The Virtual Musical Instrument

Real-Time Physical Modelling of Musical Instruments using Finite-Difference Time-Domain Methods

Ph.D. Dissertation

Silvin Willemsen

Dissertation submitted April 28, 2021

Thesis submitted:	April 28, 2021
PhD Supervisor:	Prof. Stefania Serafin Aalborg University
PhD Committee:	Prof. X, Y University Prof. X, Y University Prof. X, Y University
PhD Series:	Technical Faculty of IT and Design, Aalborg University
Department:	Department of Architecture, Design and Media Technology

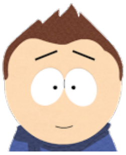
ISSN: xxxx-xxxx
ISBN: xxx-xx-xxxx-xxx-x

Published by:
Aalborg University Press
Skjernvej 4A, 2nd floor
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Rosendahls, 2021

Curriculum Vitae

Silvin Willemsen



Here is the CV text.

Acknowledgements

I would like to thank my mom..

List of Publications

Listed below are the publications that I (co)authored during the Ph.D. project grouped by: the main papers, which are also included in Part IX, papers where I had a supervisory role, and finally, miscellaneous publications.

Main Publications

[A] 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 Sound and Music Computing (SMC) Conference*, 2019, pp. 275–280.

[B] S. Willemsen, S. Bilbao, N. Andersson, and S. Serafin, "Physical models and real-time control with the sensel morph," in *Proceedings of the 16th Sound and Music Computing (SMC) Conference*, 2019, pp. 95–96.

[C] S. Willemsen, S. Bilbao, and S. Serafin, "Real-time implementation of an elasto-plastic friction model applied to stiff strings using finite difference schemes," in *Proceedings of the 22nd International Conference on Digital Audio Effects (DAFx-19)*, 2019, pp. 40–46.

[D] S. Willemsen, S. Serafin, S. Bilbao, and M. Ducceschi, "Real-time implementation of a physical model of the tromba marina," in *Proceedings of the 17th Sound and Music Computing (SMC) Conference*, 2020, pp. 161–168.

[E] S. Willemsen, R. Paisa, and S. Serafin, "Resurrecting the tromba marina: A bowed virtual reality instrument using haptic feedback and accurate physical modelling," in *Proceedings of the 17th Sound and Music Computing (SMC) Conference*, 2020, pp. 300–307.

[F] S. Willemsen, A.-S. Horvath, and M. Nascimben, "Digidrum: A haptic-based virtual reality musical instrument and a case study," in *Proceedings of the 17th Sound and Music Computing (SMC) Conference*, 2020, pp. 292–299.

[G] S. Willemsen, S. Bilbao, M. Ducceschi, and S. Serafin, "Dynamic grids for finite-difference schemes in musical instrument simulations," in

Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx2020in21), 2021.

[H] ——, "A physical model of the trombone using dynamic grids for finite-difference schemes," in *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx2020in21)*, 2021.

Publications with a Supervisory Role

[S1] R. S. Alecu, S. Serafin, S. Willemsen, E. Parravicini, and S. Lucato, "Embouchure interaction model for brass instruments," in *Proceedings of the 17th Sound and Music Computing (SMC) Conference*, 2020, pp. 153–160.

[S2] T. Lasickas, S. Willemsen, and S. Serafin, "Real-time implementation of the shamisen using finite difference schemes," in *Proceedings of the 18th Sound and Music Computing (SMC) Conference*, 2021.

[S3] C. Kaloumenou, S. Lambda, P. Pouliou, M. G. Onofrei, S. Willemsen, C. Jaller, and S. Serafin, "Recreating the amoeba violin using physical modelling and augmented reality," in *Proceedings of the 18th Sound and Music Computing (SMC) Conference*, 2021.

[S4] M. G. Onofrei, S. Willemsen, and S. Serafin, "Implementing complex physical models in real-time using partitioned convolution: An adjustable spring reverb," in *Proceedings of the 18th Sound and Music Computing (SMC) Conference*, 2021.

[S5] ——, "Real-time implementation of an elasto-plastic friction drum using finite difference schemes," in *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx2020in21)*, 2021.

Miscellaneous Publications

[M1] J. M. Hjerrild, S. Willemsen, and M. G. Christensen, "Physical models for fast estimation of guitar string, fret and plucking position," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 155–159, 2019.

[M2] K. Prawda, S. Willemsen, S. Serafin, and V. Välimäki, "Flexible real-time reverberation synthesis with accurate parameter control," in *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx-20)*, 2020.

[M3] M. Ducceschi, S. Bilbao, S. Willemsen, and S. Serafin, "Linearly-implicit schemes for collisions in musical acoustics based on energy quadratisation," *Journal of the Acoustical Society of America (JASA)*, 2021.

Abstract

English abstract

Abstract

Resumé

Danish Abstract

Contents

Todo list

Preface

Starting this Ph.D. project, I did not have a background in mathematics, physics or computer science, which were three equally crucial components in creating the result of this project. After the initial steep learning curve of notation and terminology, I was surprised to find that the methods used for physical modelling are actually quite straightforward!

