcal{D}}_{xx}^n$ and $\boldsymbol{\mathcal{D}}_{yy}^n$ include the effect of the interpolation at the inner boundaries presented in Eq. (37) and are as defined in Eq. (23) with $\mathcal{I}_x^n$ and $\mathcal{I}_y^n$ as defined in Eq. (40), respectively. Finally, $h^n = \sqrt{2}c^nk$.

4.3 Damped Thin Plate

Similar to the damped stiff string presented in Sec. 3, one can extend the dynamic grid method to other systems using higher-order spatial derivatives. Consider the PDE of a damped thin plate [7]:

$$\partial_t^2 q = -\kappa^2\Delta\Delta q - 2\sigma_0\partial_t q + 2\sigma_1\partial_t\Delta q, \tag{59}$$

where parameters are similarly defined to the damped stiff string in Eq. (28). This can be discretized to

$$\delta_{tt}q_{l,m}^n = -\kappa^2\delta_{\Delta\Delta}q_{l,m}^n - 2\sigma_0\delta_{t\cdot}q_{l,m}^n + 2\sigma_1\delta_{t-}\delta_\Delta q_{l,m}^n, \tag{60}$$

where $\delta_{\Delta\Delta} = \delta_\Delta\delta_\Delta$ and $h \geq 2\sqrt{\kappa k}$. Using state vector $\mathbf{q}^n$ from Eq. (35), Eq. (60) can be rewritten using the general matrix form in Eq. (8) using

$$\begin{gathered}\mathbf{A} = (1+\sigma_0 k)\mathbf{I}, \quad \mathbf{B} = 2\mathbf{I} - \mu^2\mathbf{D}_{\Delta\Delta} + \tfrac{2\sigma_1 k}{h^2}\mathbf{D}_\Delta, \\ \text{and} \quad \mathbf{C} = -(1-\sigma_0 k)\mathbf{I} - \tfrac{2\sigma_1 k}{h^2}\mathbf{D}_\Delta.\end{gathered} \tag{61}$$

Here, $\mathbf{I} = \mathbf{I}_{(N_x-1)(N_y-1)}$, and, for simply supported boundary conditions, same-sized matrix

$$\mathbf{D}_{\Delta\Delta} = \mathbf{D}_\Delta\mathbf{D}_\Delta, \tag{62}$$

with $\mathbf{D}_\Delta$ as defined in Eq. (55).

Similar to before, the dynamic grid can be applied to this system by replacing the definitions of $\mathbf{D}_\Delta$ in Eqs. (61) and (62) by $\boldsymbol{\mathcal{D}}_\Delta^n$ from Eq. (58). Using $\boldsymbol{u}^n$ from Eq. (42), the

Table 1. Maximum frequency deviation for the dynamic thin plate (in cents) for different values of $\mathcal{N}_x^n$ and $\mathcal{N}_y^n$.

$\mathcal{N}_x^n \backslash \mathcal{N}_y^n$	15 → 16	16 → 17	17 → 18	18 → 19	19 → 20
15 → 16	–96.00	–93.78	–91.82	–90.06	–88.49
16 → 17	–93.78	–91.54	–89.55	–87.75	–86.15
17 → 18	–91.82	–89.55	–87.52	–85.69	–84.05
18 → 19	–90.06	–87.75	–85.69	–83.84	–82.16
19 → 20	–88.49	–86.15	–84.05	–82.16	–80.47

dynamic thin plate can be written as the general form in Eq. (20) with

$$\begin{aligned}\boldsymbol{\mathcal{A}}^n &= (1+\sigma_0^n k)\mathbf{I}_{\mathcal{M}_2^n},\\ \boldsymbol{\mathcal{B}}^n &= 2\mathbf{I}_{\mathcal{M}_2^n} - (\mu^n)^2\boldsymbol{\mathcal{D}}_{\Delta\Delta}^n + \frac{2\sigma_1^n k}{(h^n)^2}\boldsymbol{\mathcal{D}}_{\Delta}^n, \quad \text{and}\\ \boldsymbol{\mathcal{C}}^n &= -(1-\sigma_0^n k)\mathbf{I}_{\mathcal{M}_2^n} - \frac{2\sigma_1^n k}{(h^n)^2}\boldsymbol{\mathcal{D}}_{\Delta}^n.\end{aligned} \tag{63}$$

