

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

SILVIN WILLEMSSEN,¹ STEFAN BILBAO,² MICHELE DUCCESCHI,³ AND STEFANIA SERAFIN¹

(sil@create.aau.dk)

(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 modelling synthesis for musical instruments exist. Finite-difference time-domain (FDTD) methods are known for their generality and flexibility in terms of the systems one can model, but are inflexible 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 FD schemes based on (sub-audio rate) time variation of parameters. This allows for extensions of the behaviour 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 and the stiff string, as well as to 2D systems, including the 2D wave equation and the thin plate. Results show that the method does not introduce noticeable artefacts when changing between grid configurations, though this needs to be confirmed by listening tests. Some insights into stability analysis and real-time implementation are provided.

0 INTRODUCTION

The functioning of nearly any musical instrument can be subdivided into an exciter and resonator components [?, ?]. Examples of resonators are the stringed instrument body and the brass instrument bore, which are excited by the bow and the lips of the player respectively. The resonator is often assumed to be linear (exceptions being the nonlinear string [?], or shock waves in bores [?, ?, ?]) 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 where the defining parameters of the resonator are time-varying exist. A notable example is the trombone, where the length of the acoustic tube is changed during performance. Furthermore, membrane tension in timpani or “hourglass drums”¹ are varied. See [?, 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 modelling techniques. Many techniques exist and a detailed summary can be found here [?]. Finite-difference time-domain

(FDTD) methods, while not the most efficient, are flexible and generalisable in terms of the systems one can model [?].

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. First of all, difficulties arise when working with time-varying systems, both in the underlying continuous equations, as well as stability issues arising in their numerical implementation. Secondly, 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 virtual musical instrument models is to exploit the virtual nature of the instrument and go beyond what is physically possible. Instrument properties that are normally fixed could be made time-varying to create sounds impossible in the real world and potentially extend the range of expression for the musician. Already existing work on time-varying parameters in physical models using modal synthesis [?] are shown in [?, ?], digital waveguides [?]

¹Ayan Bisi Adeleke - Master talking drummer - drum talks: <https://youtu.be/B4oQJZ2TEVI?t=9>

are shown in [?, ?], and mass-spring systems are shown in [?]. An acoustic tube with time-varying length implemented using FDTD methods is presented in [?] 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 generalises the method presented in [?] where 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 the thin plate. Changes in parameter values are assumed to be sub-audio rate (control rate) such that they can be applied to commonly used FD schemes. The method appears in part in [?, Ch. 12] and has been used to model the trombone, including time-varying length in [?]. Grid points are added along the grid as opposed to [?], where this only happens at the radiating end.

This paper is structured as follows: Sec. ?? presents the 1D wave equation as a starting point and Sec. ?? introduces FDTD methods and discusses stability and simulation quality. Sec. ?? introduces the dynamic grid and its application to the 1D wave equation and the stiff string, and Sec. ?? extends the method to 2D systems. Sec. ?? presents the analysis of the method and its results and a discussion and concluding remarks appear in Sec. ?? and Sec. ?? respectively.

1 1D SYSTEMS

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

$$\partial_t^2 q = c^2 \partial_x^2 q, \quad (1)$$

Here, c is the wave speed (in m/s), and derivatives with respect to t and x are denoted by ∂_t and ∂_x respectively. In the above, q can be interpreted as the transverse displacement of an ideal string, or the acoustic pressure in a cylindrical tube. Two possible choices of boundary condition are the Dirichlet and Neumann conditions, and can be interpreted as ‘fixed’ and ‘free’ boundaries respectively for the ideal string, or ‘open’ and ‘closed’ boundaries respectively for the acoustic tube. The Dirichlet boundary condition is defined as

$$q(0, t) = q(L, t) = 0. \quad (2)$$

The Neumann condition will not be used in this paper and is thus excluded for brevity.

1.1 Numerical Methods

To discretise Eq. (??) using FDTD methods, a spatio-temporal grid needs to be defined first. Time $t \geq 0$ can be discretised as $t = nk$ with temporal index $n = 0, 1, 2, \dots$ and time-step $k = 1/f_s$ (in s) with sample rate f_s (in Hz). 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 $l \in \{0, \dots, N\}$.

Using these definitions, the continuous state can be approximated as $q(x, t) \approx q_l^n$ where 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. (??) are approximated as

$$\partial_t^2 q \approx \delta_{tt} q_l^n \triangleq \frac{1}{k^2} (q_l^{n+1} - 2q_l^n + q_l^{n-1}), \quad (3a)$$

$$\partial_x^2 q \approx \delta_{xx} q_l^n \triangleq \frac{1}{h^2} (q_{l+1}^n - 2q_l^n + q_{l-1}^n). \quad (3b)$$

Eq. (??) can then be discretised to the following finite-difference (FD) scheme:

$$\delta_{tt} q_l^n = c^2 \delta_{xx} q_l^n, \quad (4)$$

which can be expanded to the following update equation

$$q_l^{n+1} = 2q_l^n - q_l^{n-1} + \lambda^2 (q_{l+1}^n - 2q_l^n + q_{l-1}^n), \quad (5)$$

where here,

$$\lambda = \frac{ck}{h} \quad (6)$$

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

From Eq. (??) one can observe that at the end points of the system ($l = 0$, and $l = N$) grid points outside the defined domain seem to be needed, i.e., q_{-1}^n and q_{N+1}^n . One can discretise the Dirichlet boundary conditions in Eq. (??) to

$$q_0^n = q_N^n = 0. \quad (7)$$

In practice, this means to simply reduce the range of calculation to $l = \{1, \dots, N - 1\}$, which takes care of the aforementioned issue.

1.2 Matrix Form

Both for compact implementation of FD schemes as well as for applying the dynamic grid to more complex systems, it is useful to write the update equation in Eq. (??) in matrix form.