Of course it should take a bit of time to learn these things, but

Many concepts that seemed impossible at the beginning

I feel that the literature lacks a lot of the intuition needed for readers without a background in any of these topics. Rather, much of the literature I came across assumes that the reader has a degree in at least one of the aforementioned topics. This is why I decided to write this work a bit more pedagogical than what could be expected.

I believe that anyone with some basic skills in mathematics and programming is able to create a simulation based on physics within a short amount of time, given the right tools, which I hope that this dissertation could be.

The knowledge dissemination of this dissertation is thus not only limited to the research done and publications made over the course of the project, but also its pedagogical nature hopefully allowing future (or past) students to benefit from.

As with a musical instrument itself, a low entry level, a gentle learning curve along with a high virtuosity level is desired. Take a piano, for instance. Most will be able to learn a simple melody — such as "Frère Jacques" — in minutes, but to become virtuous requires years of practice.

This is the way I wanted to write this dissertation: easy to understand the basic concepts, but many different aspects touched upon to allow for virtuosity. Hopefully by the end, the reader will at least grasp some highly complex concepts in the fields of mathematics, physics and computer science (which will hopefully take less time than it takes to become virtuous at playing the piano).

Some basic calculus knowledge is assumed.

I wanted to show my learning process and (hopefully) explain topics such as *Energy Analysis*, *Stability Analysis*, etc. in a way that others lacking the same knowledge will be able to understand.

Make physical modelling more accessible to the non-physicist.

Interested in physically impossible manipulations of now-virtual instruments.

Silvin Willemsen
Aalborg University, April 28, 2021

Part I

Introduction

Chapter 1

Physical Modelling of Musical Instruments

The history of physical modelling of musical instruments

Exciter-resonator approach.

The time-evolution of dynamic systems can be conveniently described by differential equations. Examples of a dynamic systems are a guitar string, a drum-membrane, or a concert hall; three very different concepts, but all based on the same types of equations of motion.

Though these equations are very powerful, only few have a closed-form solution. What this means is that in order for them to be implemented, they need to be approximated. There exist different approximation techniques to do this

1.1 Physical Modelling Techniques

- Modal Synthesis
- Finite-difference Time-domain methods
- Finite Element Methods
- Digital waveguides
- Mass-spring systems
- Functional transformation method
- State-space
- Wave-domain

- Energy-based

Advantages of finite-difference methods
Using FDTD methods can be quite computationally heavy. Moore's law [9]

1.2 Real-Time Implementation

Although many techniques to digitally simulate musical instruments exist proving that we have only recently reached the computing power in personal computers to make real-time playability of these models an option. The biggest challenge in real-time audio applications as opposed to those only involving graphics, is that the sample rate is extremely high. As Nyquist's sampling theory tells us, a sampling rate of at least 40 kHz is necessary to produce frequencies up to the human hearing limit of 20 kHz **[Nyquist]**. Visuals

Real-time: no noticable latency

exactly as in [2]

1.3 Why?

1.3.1 Audio plugins

Samples, or recordings, of real instruments are static and unable to adapt to changes in perfomance. Moreover, capturing the the entire interaction space of an instrument is nearly impossible. Imagine recording a violin with every single combination of bowing force, velocity, position, duration and other aspects such as vibrato, pizzicato. Even if a complete sample library could be created, this would contain an immense amount of data.

Samples vs. Physical Modelling:

Trade off between storage and speed

Using musical instrument simulations, on the other hand, allows the sound to be generated on the spot based on physical parameters that the user can interact with.

1.3.2 Resurrect old or rare instruments

Even popular instruments require maintenance and might need to be replaced after years of usage.

1.3.3 Go beyond what is physically possible

1.4 Thesis Objectives and Main Contributions

Over the past few decades, much work has been done on the accurate modelling of physical phenomena. In the field of sound and musical instruments..

From [5] to [4]

The main objective of this thesis is to implement existing physical models in real time using FDTD methods. Many of the physical models and methods presented in this thesis are taken from the literature and are thus not novel.

Secondly, to combine the existing physical models to get complete instruments and be able to control them in real time.

As FDTD methods are quite rigid, changing parameters on the fly, i.e., while the instrument simulation is running, is a challenge. Other techniques, such as modal synthesis, are much more suitable for this, but come with the drawbacks mentioned in Section 1.1. Therefore, a novel method was devised to smoothly change parameters over time, introducing this to FDTD methods.

1.5 Thesis Outline

- Physical models
  - Resonators
  - Exciters
  - Interactions
- Dynamic Grids
- Real-Time Implementation and Control
- Complete instruments
  - Large-scale physical models
  - Tromba Marina
  - Trombone

Notes

- Think about how to define real-time.
- Create an intuition for different parts of the equation
- Talk about input and output locations and how that affects frequency content (modes).