where $h^n = 2\sqrt{\kappa^n k}$, and for simply supported boundary conditions

$$\boldsymbol{\mathcal{D}}_{\Delta\Delta}^n = \boldsymbol{\mathcal{D}}_{\Delta}^n\boldsymbol{\mathcal{D}}_{\Delta}^n, \tag{64}$$

which is of size $\mathcal{M}_2^n \times \mathcal{M}_2^n$. Sound examples can be found via [21]. One example shows a change in κ^n (and σ_1^n) modeling an increase in plate thickness, whereas another shows changes in L_y^n.

5 RESULTS

5.1 Modal Analysis

One can retrieve the modal frequencies of the implementation of the dynamic grid by performing a modal analysis of the update equation in matrix form. For reference, the general form is Eq. (20) with matrices of the various systems defined in Eqs. (22), (33), (57), and (63). Assuming $\sigma_0^n = \sigma_1^n = 0$, because the influence of losses on modal frequencies are very small for damped systems and for sub-audio–rate parameter variations, the p^{th} numerical modal frequency can be calculated as

$$f_p^n \approx \frac{1}{2\pi k}\cos^{-1}\left(\frac{1}{2}\text{eig}_p(\boldsymbol{\mathcal{B}}^n)\right), \tag{65}$$

where $\text{eig}_p(\cdot)$ denotes the "p^{th} eigenvalue of."

To determine the accuracy of the frequencies obtained in Eq. (65), one can compare modal frequencies against expected values. Results of a modal analysis of the dynamic stiff string with $\kappa^n \approx 1.26$ m^2/s and linear changes in c^n corresponding to $\mathcal{N}^n = 15 \rightarrow 20$ can be seen in Fig. 3(a), as well as the expected modal frequencies.

The largest frequency deviation for the dynamic stiff string with $N^n = 15$ is around -67 cents, whereas this decreases to around -56 cents for $N^n = 19$. This is similar to the results presented in [16]. Setting $c^n = 0$ (effectively reducing the stiff string to an ideal bar [7]), the maximum frequency deviation for $N^n = 15$ is around –96 cents.

Results for the dynamic thin plate appear in Table 1. It can be shown that for a square 2D system, the maximum frequency deviations are identical to the case of the analogous 1D system. In other words, the 1D and 2D wave equations share maximum frequency deviations for equal values of N^n (1D) and N_x^n and N_y^n (2D). This also holds for the ideal bar and thin plate. This indicates that frequency deviations do not worsen when applying the dynamic grid to higher dimensional systems.