Using Dirichlet boundary conditions, the system state at time n can be written as the following $(N - 1) \times 1$ column vector $\mathbf{q}^n = [q_1^n, \dots, q_{N-1}^n]^T$ where T denotes the transpose operation. Notice that the boundaries (q_0^n and q_N^n) are not included in the state vector as they are 0 at all times.

Expanding δ_{tt} , Eq. (??) can then be written in matrix form as

$$\mathbf{q}^{n+1} = (2\mathbf{I}_{N-1} + \lambda^2 \mathbf{D}_{xx}) \mathbf{q}^n - \mathbf{q}^{n-1} \quad (8)$$

with $(N - 1) \times (N - 1)$ matrix

$$\mathbf{D}_{xx} = \begin{bmatrix} \ddots & \ddots & & \mathbf{0} \\ \ddots & -2 & 1 & \\ & 1 & -2 & 1 \\ & & 1 & -2 & \ddots \\ \mathbf{0} & & & \ddots & \ddots \end{bmatrix}, \quad (9)$$

and same-sized identity matrix \mathbf{I}_{N-1} .

1.3 Stability and Simulation Quality

In order to ensure stability for the scheme in Eq. (??) the Courant number in Eq. (??) needs to satisfy the CFL condition [?]:

$$\lambda \leq 1. \quad (10)$$

If $\lambda = 1$, Eq. (??) provides an exact solution to Eq. (??). 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.

Usually, Eq. (??) is rewritten in terms of the grid spacing such that

$$h \geq ck, \quad (11)$$

which is implemented as

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

Here, $\lfloor \cdot \rfloor$ denotes the flooring operation which is necessary to ensure an integer number of intervals. Equation (??) 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. ??, the goal of this work is to introduce time-varying parameters into FDTD-based simulations. Leaving sample rate f_s , and thus k fixed, one can observe from the 1D wave equation 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. Several computational issues emerge.

OK, this maybe needs more explanation. As in, you should say that you are holding k and L fixed. Added static k above. L can still change as will be done in the 2D case. Also, you're working implicitly from the assumption that $\lambda = 1$ to start with. But you don't need to do this...might be better to introduce λ through an initial $h = ck/\lambda$. Hmm.. not sure how to write this. Should I say that we can now set λ freely (not calculating it from existing parameters) as long as it satisfies Eq. (??)? Can't you just write $h = ck/\lambda$ in Eq. 12? I suppose so, but I do refer back to (??) quite a bit in the explanation. I think it's important to show that the fractional \mathcal{N} removes the flooring operation in (??), so it's better to write it out like this I think. Besides, L is not necessarily fixed, so then $h = ck/\lambda$ would not be correct. In any case please tell me if you think differently! OK, here, not sure I get it. $h = ck/\lambda$ is just a definition isn't it? Like it always has to be correct, right? I suppose it depends on how you calculate λ . If it is defined as the usual $\lambda = ck/h$, the calculation $h = ck/\lambda$ will not include the length (it will simply boil down to $h = h$). To have L included, λ will have to be defined as $\lambda = (ck/L)\lfloor L/ck \rfloor$ I believe. First, a change in the wave speed c will cause a change in λ , which

causes issues regarding stability and simulation quality as detailed in Sec. ??. Secondly, and more importantly, changing c (or L) changes the number of intervals N according to Eq. (??) 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.

In this work, a *fractional* number of intervals $\mathcal{N} = L/h$ is proposed such that $N = \lfloor \mathcal{N} \rfloor$. This removes the need for the flooring operation in Eq. (??) (and therefore the recalculation of h), and allows $\lambda = 1$ at all times. Furthermore, \mathcal{N} will potentially allow 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 $x_{q_l}^n$ (in m 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_l^n , with $l = \{0, \dots, N^n\}$ is split into two subsystems: $u_{l_u}^n$ with $l_u = \{0, \dots, M_u^n\}$ and $w_{l_w}^n$ with $l_w = \{0, \dots, M_w^n\}$:

$$\delta_{tt} u_{l_u}^n = (c^n)^2 \delta_{xx} u_{l_u}^n, \quad (13a)$$

$$\delta_{tt} w_{l_w}^n = (c^n)^2 \delta_{xx} w_{l_w}^n, \quad (13b)$$

which have $M_u^n + 1$ grid points and $M_w^n + 1$ grid points respectively.² Here, $0 < M_u^n < N^n$ and $M_w^n = N^n - M_u^n$ and will thus contain one more grid point than the original system. Both systems are placed on the same domain x with their locations defined as

$$x_{u_{l_u}}^n = l_u h^n, \quad x_{w_{l_w}}^n = L^n - (M_w^n - l_w) h^n. \quad (14)$$

The subscripting gets 4 layers deep here...this is pretty heavy! Any way to simplify? Hmm.. possibly using a subscript m for w and keep l for u . But in the 2D case we'll get a general indexing l_i for the i^{th} subsystem...Seems to me you could write just x_{l_u} rather than $x_{u_{l_u}}$. Hmm.. when using a number it will become ambiguous which system it is. (Is x_0^n the point where $l_u = 0$ or $l_w = 0$)? OK, I sort of see, but this is getting very very hard to follow for me! I get that.. Maybe a grid point as a function of x^n ? So $x^n(u_{l_u})$? That's also nasty... I'll think a little more about it.

What about $\mathcal{X}^n(u_{l_u})$ or some other "special" 'x'-symbol? See Figure ??. The grid locations $l_u = 0$ and $l_w = M_w^n$ are referred to as the *outer boundaries* and, as can be observed from Eq. (??), are fixed to be at the limits of the system domain, i.e., $x_{u_0}^n = 0$ and $x_{w_{M_w^n}}^n = L^n, \forall n$. Furthermore, Dirichlet boundary conditions are imposed on the outer boundaries according to Eq. (??). The grid locations $l_u = M_u^n$ and $l_w = 0$ are referred to as the *inner boundaries*. If $\mathcal{N}^n = N^n$, i.e. \mathcal{N}^n is an integer, the inner boundaries overlap and need to abide the following condition

$$u_{M_u^n}^n = w_0^n, \quad \text{if } x_{u_{M_u^n}}^n = x_{w_0}^n, \quad (15)$$

²It is important to note that the superscripts n in M_u^n and M_w^n are unaffected by the δ_{tt} operator after expansion.