One over number $\rightarrow$ reciprocal of number

Example: When the waveform consists entirely of harmonically related frequencies, it will be periodic, with a period equal to the reciprocal of the fundamental frequency (from An Introduction to the Mathematics of Digital Signal Processing Pt 2 by F. R. Moore)

Chapter 2

FDTD Methods and Analysis Techniques

"Since Newton, mankind has come to realize that the laws of physics are always expressed in the language of differential equations."
- Steven Strogatz
(https://youtu.be/O85OWBJ2ayo?t=44)

This chapter introduces some important concepts needed to understand the physical models presented later on in this document. By means of a simple mass-spring system and the 1D wave equation, the notation (and terminology) used throughout this document will be explained, together with some important analysis techniques. Before we dive into the mathematics, let us go over some useful terminology.

Differential equations

As mentioned in Chapter 1 differential equations are used to describe the motion of dynamic systems. These systems can describe anything from the transverse displacement of a string to the air pressure in an acoustic tube.

A characteristic feature of these equations is that, rather than an absolute value or *state* of a system, the time derivative of its state – its velocity – or the second-order time derivative – its acceleration – is described. From this, the absolute state of the system can be computed.

This state is usually described by the variable u which is (nearly) always a function of time, i.e., $u = u(t)$. If the system is distributed in space, u also becomes a function of space, i.e., $u = u(x, t)$, or with two spatial dimensions, $u = u(x, y, t)$, etc. Though this work only describes systems of up to two spatial dimensions, one could potentially extend to systems of infinite spatial dimensions evolving over time!

If u is only a function of time, the differential equation that describes the motion of this system is called an *ordinary differential equation* (ODE). If u is also a function of at least one spatial dimension, the equation of motion is a called a *partial differential equation* (PDE).

The literature uses different types of notation for taking (continuous-time) partial derivatives. Applied to a state variable u these can look like

$$\frac{\partial^2 u}{\partial t^2} \quad \text{(classical notation)}$$
$$u_{tt} \quad \text{(subscript notation)}$$
$$\partial_t^2 u \quad \text{(operator notation)}$$

all of which mean a second-order derivative with respect to time t, i.e., u's acceleration. In this remainder of this document, the operator notation will be used. Often-used partial derivatives and their interpretation are shown below

maybe not yet as this is super general still

$$\partial_t^2 u \quad \text{(acceleration)} \qquad \partial_x^2 u \quad \text{(curvature)}$$
$$\partial_t u \quad \text{(velocity)} \qquad \partial_x u \quad \text{(slope)}$$

Note: difference between 1D, 2D spatial, and 1D, 2D displacement (polarisation). Transverse displacement of 1D wave here

Now that an equation has been established, how

Discretisation using FDTD methods

Finite-difference time-domain (FDTD) methods essentially subdivide a continuous equation in discrete points in time and space, a process called *discretisation*. Once an ODE or PDE is discretised using these methods it is now called a *Finite-Difference (FD) Scheme*.

We start by defining a discrete *grid* over time and space which we will use to approximate our continuous equations . A system $u = u(x,t)$ defined over time t and one spatial dimension x, can be approximated using a *grid function* u_l^n. Here, subscript l superscript n describe the spatial and temporal indices respectively and arise from the discretisation of the continuous variables x and t according to $x = lh$ and $t = nk$. The spatial step h, also called the *grid spacing* describes the distance (in m) between two neighbouring grid points and the temporal step k, or *time step* is the time (in s) between two consecutive temporal indices. The latter can be calculated $k = 1/f_\text{s}$ for a sample rate f_s (in Hz). In many audio applications $f_\text{s} = 44100$ Hz which will be used in this work (unless denoted otherwise).

grid figure

To summarise

$$u(x,t) \approx u_l^n \quad \text{with} \quad x = lh \quad \text{and} \quad t = nk \tag{2.1}$$

Now that the state variable has a discrete counterpart, this leaves the derivatives to be discretised. We start by introducing shift operators that can be applied to a grid function and changes its indexing (either spatial or temporal). Forward and backward shifts in time, together with the identity operation (not altering u_l^n) are

$$e_{t+}u_l^n = u_l^{n+1}, \quad e_{t-}u_l^n = u_l^{n-1}, \quad \text{and} \quad 1u_l^n = u_l^n. \tag{2.2}$$

Similarly, forward and backward shifts in space are

$$e_{x+}u_l^n = u_{l+1}^n, \quad \text{and} \quad e_{x-}u_l^n = u_{l-1}^n. \tag{2.3}$$

These shift operators are rarely used in isolation, though they do appear in energy analysis techniques detailed in Section 2.3. The operators do, however, form the basis of commonly used FD operators. The first-order derivative in time can be discretised three different ways. The forward, backward and centred difference are

check centred instead of centered, full document sweep

$$\partial_t \approx \begin{cases} \delta_{t+} \triangleq \frac{1}{k}\left(e_{t+} - 1\right), & (2.4\text{a})\\ \delta_{t-} \triangleq \frac{1}{k}\left(1 - e_{t-}\right), & (2.4\text{b})\\ \delta_{t\cdot} \triangleq \frac{1}{2k}\left(e_{t+} - e_{t-}\right), & (2.4\text{c}) \end{cases}$$

where "$\triangleq$" means "equal to by definition". These operators can then be applied to grid function u_l^n to get

$$\partial_t u \approx \begin{cases} \delta_{t+}u_l^n = \frac{1}{k}\left(u_l^{n+1} - u_l^n\right), & (2.5\text{a})\\ \delta_{t-}u_l^n = \frac{1}{k}\left(u_l^n - u_l^{n-1}\right), & (2.5\text{b})\\ \delta_{t\cdot}u_l^n = \frac{1}{2k}\left(u_l^{n+1} - u_l^{n-1}\right). & (2.5\text{c}) \end{cases}$$

Note that the centred difference has a division by $2k$ as the time difference between $n+1$ and $n-1$ is twice the time step.