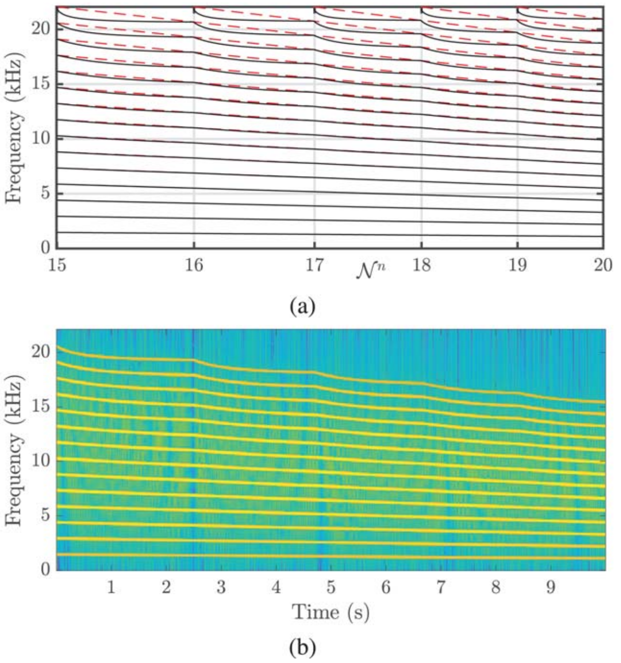


Fig. 3. The dynamic stiff string [Eq. (20) with Eq. (33)] with $\kappa \approx 1.26$ and c^n is (linearly) swept such that $\mathcal{N}^n = 15 \rightarrow 20$. (a) Results of Eq. (65). Modes exhibited by the system are shown in solid black, and expected modal frequencies are shown using dashed red lines. (b) Output of the dynamic stiff string excited at $n = 0$ and $l = 1$ and output retrieved at the same location (f_s = 44.1 kHz).

In both the 1D and 2D cases, the system exhibits the following behaviors:

- If $\alpha^n = 0$ and parameters are static, the split system exhibits identical behavior to the original.
- The number of modes is always equal to the number of grid points in the system (excluding boundary points).
- The higher the mode number, the more its frequency deviates from the expected frequency for one value of N^n.
- The higher the total number of grid points in the simulation, the smaller the frequency deviation.
- The location where a system is split (and thus the location where grid points are added/removed) does not influence the frequency content.

5.2 System Behavior

Fig. 3(b) shows the output of the dynamic stiff string (simulated at $f_s = 44.1$ kHz) with $\mathcal{N}^n = 15$ excited at $t = 0$ s, after which c^n is (linearly) swept such that $\mathcal{N}^n = 20$ at $t = 10$ s. One can observe that the highest mode pre-

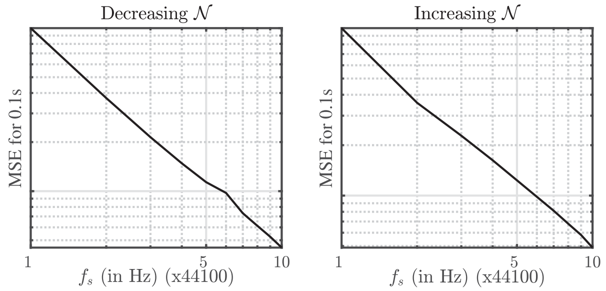


Fig. 4. Mean squared error (MSE) for Eq. (13) at a different sample with a reference of $f_s = 30 \cdot 44{,}100$ Hz.

dicted by the analysis in Fig. 3(a) is not excited, because the inner boundaries act as one grid point because of the rigid connection at the time of excitation [see Eq. (15)]. Furthermore, when a change in grid configuration occurs, the energy of modes follows the "path of least resistance" predicted in Fig. 3(a). If modes of two grid configurations align, the energy transfer happens smoothly, and no noticeable artefacts occur (under sub-audio–rate parameter variations). This also means that every time a grid point, and thus a mode, is added at the top of the spectrum, it will not obtain any energy.

If, on the other hand, $\mathcal{N}^n = 20 \rightarrow 15$ [effectively horizontally flipping Fig. 3(a)], modes seem to "disappear" around the Nyquist frequency when grid points are removed. Because there are no modes in the new grid configuration to receive energy, its energy is distributed over the remaining modes, causing audible artefacts. Informal experiments show that if modes are damped before they "disappear," noticeable artefacts are prevented. In the case of the stiff string, this is what frequency-dependent losses automatically allow. The minimum value required for σ_1^n to damp the highest mode depends on various factors, such as the values of other parameters and speed of parameter variation. In the case of 2D systems, if parameters are varied such that grid points need to be removed, multiple grid points, and thus modes, "disappear" at once. Although these do not only include the highest frequency modes, damping does seem to take care of noticeable artefacts.