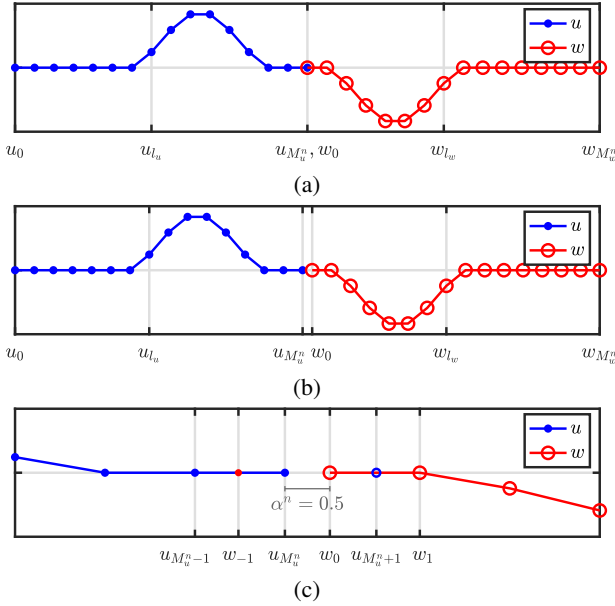


Fig. 1: Illustration of the dynamic grid. The x-axis shows the location of the respective grid points with ‘ x^n ’ omitted for brevity. (a) $N^n = 30$. As $N^n = N^n$, the inner boundaries overlap. (b) $N^n = 30.5$. The wave speed c^n and thus h^n are decreased, $N^n \neq N^n$ and the inner boundaries no longer overlap. (c) Figure ?? zoomed-in. All grid points (including virtual grid points) used in Eq. (??) are shown. The distance between the inner boundaries is expressed using α^n in Eq. (??). (Image adapted from [?].)

which acts as a rigid connection between the inner boundaries. As shown in [?], combining this condition with Neumann boundary conditions imposed on the inner boundaries shows that the virtual grid points needed to calculate $u_{M_u^n}^n$ and w_0^n can be set according to:

$$u_{M_u^n+1}^n = w_1^n, \quad \text{and} \quad w_{-1}^n = u_{M_u^n-1}^n. \quad (16)$$

This can be shown to yield identical behaviour to the original system.

Consider a decrease in wave speed c^n , which yields a decrease in h^n . This causes all grid points to move towards their respective outer boundary according to Eq. (??) (see Figure ??). As the inner boundaries no longer overlap, i.e., $N^n \neq N^n$, Eq. ?? can not be used and alternative definitions for the virtual grid points need to be found. To this end, quadratic Lagrange interpolation can be used to yield:

$$u_{M_u^n+1}^n = \mathcal{A}^n u_{M_u^n}^n + w_0^n - \mathcal{A}^n w_1^n, \quad (17a)$$

$$w_{-1}^n = -\mathcal{A}^n u_{M_u^n-1}^n + u_{M_u^n}^n + \mathcal{A}^n w_0^n. \quad (17b)$$

where

$$\mathcal{A}^n = \frac{\alpha^n - 1}{\alpha^n + 1}, \quad (18)$$

and

$$\alpha^n = N^n - N^n, \quad (19)$$

is the fractional part of N^n . Also see Figure ??.

2.2 Matrix Form

Similar to Sec. ??, system (??) can be written in matrix form. The state of the subsystems can be written in vector form according to $\mathbf{u} = [u_1^n, u_2^n, \dots, u_{M_u^n}^n]^T$ and $\mathbf{w} = [w_0^n, w_1^n, \dots, w_{M_w^n-1}^n]^T$ which are $M_u^n \times 1$ and $M_w^n \times 1$ respectively. The full state can then be written as $\mathcal{M}^n \times 1$ vector

$$\mathbf{u}^n = \begin{bmatrix} \mathbf{u}^n \\ \mathbf{w}^n \end{bmatrix}, \quad (20)$$

where $\mathcal{M}^n = M_u^n + M_w^n$. The update equation of the 1D wave equation including the dynamic grid can then be written in matrix form as

$$\mathbf{u}^{n+1} = (2\mathbf{I}_{\mathcal{M}^n} + (\lambda^n)^2 \mathcal{D}_{xx}^n) \mathbf{u}^n - \mathbf{u}^{n-1} \quad (21)$$

where $\lambda^n = c^n k / h^n$ (and is always 1 in the case of the 1D wave equation), and \mathcal{D}_{xx}^n is an adapted version of \mathbf{D}_{xx} in Eq. (??) to include the quadratic interpolation presented in Eq. (??):

$$\mathcal{D}_{xx}^n = \left[\begin{array}{ccc|ccc} \ddots & \ddots & & & & \mathbf{0} \\ & \ddots & -2 & 1 & & \\ & 1 & \mathcal{A}^n - 2 & 1 & -\mathcal{A}^n & \\ & -\mathcal{A}^n & 1 & \mathcal{A}^n - 2 & 1 & \\ & & & 1 & -2 & \ddots \\ \mathbf{0} & & & & \ddots & \ddots \end{array} \right] \quad (22)$$