The same can be done in for a first-order derivative in space, where the forward, backward and centred difference are

figure here for interpretation of accuracy of operators

$$\partial_x \approx \begin{cases} \delta_{x+} \triangleq \frac{1}{h}\left(e_{x+} - 1\right), & (2.6\text{a})\\ \delta_{x-} \triangleq \frac{1}{h}\left(1 - e_{x-}\right), & (2.6\text{b})\\ \delta_{x\cdot} \triangleq \frac{1}{2h}\left(e_{x+} - e_{x-}\right), & (2.6\text{c}) \end{cases}$$

and when applied to u_l^n are

$$\partial_x u \approx \begin{cases} \delta_{x+}u_l^n = \frac{1}{h}\left(u_{l+1}^n - u_l^n\right), & (2.7\text{a})\\ \delta_{x-}u_l^n = \frac{1}{h}\left(u_l^n - u_{l-1}^n\right), & (2.7\text{b})\\ \delta_{x\cdot}u_l^n = \frac{1}{2h}\left(u_{l+1}^n - u_{l-1}^n\right). & (2.7\text{c}) \end{cases}$$

Higher order differences can be obtained from first-order difference operators. The second-order difference in time may be approximated using

$$\partial_t^2 \approx \delta_{t+}\delta_{t-} = \delta_{tt} = \frac{1}{k^2}\left(e_{t+} - 2 + e_{t-}\right), \tag{2.8}$$

where "2" is the identity operator applied twice. and similarly for the second-order difference in space

$$\partial_x^2 \approx \delta_{x+}\delta_{x-} = \delta_{xx} = \frac{1}{h^2}\left(e_{x+} - 2 + e_{x-}\right). \tag{2.9}$$

More explanation about combining operators can be found in Section 3.2.1.

Averaging operators

Operators and derivatives in 2D will be discussed in Chapter 4.

Accuracy

Taylor series expansion...

2.0.1 Identities

Extremely useful are

An identity is something that holds true

$$\delta_{tt}u_l^n = \frac{2}{k}\left(\delta_{t\cdot}u_l^n - \delta_{t-}u_l^n\right) \tag{2.10a}$$

$$\delta_{tt} \tag{2.10b}$$

That these identities hold can easily be proven by expanding the operators

Implementation

In the following

- Continuous-time
- Discrete-time
- Implementation (update equation)

Something something newton's second law

initial conditions

It is important to note that a discrete FD scheme is an *approximation* to a continuous PDE, not a sampled version of it. This means that the resulting schemes are rarely an exact solution to the original continuous equation.

2.1 The Mass-Spring System

Though a complete physical modelling field on its own (see Chapter 1), mass-spring systems are also sound-generating systems themselves.

Continuous-time

The ODE of a simple mass-spring system is defined as

$$\frac{d^2u}{dt^2} = -\omega_0^2 u \tag{2.11}$$

Discrete-time

The , u is approximated using

$$u(t) \approx u^n \tag{2.12}$$

where $t = nk$

Implementation

2.2 The 1D Wave Equation

Arguably the most important PDE in the field of physical modelling for sound synthesis is the 1D wave equation.

2.2.1 Continuous-time

The state of the system $u = u(x,t)$ meaning that is on top of being defined in time t it is distributed over space x. The 1D wave equation is defined as follows

$$\partial_t^2 u = c^2 \partial_x^2 u, \tag{2.13}$$

where c is the wave speed (in m/s). The state variable u can be used to describe transverse vibration in an ideal string, longitudinal vibration in an ideal bar and pressure in an acoustic tube. Although the behaviour of this equation alone does not appear in the real world, as no physical system is ideal, it is extremely useful as a test case and a basis for more complicated models. Here, it will be used to introduce various concepts and analysis techniques in the field of FDTD methods.

Intuition

Although the 1D wave equation often appears in the literature, an intuition or interpretation of why it works the way it does is hard to find. In the following, u describes the transverse displacement of an ideal string.

Going back to Newton's second law

The acceleration of $u = u(x,t)$ at location x is determined by the second-order spatial derivative of u at that same location (scaled by a constant c^2). In the case that u describes the transverse displacement of an ideal string, this second-order derivative denotes the *curvature* of this string. As c^2 is always positive, the sign (or direction) of the acceleration is fully determined by the sign of the curvature. In other words, a 'positive' curvature at location x along the ideal string yields a 'positive' or upwards acceleration at that same location.

What a 'positive' or 'negative' curvature implies is more easily seen when we take a simple function describing a parabola, $y(x) = x^2$, and take its second derivative to get $y''(x) = 2$. The answer is a positive number which means that y has a positive curvature.