For the lossless systems presented in this paper, local losses could be added at the inner boundaries, presented in [16, 17] as displacement or state correction. Although this successfully prevents noticeable artefacts, this method does introduce unnatural losses to the system.

5.3 Convergence

Although a full proof of convergence for the dynamic grid is left for future work, Fig. 4 demonstrates converging behavior. Eq. (13) is excited, and the length is changed from $L = 1$ m to $L = 0.5$ m over the course of 0.1s (left panel) or vice versa (right panel) and ran at different sample rates. The reference signal is created by retrieving the output of running Eq. (13) at $f_s = 30 \cdot 44{,}100$ Hz. Using a log-log plot to show the mean squared error, one can observe a quasi-linear decrease, indicating convergence with first-order accuracy in k. This result is confirmed by the fact that at higher sample rates, audible artefacts happen to a lesser degree.

6 DISCUSSION

As shown in SEC. 5, modal frequencies exhibited by the dynamic grid deviate from those expected. However, the largest frequency deviations occur at higher-frequency ranges, which are less perceptually relevant, and damped most quickly because of visco-thermal effects. For the 1D systems presented in this paper, when a (quasi-)harmonic output is expected, these deviations could cause beating effects, though listening tests will have to confirm this.

It is obviously not ideal to rely on (frequency-dependent) losses for grid points to be removed without noticeable artefacts. However, it can be argued that relying on losses that are often used in existing models is more natural than adding artificial damping at specific locations in the grid, as has been done in previous work [16, 17]. Furthermore, convergence of the method, although not proven, has been demonstrated in SEC. 5.3 and shows that at higher sample rates—although less suited for real-time applications—audible artefacts happen to a lesser degree.

7 CONCLUSION

This paper presents the dynamic grid, a method to smoothly change grid configurations of FDTD schemes allowing for runtime parameter variation. Furthermore, the Courant-Friedrichs-Lewy–type conditions of these schemes can be satisfied with equality at all times, maximizing simulation quality or bandwidth for any choice of parameters. The authors extend previous work on the time-varying 1D wave equation in [16] by applying the dynamic grid to the stiff string, 2D wave equation, and thin plate.

Future directions include the extension of the method presented here to 3D, performing another Kronecker sum as in Eq. (58) for the third dimension (see [23]). Other future considerations include listening tests to confirm the absence of audible artefacts and real-time implementation and control, such that a player can "mold" their instrument while performing, potentially discovering new ways of expression. Finally, full proof of convergence of the method needs to be established.

8 ACKNOWLEDGMENT

This work has been funded in part by the European Art-Science-Technology Network for Digital Creativity (EASTN-DC), project number 883023, and the Nordic Sound and Music Computing Network by NordForsk. This work is also funded in part by the European Research Council under the European Union's Horizon 2020 Research and Innovation Programme, grant agreement number 950084-StG-NEMUS. Thanks to the Conference on Digital Audio Effects (DAFx20in21) committee for providing the open access publication of this work.

9 REFERENCES

[1] M. E. McIntyre, R. T. Schumacher, and J. Woodhouse, “On the Oscillations of Musical Instruments,” *J. Acoust. Soc. Am.*, vol. 74, no. 5, pp. 1325–1345 (1983 Nov.). https://doi.org/10.1121/1.390157.

[2] G. Borin, G. De Poli, and A. Sarti, “A Modular Approach to Excitator-Resonator Interaction in Physical Models Syntheses,” in *Proceedings of the International Computer Music Conference*, pp. 46–50 (Columbus, OH) (1989 Nov.).

[3] G. F. Carrier, “On the Non-Linear Vibration Problem of the Elastic String,” *Quart. Appl. Math.*, vol. 3, no. 2, pp. 157–165 (1945). https://doi.org/10.1090/qam/12351.

[4] A. Hirschberg, J. Gilbert, R. Msallam, and A. P. J. Wijnands, “Shock Waves in Trombones,” *J. Acoust. Soc. Am.*, vol. 99, no. 3, pp. 1754–1758 (1996 Mar.). https://doi.org/10.1121/1.414698.