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

2.3 Adding and Removing 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{u} or \mathbf{w} , or in an alternating fashion as in [?], but here, only changes in \mathbf{u} are considered. A grid point can be added to \mathbf{u} according to

$$\begin{aligned} \mathbf{u}^n &= [(\mathbf{u}^{n-1})^T, I_3^n \mathbf{z}^n]^T, \\ \mathbf{u}^{n-1} &= [(\mathbf{u}^{n-2})^T, I_3^{n-1} \mathbf{z}^{n-1}]^T, \end{aligned} \quad \text{if } N^n > N^{n-1} \quad (23)$$

where

$$\mathbf{z}^n = [u_{M_u^{n-1}-1}^n, u_{M_u^{n-1}}^n, w_0^n, w_1^n]^T, \quad \text{and}$$

$$\mathbf{z}^{n-1} = [u_{M_u^{n-2}-1}^{n-1}, u_{M_u^{n-2}}^{n-1}, w_0^{n-1}, w_1^{n-1}]^T.$$

Notice that I_3^n is used for adding a grid point to both \mathbf{u}^n and \mathbf{u}^{n-1} and that M^{n-1} is used as an index for both $u_{M_u^n}^n$ and $u_{M_u^n}^{n-1}$. Furthermore, cubic Lagrangian interpolator

$$I_3^n = \left[-\frac{\alpha^n(\alpha^n+1)}{(\alpha^n+2)(\alpha^n+3)}, \frac{2\alpha^n}{\alpha^n+2}, \frac{2}{\alpha^n+2}, -\frac{2\alpha^n}{(\alpha^n+3)(\alpha^n+2)} \right], \quad (24)$$

where α^n is as defined in Eq. (??). Notice that, as $\alpha^n \gtrsim 0$ the moment a grid point is added, $I_3^n \approx [0, 0, 1, 0]$ 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{u} according to

$$\begin{aligned} \mathbf{u}^n &= [u_1^n, \dots, u_{M^{n-1}-1}^n]^T, \\ \mathbf{u}^{n-1} &= [u_1^{n-1}, \dots, u_{M^{n-1}-1}^{n-1}]^T, \end{aligned} \quad \text{if } N^n < N^{n-1}. \quad (25)$$

The limit on changing grid configurations is the addition / removal of one grid point per sample. Practically, this limit needs to be much lower to keep the system well-behaved. In [?], a lower limit of 20 samples per change is used together with $\lambda = 0.999$ and yields no noticeable artefacts.

2.4 State Correction

At the moment that a point is removed, and thus $x_{u_{M_u}^n}^n \approx x_{w_0}^n$, the states of the inner boundaries might not be approximately equal, i.e., $u_{M_u}^n \not\approx w_0^n$. This violates the rigid connection in Eq. (??) and – in practice – causes audible artefacts. To reduce this, a method of *state correction*³ is introduced and adds an artificial spring force between the inner boundaries:

$$\begin{aligned} \delta_{tt} u_{lu}^n &= (c^n)^2 \delta_{xx} u_{lu}^n + J_u(x_{u_{M_u}^n}^n) F_c^n, \\ \delta_{tt} w_{lw}^n &= (c^n)^2 \delta_{xx} w_{lw}^n - J_w(x_{w_0}^n) F_c^n, \end{aligned}$$

where

$$\begin{aligned} J_u(x_i^n) &= \begin{cases} \frac{1}{h^n}, & l_u = \lfloor x_i^n / h^n \rfloor \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \\ J_w(x_i^n) &= \begin{cases} \frac{1}{h^n}, & l_w = \lfloor x_i^n / h^n \rfloor - M_u^n \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (26)$$

apply the correction force to the inner boundaries. Defining centred averaging and difference operators as

$$\mu_r q_l^n = \frac{1}{2} (q_l^{n+1} + q_l^{n-1}), \quad (27a)$$

$$\delta_r q_l^n = \frac{1}{2k} (q_l^{n+1} - q_l^{n-1}), \quad (27b)$$

the effect of the artificial spring can be defined as

$$F_c^n = \beta^n (\mu_r \eta^n + \sigma_{sc} \delta_r \eta^n), \quad (28)$$

with damping coefficient σ_{sc} and

$$\beta^n = \beta(\alpha^n) = \frac{1 - \alpha^n}{\alpha^n}. \quad (29)$$

Note that Eq. (??) is still defined for $\alpha^n = 0$ when solving for the connection force and acts as a rigid connection. See [?, Ch. 12] for a derivation. It is also important to note that despite the appearance of future grid values in Eq. (??), the correction force can be calculated explicitly.

2.5 Stiff String

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

$$\partial_t^2 q = c^2 \partial_x^2 q - \kappa^2 \partial_x^4 q, \quad (30)$$

³displacement correction in [?].

with stiffness coefficient κ (in m^2/s).

Equation (??) can be discretised using the following FD scheme:

$$\delta_{tt} q_l^n = c^2 \delta_{xx} q_l^n - \kappa^2 \delta_{xxxx} q_l^n, \quad (31)$$

where $\delta_{xxxx} = \delta_{xx} \delta_{xx}$. Note that the stability condition is now

$$h \geq \sqrt{\frac{c^2 k^2 + \sqrt{c^4 k^4 + 16 \kappa^2 k^2}}{2}} \quad (32)$$

Expanding δ_{tt} , and writing this scheme in matrix form yields

$$\mathbf{q}^{n+1} = (2\mathbf{I}_{N-1} + \lambda^2 \mathbf{D}_{xx} - \mu^2 \mathbf{D}_{xxxx}) \mathbf{q}^n - \mathbf{q}^{n-1}, \quad (33)$$

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

$$\mathbf{D}_{xxxx} = \mathbf{D}_{xx} \mathbf{D}_{xx}, \quad (34)$$

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

To apply the dynamic grid method to the stiff string, the definition of \mathbf{D}_{xx} in Eq. (??) can simply be replaced by the alternative matrix in Eq. (??) to obtain

$$\mathbf{D}_{xxxx}^n = \mathbf{D}_{xx}^n \mathbf{D}_{xx}^n, \quad (35)$$

and applied to the alternative vector \mathbf{u} in Eq. (??) according to

$$\mathbf{u}^{n+1} = (2\mathbf{I}_{M^n} + (\lambda^n)^2 \mathbf{D}_{xx}^n - (\mu^n)^2 \mathbf{D}_{xxxx}^n) \mathbf{u}^n - \mathbf{u}^{n-1}, \quad (36)$$

where $\mu^n = \kappa^n k / (h^n)^2$. Notice that κ^n is now allowed to be time-varying. Also note that damping terms have been excluded for brevity, but can be added trivially. Sound examples can be found via [?].