So, what does this mean for the 1D wave equation? As a positive curvature implies a positive or upwards acceleration as per Eq. (2.13), u with a positive curvature at a location x will start to move upwards and vice versa. Of course, the state of a physical system such as u will rarely have a perfect parabolic shape, but the argument still applies.

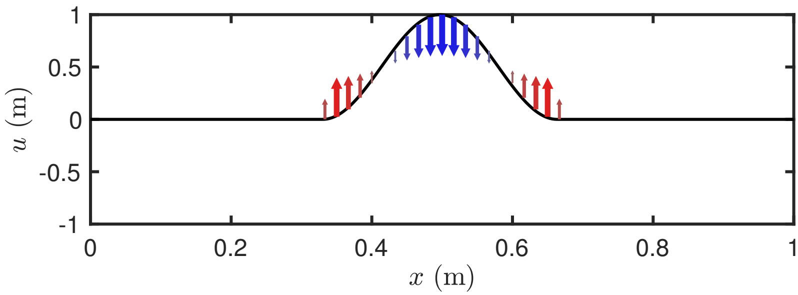


Fig. 2.1: The forces acting on the 1D wave equation initialised with a raised cosine at $x = 0.5$. Arrows indicate the direction and magnitude of the force.

Boundary Conditions

When a system is distributed in space,

For analysis purposes, an infinite domain may be adopted, but for implementation, a finite domain needs to be established.

Consider a system of length L (in m) $x \in \mathcal{D}$ (x is an element in $\mathcal{D}$) where domain $\mathcal{D} = [0, L]$ Two alternatives are

$$u(0,t) = u(L,t) = 0 \quad \text{(Dirichlet, fixed)}, \tag{2.14a}$$
$$\partial_x u(0,t) = \partial_x u(L,t) = 0 \quad \text{(Neumann, free)}, \tag{2.14b}$$

2.2.2 Discrete-time

The most straightforward discretisation of Eq. (2.13) is the following FD scheme

$$\delta_{tt} u_l^n = c^2 \delta_{xx} u_l^n. \tag{2.15}$$

Other schemes exist (such as implicit)

Stability

The parameters that define (2.19) are linked through a stability condition. The famous Courant-Friedrichs-Lewy or *CFL condition* is defined as [**?**]

$$\lambda \leq 1 \quad \text{where} \quad \lambda = \frac{ck}{h} \tag{2.16}$$

where

$$\lambda = \frac{ck}{h} \tag{2.17}$$

is referred to as the *Courant number*.

Regarding implementation, as the time step k is based on the sample rate and thus usually fixed and c is a user-defined wavespeed, it is useful to rewrite Eq. (2.16) to be in terms of the grid spacing h:

$$h \geq ck. \tag{2.18}$$

Implementation

$$u_l^{n+1} = 2u_l^n - u_l^{n-1} + \lambda^2 \left(u_{l+1}^n - 2u_l^n + u_{l-1}^n \right) \tag{2.19}$$

If $\lambda = 1$, (2.19) is an exact solution to Eq. (2.13). This is no

Matrix form

2.2.3 Operators in Matrix Form

Finite-difference operators, such as δ_{x+}, δ_{x-} and $\delta_{x\cdot}$ can be written in matrix form:

$$\mathbf{D}_{x+} = \frac{1}{h}\begin{bmatrix} \ddots & \ddots & & & \mathbf{0} & \\ & -1 & 1 & & & \\ & & -1 & 1 & & \\ & & & -1 & 1 & \\ & & & & -1 & \ddots \\ & \mathbf{0} & & & & \ddots \end{bmatrix} \qquad \mathbf{D}_{x-} = \frac{1}{h}\begin{bmatrix} \ddots & & & & \mathbf{0} & \\ \ddots & 1 & & & & \\ & -1 & 1 & & & \\ & & -1 & 1 & & \\ & & & -1 & 1 & \\ & \mathbf{0} & & & \ddots & \ddots \end{bmatrix}$$

$$\mathbf{D}_{x\cdot} = \frac{1}{2h}\begin{bmatrix} \ddots & \ddots & & & \mathbf{0} & \\ \ddots & 0 & 1 & & & \\ & -1 & 0 & 1 & & \\ & & -1 & 0 & 1 & \\ & & & -1 & 0 & \ddots \\ & \mathbf{0} & & & \ddots & \ddots \end{bmatrix}$$

The matrices $\mathbf{D}_{x+}$ and $\mathbf{D}_{x-}$ can be multiplied to get $\mathbf{D}_{xx}$:

$$\mathbf{D}_{xx} = \mathbf{D}_{x+}\mathbf{D}_{x-} = \frac{1}{h^2}\begin{bmatrix} \ddots & \ddots & & & \mathbf{0} & \\ \ddots & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & 1 & -2 & 1 & \\ & & & 1 & -2 & \ddots \\ & \mathbf{0} & & & \ddots & \ddots \end{bmatrix} \tag{2.20}$$

and two $\mathbf{D}_{xx}$'s to get

$$\mathbf{D}_{xx}\mathbf{D}_{xx} = \mathbf{D}_{xxxx} = \frac{1}{h^4}\begin{bmatrix} 5 & -4 & 1 & & & \mathbf{0} & \\ -4 & 6 & \ddots & \ddots & & & \\ 1 & \ddots & \ddots & -4 & 1 & & \\ & \ddots & -4 & 6 & -4 & \ddots & \\ & & 1 & -4 & \ddots & \ddots & 1 \\ & & & \ddots & \ddots & 6 & -4 \\ & \mathbf{0} & & & 1 & -4 & 5 \end{bmatrix} \tag{2.21}$$

which is used for a stiff string with a simply supported boundary condition.