[5] S. Willemsen, *The Emulated Ensemble: Real-Time Simulation of Musical Instruments Using Finite-Difference Time-Domain Methods*, Ph.D. thesis, Aalborg University Copenhagen, Copenhagen, Denmark (2021 Jul.).

[6] V. Välimäki, J. Pakarinen, C. Erkut, and M. Karjalainen, “Discrete-Time Modelling of Musical Instruments,” *Rep. Prog. Phys.*, vol. 69, no. 1, pp. 1–78 (2006 Jan.). http://dx.doi.org/10.1088/0034-4885/69/1/R01.

[7] S. Bilbao, *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics* (Wiley, Hoboken, NJ, 2009). https://doi.org/10.1002/9780470749012.

[8] J. D. Morrison and J.-M. Adrien, “Mosaic: A Framework for Modal Synthesis,” *Comp. Music J.*, vol. 17, no. 1, pp. 45–56 (1993 Jan.). https://doi.org/10.2307/3680569.

[9] S. Mehes, M. van Walstijn, and P. Stapleton, “Towards a Virtual-Acoustic String Instrument,” in *Proceedings of the 13th International Conference on Sound and Music Computing (SMC)*, pp. 286–292 (Hamburg, Germany) (2016 Aug.). https://doi.org/10.5281/zenodo.851257.

[10] S. Willemsen, S. Serafin, and J. R. Jensen, “Virtual Analog Simulation and Extensions of Plate Reverberation,” in *Proceedings of the 14th International Conference on Sound and Music Computing (SMC)*, pp. 314–319 (Espoo, Finland) (2017 Jul.). https://doi.org/10.5281/zenodo.1401951.

[11] III J. Smith, “Physical Modeling Using Digital Waveguides,” *Comp. Music J.*, vol. 16, no. 4, pp. 74–91 (1992). https://doi.org/10.2307/3680470.

[12] R. Michon and J. Smith, “A Hybrid Guitar Physical Model Controller: The BladeAxe,” in *Proceedings of the 11th International Conference on Sound and Music Computing (SMC)*, pp. 573–579 (Athens, Greece) (2014 Sep.). https://doi.org/10.5281/zenodo.850619.

[13] S. Serafin, S. Gelineck, N. Böttcher, and L. Martinussen, “Virtual Reality Instruments Capable of Changing Physical Dimensions in Real-Time,” in *Proceedings of the 2nd International Conference on Enactive Interfaces* (Genova, Italy) (2005 Nov.).

[14] J. Leonard, C. Cadoz, N. Castagné, J.-L. Florens, and A. Luciani, “A Virtual Reality Platform for Musical Creation: GENESIS-RT,” in M. Aramaki, O. Derrien, R. Kronland-Martinet, S. Ystad (Eds), *Sound, Music, and Motion*, Lecture Notes in Computer Science, vol. 8905, pp. 346–371 (Springer, Cham, Switzerland, 2013). https://doi.org/10.1007/978-3-319-12976-1_22.

[15] A. Hofmann, S. Schmutzhard, M. Pàmies-Vilà, G. Erdogan, and V. Chatziioannou, “An Opcode Implementation of a Finite Difference Viscothermal Time-Domain Model of a Tube Resonator for Wind Instrument Simulations,” in *Proceedings of the 5th International Csound Conference (ICSC)* (Cagli, Italy) (2019 Sep.).

[16] S. Willemsen, S. Bilbao, M. Ducceschi, and S. Serafin, “Dynamic Grids for Finite-Difference Schemes in Musical Instrument Simulations,” in *Proceedings of the 24th International Conference on Digital Audio Effects (DAFx)*, pp. 144–151 (Online) (2021 Sep.). https://doi.org/10.23919/DAFx51585.2021.9768302.