3 2D SYSTEMS

The framework presented in the previous section can be extended to higher-dimensional systems, such as membranes and plates. Before that, some theory on 2D systems in the context of FDTD methods will be presented.

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 sidelengths L_x and L_y (both in m). Discretising q will result in grid function $q_{l,m}^n$ where $l \in \{0, \dots, N_x\}$ and $m \in \{0, \dots, N_y\}$ and the number of intervals in the x 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 and y -direction.

Higher-dimensional systems can be written in matrix form by stacking or ‘flattening’ the state. For Dirichlet or simply supported boundary conditions the following $(N_x + N_y - 2) \times 1$ vector can be used to describe the state:

$$\begin{aligned} \mathbf{q}^n &= [(q_1^n)^T, \dots, (q_{N_x-1}^n)^T]^T \quad \text{with} \\ \mathbf{q}_l^n &= [q_{l,1}^n, \dots, q_{l,N_y-1}^n]^T. \end{aligned} \quad (37)$$

3.1 2D Dynamic Grid

Rather than splitting the original system into two subsystems, one can split it into four: u_{l_i, m_i}^n where $i \in \{1, \dots, 4\}$ is the number of the system (also see Figure ??). The systems 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: $l_i \in \{0, \dots, M_{x,i}^n\}$ and $m_i \in \{0, \dots, M_{y,i}^n\}$.

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 constraints apply to the number of intervals per subsystem $M_{x,1}^n = M_{x,3}^n, M_{x,2}^n = M_{x,4}^n, M_{y,1}^n = M_{y,2}^n, M_{y,3}^n = M_{y,4}^n$. 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 systems are positioned in the x - y -plane as follows:

$$(x_{u_{1,l_1,m_1}}^n, y_{u_{1,l_1,m_1}}^n) = (l_1 h^n, m_1 h^n), \quad (38a)$$

$$(x_{u_{2,l_2,m_2}}^n, y_{u_{2,l_2,m_2}}^n) = (L_x - (M_{x,2}^n - l_2) h^n, m_2 h^n), \quad (38b)$$

$$(x_{u_{3,l_3,m_3}}^n, y_{u_{3,l_3,m_3}}^n) = (l_3 h^n, L_y - (M_{y,3}^n - m_3) h^n), \quad (38c)$$

$$(x_{u_{4,l_4,m_4}}^n, y_{u_{4,l_4,m_4}}^n) = (L_x - (M_{x,4}^n - l_4) h^n, L_y - (M_{y,4}^n - m_4) h^n). \quad (38d)$$

Notice that, like in the 1D case, 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

$$\begin{aligned} l_1 &= \{1, M_{x,1}^n\}, & m_1 &= \{1, M_{y,1}^n\}, \\ l_2 &= \{0, M_{x,2}^n - 1\}, & m_2 &= \{1, M_{y,2}^n\}, \\ l_3 &= \{1, M_{x,3}^n\}, & m_3 &= \{0, M_{y,3}^n - 1\}, \\ l_4 &= \{0, M_{x,4}^n - 1\}, & m_4 &= \{0, M_{y,4}^n - 1\}. \end{aligned} \quad (39)$$

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 u_2 and u_4 , respectively.

The virtual grid points at the first vertical and horizontal inner boundary are then calculated through

$$u_{1,M_{x,1}^n+1,m_1}^n = \mathcal{A}_x^n u_{1,M_{x,1}^n,m_1}^n + u_{2,0,m_2}^n - \mathcal{A}_x^n u_{2,1,m_2}^n, \quad (40a)$$

$$u_{2,-1,m_2}^n = -\mathcal{A}_x^n u_{1,M_{x,1}^n-1,m_1}^n + u_{1,M_{x,1}^n,m_1}^n + \mathcal{A}_x^n u_{2,0,m_2}^n, \quad (40b)$$

$$u_{1,l_1,M_{y,1}^n+1}^n = \mathcal{A}_y^n u_{1,l_1,M_{y,1}^n}^n + u_{3,l_3,0}^n - \mathcal{A}_y^n u_{3,l_3,1}^n, \quad (40c)$$

$$u_{3,l_3,-1}^n = -\mathcal{A}_y^n u_{1,l_1,M_{y,1}^n-1}^n + u_{1,l_1,M_{y,1}^n}^n + \mathcal{A}_y^n u_{3,l_3,0}^n, \quad (40d)$$

respectively, and can be applied in the same manner to the second vertical and horizontal inner boundaries. [← possibly elaborate here](#) Here,

$$\mathcal{A}_x^n = \frac{\alpha_x^n - 1}{\alpha_x^n + 1}, \quad \text{and} \quad \mathcal{A}_y^n = \frac{\alpha_y^n - 1}{\alpha_y^n + 1}, \quad (41)$$

with

$$\alpha_x^n = \mathcal{N}_x^n - N_x^n, \quad \text{and} \quad \alpha_y^n = \mathcal{N}_y^n - N_y^n, \quad (42)$$