Averaging operators μ_{x+}, μ_{x-} and $\mu x \cdot$ are defined in a similar way:

$$\mathbf{M}_{x+} = \frac{1}{2}\begin{bmatrix} \ddots & \ddots & & & \mathbf{0} & \\ & 1 & 1 & & & \\ & & 1 & 1 & & \\ & & & 1 & 1 & \\ & & & & 1 & \ddots \\ & \mathbf{0} & & & & \ddots \end{bmatrix} \qquad \mathbf{M}_{x-} = \frac{1}{2}\begin{bmatrix} \ddots & & & & \mathbf{0} & \\ \ddots & 1 & & & & \\ & 1 & 1 & & & \\ & & 1 & 1 & & \\ & & & 1 & 1 & \\ & \mathbf{0} & & & \ddots & \ddots \end{bmatrix}$$

$$\mathbf{M}_{x\cdot} = \frac{1}{2}\begin{bmatrix} \ddots & \ddots & & & \mathbf{0} & \\ \ddots & 0 & 1 & & & \\ & 1 & 0 & 1 & & \\ & & 1 & 0 & 1 & \\ & & & 1 & 0 & \ddots \\ & \mathbf{0} & & & \ddots & \ddots \end{bmatrix}$$

Note the multiplication by $1/2$ rather than $1/h$ (or $1/2h$) for all operators.

Only spatial operators are written in this matrix form and then applied to state vectors at different time steps ($n+1$, n and $n-1$), examples of which can be found below.

Output sound

After the system is excited (see III), one can listen to the output

2.3 Energy Analysis

Debugging physical models.

2.3.1 Mathematical tools

Continuous-time

For two functions $f(x)$ and $g(x)$ and $x \in \mathcal{D}$ their l_2 inner product along with the l_2 norm is defined as

$$\langle f, g\rangle_{\mathcal{D}} \int_{\mathcal{D}} fg dx \quad \text{and} \quad \|f\|_{\mathcal{D}} = \sqrt{\langle f, f\rangle_{\mathcal{D}}} \tag{2.22}$$

Discrete-time

Inner product of any time series f^n and g^n and the discrete counterpart to (2.22) is

$$\langle f^n, g^n \rangle_{\mathcal{D}} = \sum_{l\in\mathcal{D}} h f_l^n g_l^n \tag{2.23}$$

where the multiplication by h is the discrete counterpart of dx the continuous definition.

2.4 von Neumann Analysis

Much literature gives the stability condition using an "it can be shown that" argument. Here, I would like to take the opportunity to

Finding stability conditions for models

Sin identity:

$$\sin(x) = \frac{e^{jx} - e^{-jx}}{2j} \quad \Longrightarrow \quad \sin^2(x) = \frac{e^{j2x} - 2e^{jx-jx} + e^{-j2x}}{-4} = \frac{e^{j2x} + e^{-j2x}}{-4} + \frac{1}{2}. \tag{2.24}$$

Cos identity:

$$\cos(x) = \frac{e^{jx} + e^{-jx}}{2} \quad \Longrightarrow \quad \cos^2(x) = \frac{e^{j2x} + 2e^{jx-jx} + e^{-j2x}}{4} = \frac{e^{j2x} + e^{-j2x}}{4} + \frac{1}{2}. \tag{2.25}$$

2.5 Modal Analysis

This section will show how to obtain the modes for an FD scheme implementing the 1D wave equation as done in Section . We start with Eq. (6.34) (using c instead of γ)

$$\delta_{tt} u = c^2 \delta_{xx} u, \tag{2.26}$$

which can be written in matrix form as

$$\frac{1}{k^2}\left(\mathbf{u}^{n+1} - 2\mathbf{u}^n + \mathbf{u}^{n-1}\right) = c^2 \mathbf{D}_{xx}\mathbf{u} \tag{2.27}$$

Following [2] we assume a solution of the form $\mathbf{u} = \boldsymbol{\phi} z^n$. Substituting this into Eq. (2.27) yields the characteristic equation

$$(z - 2 + z^{-1})\boldsymbol{\phi} = c^2 k^2 \mathbf{D}_{xx}\boldsymbol{\phi}. \tag{2.28}$$