[17] S. Willemsen, S. Bilbao, M. Ducceschi, and S. Serafin, “A Physical Model of the Trombone Using Dynamic Grids for Finite-Difference Schemes,” in *Proceedings of the 24th International Conference Digital Audio Effects (DAFx)*, pp. 152–159 (Online) (2021 Sep.). https://doi.org/10.23919/DAFx51585.2021.9768286.

[18] R. Courant, K. Friedrichs, and H. Lewy, “Über die Partiellen Differenzengleichungen de Mathematischen Physik,” *Math. Ann.*, vol. 100, pp. 32–74 (1928 Dec.). https://doi.org/10.1007/BF01448839.

[19] S. Willemsen, N. Andersson, S. Serafin, and S. Bilbao, “Real-Time Control of Large-Scale Modular Physical Models Using the Sensel Morph,” in *Proceedings of the 16th International Conference on Sound and Music Computing (SMC)*, pp. 275–280 (Málaga, Spain) (2019 May). https://doi.org/10.5281/zenodo.3249295.

[20] J. Bensa, S. Bilbao, R. Kronland-Martinet, and III J. O. Smith, “The Simulation of Piano String Vibration: From Physical Models to Finite Difference Schemes and Digital Waveguides,” *J. Acoust. Soc. Am.*, vol. 114, no. 2, pp. 1095–1107 (2003 Aug.). https://doi.org/10.1121/1.1587146.

[21] S. Willemsen, “Dynamic Grid Repository,” https://tinyurl.com/DynGridJAES (accessed Jan. 31, 2021).

[22] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis* (Cambridge University Press, Cambridge, UK, 1991).

[23] B. Hamilton, *Finite Difference and Finite Volume Methods for Wave-Based Modelling of Room Acoustics*, Ph.D. thesis, The University of Edinburgh, Edinburgh, UK (2016 Nov.).

THE AUTHORS

Silvin Willemsen

Stefan Bilbao

Michele Ducceschi

Stefania Serafin

Silvin Willemsen is a Postdoctoral Researcher at Aalborg University in Copenhagen, Denmark. He received his M.Sc. in Sound and Music Computing from Aalborg University in 2017. In 2018, he was appointed a Ph.D. Stipend at the Department of Architecture, Design and Media Technology at Aalborg University Copenhagen and was affiliated with the Multisensory Experience Lab. In 2021, he received his Ph.D. degree, and he continues to work in the field of physical modeling for musical instruments.

•

Stefan Bilbao (B.A. Physics, Harvard, 1992; M.Sc. and Ph.D. Electrical Engineering, Stanford, 1996 and 2001, respectively) is currently Professor of Acoustics and Audio Signal Processing in the Acoustics and Audio Group at the University of Edinburgh, and he previously held positions at the Sonic Arts Research Centre, Queen's University Belfast, and Stanford Space Telecommunications and Radioscience Laboratory. He is an Associate Editor of the IEEE/ACM Transactions on Audio Speech and Language Processing. He was born in Montreal, Quebec, Canada.

•

Michele Ducceschi [B.Sc. Physics (2008), University of Padova, Italy; M.Sc. Acoustics (2010), University of Edinburgh, UK; Ph.D. Mechanical Engineering (2014), ENSTA and Ecole Polytechnique, France] is an Associate Professor at the University of Bologna in Italy. He is currently the Principal Investigator of the European Research Council–funded project NEMUS (2021–2026). Previously, he held positions at the University of Edinburgh. He was a Royal Society Newton International Fellow and Leverhulme Early Career Fellow.

•

Stefania Serafin is Professor of Sonic interaction design at Aalborg University in Copenhagen and the leader of the Multisensory Experience Lab with Rolf Nordahl. She is the President of the Sound and Music Computing (SMC) Association, Project Leader of the Nordic SMC network, and lead of the SMC Master at Aalborg University. Stefania received her Ph.D. from Stanford University in 2004. Her research on sonic interaction design and sound for virtual and augmented reality with applications in health and culture can be found at https://tinyurl.com/35wjk3jn.