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

3.1.1 Matrix Form

Similar to Eq. (??), one can stack the total state of the system into a single column vector.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:

$$\mathbf{u}^n = [\mathbf{v}^n, \mathbf{w}^n]^T, \quad (43)$$

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

$$\begin{aligned} \mathbf{v}^n &= [(\mathbf{v}_1^n)^T, \dots, (\mathbf{v}_{M_{x,1}^n}^n)^T]^T, \quad \text{and} \\ \mathbf{w}^n &= [(\mathbf{w}_0^n)^T, \dots, (\mathbf{w}_{M_{x,2}^n-1}^n)^T]^T, \end{aligned} \quad (44)$$

and

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

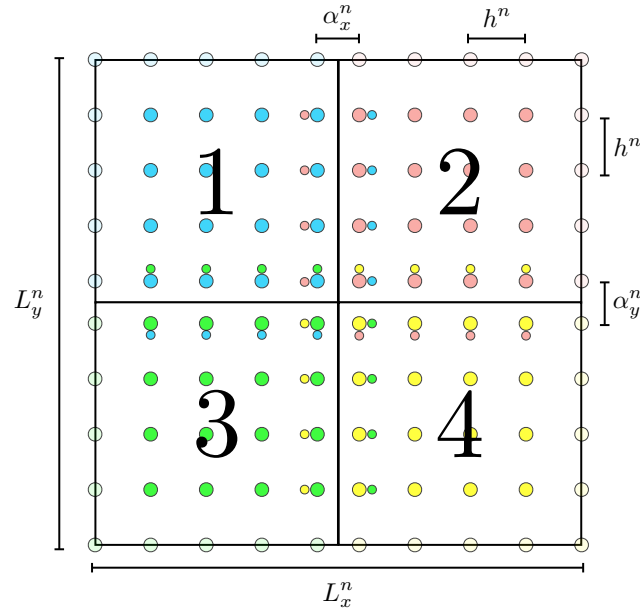


Fig. 2: Applying the dynamic grid to a 2D system. Virtual grid points are denoted as smaller circles and boundary points are not included in the calculation (for Dirichlet / simply supported boundary conditions).

⁴In terms of 'column of system no. #', the order will be: 1, 3, ..., 1, 3, 2, 4, ..., 2, 4.

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

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

3.1.2 Adding and Removing Points

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

Only considering alterations in the left systems, a column can be added to the system by carrying out the following operation on \mathbf{v} in Eq. (??)

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

where

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

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

Only considering alterations in the top systems, a row can be added by carrying out the following operation on \mathbf{u}_1 and \mathbf{u}_2 in Eq. (??) (done for both \mathbf{u}^n and \mathbf{u}^{n-1})

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

where

$$\begin{aligned} \mathbf{z}_{1,l_1}^n &= [u_{1,l_1,M_{y,1}^n-1}^n, u_{1,l_1,M_{y,1}^n}^n, u_{3,l_1,0}^n, u_{3,l_1,1}^n]^T, \\ \mathbf{z}_{2,l_2}^n &= [u_{2,l_2,M_{y,2}^n-1}^n, u_{2,l_2,M_{y,2}^n}^n, u_{4,l_2,0}^n, u_{4,l_2,1}^n]^T, \\ \mathbf{z}_{1,l_1}^{n-1} &= [u_{1,l_1,M_{y,1}^{n-1}-1}^{n-1}, u_{1,l_1,M_{y,1}^{n-1}}^{n-1}, u_{3,l_1,0}^{n-1}, u_{3,l_1,1}^{n-1}]^T, \quad \text{and} \\ \mathbf{z}_{2,l_2}^{n-1} &= [u_{2,l_2,M_{y,2}^{n-1}-1}^{n-1}, u_{2,l_2,M_{y,2}^{n-1}}^{n-1}, u_{4,l_2,0}^{n-1}, u_{4,l_2,1}^{n-1}]^T, \end{aligned}$$

contain the horizontal inner boundaries and their first neighbours in the y -direction (the ranges for l_1 and l_2 are given in Eq. (??)).

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

$$\begin{aligned} \mathbf{v}^n &= [(\mathbf{v}_1^n)^T, \dots, (\mathbf{v}_{M_{x,1}^n-1}^n)^T]^T \\ \mathbf{v}^{n-1} &= [(\mathbf{v}_1^{n-1})^T, \dots, (\mathbf{v}_{M_{x,1}^n-1}^{n-1})^T]^T \quad \text{if } N_x^n < N_x^{n-1} \end{aligned} \quad (47)$$

and removing a row from the system happens according to

$$\begin{aligned} \mathbf{u}_{1,l_1}^n &= [u_{1,l_1,1}^n, \dots, u_{1,l_1,M_{y,1}^n-1}^n]^T, \\ \mathbf{u}_{2,l_2}^n &= [u_{2,l_2,1}^n, \dots, u_{2,l_2,M_{y,1}^n-1}^n]^T, \\ \mathbf{u}_{1,l_1}^{n-1} &= [u_{1,l_1,1}^{n-1}, \dots, u_{1,l_1,M_{y,1}^n-1}^{n-1}]^T, \\ \mathbf{u}_{2,l_2}^{n-1} &= [u_{2,l_2,1}^{n-1}, \dots, u_{2,l_2,M_{y,1}^n-1}^{n-1}]^T, \end{aligned} \quad \text{if } N_y^n < N_y^{n-1}. \quad (48)$$

3.1.3 State correction

Like in the 1D case presented in Sec. ??, an artificial spring force can be added to the inner boundaries to prevent audible artefacts when removing grid points. If the state correction is applied to all grid points along the inner boundaries except for those at the intersection of the horizontal and vertical inner boundaries, (i.e., $u_{1,M_{x,1}^n,M_{y,1}^n}^n$, $u_{2,0,M_{y,2}^n}^n$, $u_{3,M_{x,3}^n,0}^n$ and $u_{4,0,0}^n$) the system can still be solved explicitly.