This is an eigenvalue problem where the p'th solution is defined as

$$\begin{aligned} z_p - 2 + z_p^{-1} &= c^2 k^2 \text{eig}_p(\mathbf{D}_{xx}) \\ z_p + (-2 - c^2 k^2 \text{eig}_p(\mathbf{D}_{xx})) + z_p^{-1} &= 0 \end{aligned} \tag{2.29}$$

where $\text{eig}_p(\cdot)$ denoting the pth eigenvalue of '$\cdot$'. If the CFL condition for the scheme is satisfied, the roots will lie on the unit circle. Furthermore we can substitute a test solution $z_p = e^{j\omega_p k}$ solve for the eigenfrequencies:

$$
\begin{aligned}
e^{j\omega_p k} + e^{-j\omega_p k} - 2 - c^2k^2\text{eig}_p(\mathbf{D}_{xx}) &= 0 \\
\frac{e^{j\omega_p k} + e^{-j\omega_p k}}{-4} + \frac{1}{2} + \frac{c^2k^2}{4}\text{eig}_p(\mathbf{D}_{xx}) &= 0
\end{aligned}
$$

Then using Eq. (2.24) we get

$$
\begin{aligned}
\sin^2(\omega_p k/2) + c^2k^2\text{eig}_p(\mathbf{D}_{xx}) = 0 \\
\sin(\omega_p k/2) = ck\sqrt{-\text{eig}_p(\mathbf{D}_{xx})} \\
\omega_p = \frac{2}{k}\sin^{-1}\left(ck\sqrt{-\text{eig}_p(\mathbf{D}_{xx})}\right)
\end{aligned}
\tag{2.30}
$$

which is Eq. (6.53) in [2].

2.6 Dispersion analysis

Part II

Resonators

Resonators

Though the physical models described in the previous part are also considered resonators, they are *ideal* cases. In other words, you would not be able to find these “in the wild" as they do not includes effects such as losses or dispersion in the case of the 1D wave equation.

Scaling

Scaling, or non-dimensionalisation can be useful to reduce the parameter space

Domain $x \in [0, L]$ is scaled to $x' = x/L$ such that $x' \in [0, 1]$.

The 1D wave equation in (2.13) can be rewritten to

$$\partial_t^2 u = \gamma^2 \partial_{x'x'} u \tag{2.31}$$

where scaled wavespeed $\gamma = c/L$ has units of frequency.

Although this parameter reduction might be useful for resonators in isolation, when they are connected

Moreover, for the parameters to make

As, later on, different resonators in isolation will be connected, the overview is better kept when all parameters are written out

- Bars and Stiff Strings
- 2D Models
- Brass

Chapter 3

Stiff string

Consider the transverse displacement of a string of length L described by $u = u(x,t)$ defined for $x \in \mathcal{D}$ with domain $\mathcal{D} = [0, L]$ and time $t \geq 0$.

Stiff string and stuff Used in many of the publications in Part IX

3.1 Continuous time

Circular cross-sectional area

$$\rho A \partial_t^2 u = T \partial_x^2 u - EI \partial_x^4 u \tag{3.1}$$

parameterised by material density ρ (in kg/m^3), cross-sectional area $A = \pi r^2$ (in m^2), radius r (in m) tension T (in N), Young's modulus E (in Pa) and area moment of inertia $I = \pi r^4/4$ (in m^4). If either E or I is 0, Eq (3.1) reduces to the 1D wave equation in (2.13) where $c = \sqrt{T/\rho A}$. If instead $T = 0$, Eq. (3.1) reduces to the ideal bar equation.

The 4th-order spatial derivative models *frequency dispersion* a phenomenon that causes different frequencies to travel at different speeds.

3.1.1 Dispersion Analysis

Adding Losses

Before moving on to the discretisation of Eq. (3.1), losses are added

First appeared in [1]

$$\rho A \partial_t^2 u = T \partial_x^2 u - EI \partial_x^4 u - 2\sigma_0 \rho A \partial_t u + 2\sigma_1 \rho A \partial_t \partial_x^2 u \tag{3.2}$$

where the loss coefficients σ_0 and σ_1 describe the frequency dependent and frequency independent losses respectively.

A more compact way to write Eq. (3.4), and as is also found often in the literature [2] is to divide both sides by ρA to get

etc.

$$\partial_t^2 u = c^2 \partial_x^2 u - \kappa^2 \partial_x^4 u - 2\sigma_0 \partial_t u + 2\sigma_1 \partial_t \partial_x^2 u \tag{3.3}$$

check wavespeed or wave speed (entire document)

where $c = \sqrt{T/\rho A}$ is the wave speed (in m/s) as in the 1D wave equation in (2.13) and $\kappa = \sqrt{EI/\rho A}$ is a *stiffness coefficient* (in m^2/s).

Intuition

Although Eq. (3.4) might look daunting at first, the principle of Newton's second law remains the same.

Something about the 4th spatial derivative and the loss terms here...