3.2 2D Wave Equation

The PDE of the 2D wave equation is defined as

$$\partial_t^2 q = c^2 \Delta q, \quad (49)$$

with wave speed c (in m/s) and Laplacian

$$\Delta = \partial_x^2 + \partial_y^2. \quad (50)$$

Equation (??) can be discretised to the following FD scheme:

$$\delta_{tt} q_{l,m}^n = c^2 \delta_{\Delta} q_{l,m}^n. \quad (51)$$

where,

$$\delta_{\Delta} = \delta_{xx} + \delta_{yy} \quad (52)$$

is the discrete Laplacian, and

$$\partial_x^2 q \approx \delta_{xx} q_{l,m}^n \triangleq \frac{1}{h^2} (q_{l+1,m}^n - 2q_{l,m}^n + q_{l-1,m}^n), \quad (53a)$$

$$\partial_y^2 q \approx \delta_{yy} q_{l,m}^n \triangleq \frac{1}{h^2} (q_{l,m+1}^n - 2q_{l,m}^n + q_{l,m-1}^n). \quad (53b)$$

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

3.2.1 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. (??), 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 in Eq. (??) by performing a Kronecker sum [?] (see Appendix):

$$\mathbf{D}_{\Delta} = \mathbf{D}_{yy} \oplus \mathbf{D}_{xx}, \quad (54)$$

and is of size $(N_x + N_y - 2) \times (N_x + N_y - 2)$. Using same-sized identity matrix $\mathbf{I} = \mathbf{I}_{N_x+N_y-2}$ and the stacked state vector in Eq. (??), the scheme in Eq. (??) can then be rewritten in matrix form as

$$\mathbf{q}^{n+1} = (2\mathbf{I} + \lambda^2 \mathbf{D}_{\Delta}) \mathbf{q}^n - \mathbf{q}^{n-1}. \quad (55)$$

3.2.2 Applying the Dynamic Grid

Using the state vector in Eq. (??) one can write the update equation of the 2D wave equation including the dynamic grid in matrix form as

$$\mathbf{u}^{n+1} = \left(2\mathbf{I}_{\mathcal{M}_2^n} + (\lambda^n)^2 \mathcal{D}_\Delta^n \right) \mathbf{u}^n - \mathbf{u}^{n-1} \quad (56)$$

with $\mathcal{M}_2^n \times \mathcal{M}_2^n$ matrix

$$\mathcal{D}_\Delta^n = \mathcal{D}_{yy}^n \oplus \mathcal{D}_{xx}^n. \quad (57)$$

\mathcal{D}_{xx}^n and \mathcal{D}_{yy}^n include the effect of the interpolation at the inner boundaries presented in Eq. (??) and are as defined in Eq. (??) with \mathcal{A}_x^n and \mathcal{A}_y^n as defined in Eq. (??) respectively.

3.3 Thin Plate

Similar to the stiff string presented in Sec. ??, one can extend the dynamic grid method to other systems using higher-order spatial derivatives.

Consider the PDE of a thin plate [?]

$$\partial_t^2 q = -\kappa^2 \Delta \Delta q \quad (58)$$

where κ is (again) a stiffness coefficient (in m^2/s). This can be discretised to

$$\delta_{ll} q_{l,m}^n = -\kappa^2 \delta_{\Delta\Delta} q_{l,m}^n \quad (59)$$

where $\delta_{\Delta\Delta} = \delta_\Delta \delta_\Delta$, and $h = 2\sqrt{\kappa k}$. Eq. (??) can be written in matrix form as

$$\mathbf{q}^{n+1} = \left(2\mathbf{I} - \mu^2 \mathbf{D}_{\Delta\Delta} \right) \mathbf{q}^n - \mathbf{q}^{n-1} \quad (60)$$

where $\mathbf{I} = \mathbf{I}_{N_x+N_y-2}$ and for simply supported boundary conditions

$$\mathbf{D}_{\Delta\Delta} = \mathbf{D}_\Delta \mathbf{D}_\Delta, \quad (61)$$

with \mathbf{D}_Δ as defined in Eq. (??).

The dynamic grid can be applied to this system similar to before as

$$\mathbf{u}^{n+1} = \left(2\mathbf{I}_{\mathcal{M}_2^n} - (\mu^n)^2 \mathcal{D}_{\Delta\Delta}^n \right) \mathbf{u}^n - \mathbf{u}^{n-1} \quad (62)$$

where for simply supported boundary conditions

$$\mathcal{D}_{\Delta\Delta}^n = \mathcal{D}_\Delta^n \mathcal{D}_\Delta^n, \quad (63)$$

and is of size $\mathcal{M}_2^n \times \mathcal{M}_2^n$. Sound examples can be found via [?].

4 ANALYSIS AND RESULTS

To evaluate the dynamic grid several analyses have been carried out.

4.1 Stability Analysis

Notes:

I can numerically show that for all dynamic grid update equations (Eqs. (??), (??), (??) and (??)) written in one-step form, the eigenvalues of matrix \mathcal{B}^n (see eq. (??)) are located on the unit circle for all values of α^n (given that the stability condition is satisfied). However, I can't algebraically prove this. I do know that this shows stability as well as the absence of damping in the system (right?). Should I mention this fact somewhere?

Alternatively, I can numerically show that the determinant of \mathcal{B}^n is 1, for any value of α^n . Hope that that helps to prove something...

4.2 Modal Frequencies

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, these are Eqs. (??), (??), (??) and (??). The p^{th} frequency can be calculated as

$$f_p^n = \frac{1}{2\pi k} \cos^{-1} \left(\frac{1}{2} \text{eig}_p(\mathcal{B}^n) \right), \quad (64)$$

where \mathcal{B}^n is the matrix multiplied onto \mathbf{u}^n in each respective update equation and $\text{eig}_p(\cdot)$ denotes the “ p^{th} eigenvalue of”.

To determine how accurate the frequency content is, the frequencies obtained in Eq. (??) can be compared to the modal frequencies that the implementation is expected to have. *not sure if I want to include the following, but just so you know how I did the analysis :) →* The expected modal frequencies of a FD scheme as a function of the wave number can be obtained by performing a von Neumann analysis on the discrete scheme. The wave number as a function of physical parameters can then be obtained through a combination of dispersion analysis and modal analysis of the continuous PDE.

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

- If $\alpha^n = 0$, the system exhibits identical behaviour to the original.
- The number of modes is always equal to the number of moving grid points in the system.
- The higher the modal number, the higher its frequency deviation.
- The higher the number of grid points in the simulation, the less frequency deviation occurs.
- The highest deviation amount for one value of N^n happens for $\alpha^n \lesssim 0.25$.

In the following, the difference between the expected modal frequencies and those exhibited by the implementation will be expressed in cents.

Table 1: Caption

	15 - 16	20 - 21	50 - 51
1D wave	-67.02	-54.19	-25.85
Bar	-96.00	-77.38	-36.71

4.2.1 1D Wave Equation

$$1\text{D wave } \mathcal{N}^n = 15 \rightarrow \mathcal{N}^n = 16$$

4.2.2 Stiff String

Cent deviation:

4.3 State Correction

In the 1D case, the state correction presented in Sec. ?? does not affect the modal content

Also, in the 1D case, if damping is added to the system the rate of change between grid configurations is low, (such that the higher modes are generally damped), the state correction is less necessary.

It has been found that excluding the state correction at the intersection of the horizontal and vertical inner boundaries has a minimal effect on the eventual behaviour of the system, and applying the state correction everywhere else already prevents audible artefacts.

5 DISCUSSION

Damping can easily be added to the systems presented in Sections ?? and ??.

Frequency deviations occur in higher frequency ranges and are thus much less perceptually relevant than

Notes:

Multiplying the average slope of the line that connects the (higher-frequency) minima of each deviation in Hz by the values of \mathcal{N} that causes these minima, yields lines that are bounded by the samplerate...

6 CONCLUSION

Not limited to two dimensions, could extend to 3D systems

Real-time implementation and control, such that a player can ‘mould’ their instrument while performing, potentially discovering new ways of expression with the instrument at hand.

At its current stage, physical accuracy and stability conditions are not the main focus of the method. Instead, efficiency and applicability to already existing models are more important for this contribution.

7 ACKNOWLEDGMENTS

This work has been funded in part by the European Art-Science-Technology Network for Digital Creativity (EASTN-DC). Thanks to the DAFx20in21 committee for [providing open access publication of this work](#).

APPENDIX: KRONECKER SUM

To obtain the matrix form of the discrete Laplacian in Eq. (??) the Kronecker product and Kronecker sum need to be introduced. The Kronecker product between two matrices is [?]

$$\mathbf{A}_{M \times N} \otimes \mathbf{B}_{K \times L} = \begin{bmatrix} a_{11}\mathbf{B} & \dots & a_{1N}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{M1}\mathbf{B} & \dots & a_{MN}\mathbf{B} \end{bmatrix}_{MK \times NL}, \quad (1)$$

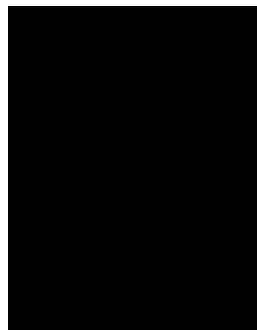
and the kronecker sum between two square matrices is [?]

$$\mathbf{A}_{M \times M} \oplus \mathbf{B}_{N \times N} = \mathbf{I}_N \otimes \mathbf{A} + \mathbf{B} \otimes \mathbf{I}_M. \quad (2)$$

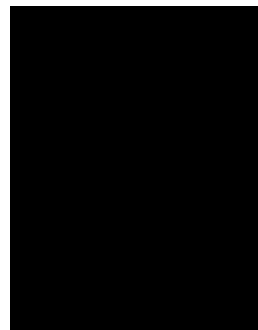
THE AUTHORS



Silvin Willemsen



Stefan Bilbao



Michele Ducceschi



Stefania Serafin

Silvin Willemsen is a Postdoctoral Researcher at Aalborg University in Copenhagen, Denmark. He received his MSc. in Sound and Music Computing from Aalborg University in 2017. In 2018, he was appointed as a PhD 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 PhD degree and continues to work in the field of physical modelling for musical instrument simulations.

●

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer pharetra massa nec erat elementum, non lacinia massa vehicula. Aenean lobortis pretium ex quis rutrum. Donec ut vehicula sem. Vestibulum sem sem, imperdiet in enim id, convallis porttitor velit. Curabitur elementum tincidunt diam. Donec pulvinar rutrum nisi, eu fringilla mauris gravida dignissim. Ut in nibh vehicula, congue lectus et, feugiat libero. Phasellus tempus turpis at lobortis condimentum. Proin egestas luctus aliquam. Cras conse-

quat, lorem sed facilisis accumsan, felis lacus facilisis lectus, at fringilla velit turpis et dui.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer pharetra massa nec erat elementum, non lacinia massa vehicula. Aenean lobortis pretium ex quis rutrum. Donec ut vehicula sem. Vestibulum sem sem, imperdiet in enim id, convallis porttitor velit. Curabitur elementum tincidunt diam. Donec pulvinar rutrum nisi, eu fringilla mauris gravida dignissim. Ut in nibh vehicula, congue lectus et, feugiat libero. Phasellus tempus turpis at lobortis condimentum. Proin egestas luctus aliquam. Cras consequat, lorem sed facilisis accumsan, felis lacus facilisis lectus, at fringilla velit turpis et dui.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer pharetra massa nec erat elementum, non lacinia massa vehicula. Aenean lobortis pretium ex quis rutrum. Donec ut vehicula sem. Vestibulum sem sem, imperdiet in enim id, convallis porttitor velit. Curabitur elementum tincidunt diam. Donec pulvinar rutrum nisi, eu fringilla mauris gravida dignissim. Ut in nibh vehicula, congue lectus et, feugiat libero. Phasellus tempus turpis at lobortis condimentum. Proin egestas luctus aliquam. Cras consequat, lorem sed facilisis accumsan, felis lacus facilisis lectus, at fringilla velit turpis et dui.