Boundary Conditions

The boundary conditions found in (2.14) can be extended

3.2 Discrete Time

Equation (3.4) can be discretised as

$$\rho A \delta_{tt} u_l^n = T\delta_{xx} u_l^n - EI\delta_{xxxx} u_l^n - 2\sigma_0 \rho A \delta_{t\cdot} u_l^n + 2\sigma_1 \rho A \delta_{t-}\delta_{xx} u_l^n \tag{3.4}$$

The δ_{xxxx} operator is the the second-order spatial difference in Eq. (2.9) applied to itself.

$$\delta_{xxxx} = \delta_{xx}\delta_{xx} = \frac{1}{h^4}\left(e_{x+}^2 - 4e_{x+} + 6 - 4e_{x-} + e_{x-}^2\right) \tag{3.5}$$

3.2.1 Combining operators

$$\delta_{t-}\delta_{xx} = \frac{1}{k}\left(1 - e_{t-}\right)\frac{1}{h^2}\left(e_{x+} - 2 + e_{x-}\right) = \frac{1}{kh^2}\left(e_{x+} - 2 + e_{x-} - e_{t-}(e_{x+} - 2 + e_{x-})\right)$$

Taking the mixed derivative for the

$$\delta_{t-}\delta_{xx} u_l^n = \begin{cases} \frac{1}{k}\left(\delta_{xx} u_l^n - \delta_{xx} u_l^{n-1}\right) & \text{expanding } \delta_{t-} \\ \frac{1}{h^2}\left(\delta_{t-} u_{l+1}^n - 2\delta_{t-} u_l^n + \delta_{t-} u_{l-1}^n\right) & \text{expanding } \delta_{xx} \end{cases}$$

and after expansion of the second operator both result in

$$\delta_{t-}\delta_{xx} u_l^n = \frac{1}{hk^2}\left(u_{l+1}^n - 2u_l^n + u_{l-1}^n - u_{l+1}^{n-1} + 2u_l^{n-1} - u_{l-1}^{n-1}\right) \tag{3.6}$$

Chapter 4

2D Systems

The 2D wave equation be used to model an ideal membrane such as done in Paper

In this work, it is mainly used to model a simplified body in papers...

State variable $u = u(x, y, t)$ where $t \geq 0$ and $(x, y) \in \mathcal{D}$ where $\mathcal{D}$ is 2-dimensional. The state variable can then be discretised according to $u(x, y, t) \approx u_{l,m}^n$ with space $x = lh$ and $y = mh$ and time $t = nk$ and $k = 1/f_\text{s}$. For simplicity the grid spacing in both the x and y directions are set to be the same but could be different.

The same shift operators as defined in Chapter 2 can be applied to grid function $u_{l,m}^n$

$$e_{y+}u_{l,m}^n = u_{l,m+1}^n, \quad \text{and} \quad e_{y-}u_{l,m}^n = u_{l,m-1}^n. \tag{4.1}$$

Additional operators:

$$\Delta = \partial_x^2 + \partial_y^2 \tag{4.2}$$

4.1 2D Wave Equation

4.2 Thin plate

4.3 Stiff membrane

Chapter 5

Brass

This will be the first appearance of a first-order system.

Part III

Exciters

Exciters

Now that a plethora of resonators have been introduced in part II, different mechanisms to excite them will be introduced here. First, different examples of

- Simple pluck ((half) raised-cos)
- Hammer
- Bow Models
- Lip reed

Chapter 6

Unmodelled Excitations

Different title here?

6.1 Initial conditions

Hammer

Full raised cosine

Pluck

- Cut-off raised cosine
- Triangle (for string)

6.2 Signals

6.2.1 Pulse train

For brass

6.2.2 Noise

Noise input

Chapter 7

Modelled Excitations

7.1 Hammer

Hammer modelling

7.2 The Bow

The bow...

Helmholtz motion..

Characteristic triangular motion

7.2.1 Static Friction Models

In static bow-string-interaction models, the friction force is defined as a function of the relative velocity between the bow and the string only. The first mathematical description of friction was proposed by Coulomb in 1773 [**?**] to which static friction, or *stiction*, was added by Morin in 1833 [**?**] and viscous friction, or velocity-dependent friction, by Reynolds in 1886 [**?**]. In 1902, Stribeck found a smooth transition between the static and the coulomb part of the friction curve now referred to as the Stribeck effect [**?**]. The latter is still the standard for static friction models today.

7.2.2 Dynamic Friction Models

As opposed to less complex bow models, such as the hyperbolic [source] and exponential [source] models, the elasto-plastic bow model assumes that the friction between the bow and the string is caused by a large quantity of bristles, each of which contributes to the total amount of friction.

7.3 Lip-reed

Lip-reed model

7.3.1 Coupling to Tube

Part IV

Interactions

The models described in part II already sound quite convincing on their own. However, these are just individual components that can be combined to approximate a fully functional (virtual) instrument. The following chapters will describe different ways of interaction between individual systems. Chapter 8 describes ways to connect different systems and Chapter 9 describes collision interactions between models.

ε

Newtons third law (action reaction)

Subscripts are needed

Somewhere in this Chapter have a section about fretting and how to generate different pitches using only one string

Interpolation and spreading operators

Using $l_c = \lfloor x_c/h \rfloor$ and $\alpha_c = x_c/h - l_c$ is the fractional part the location of interest.

$$I_0(x_c) = \begin{cases} 1, & \text{if } l = l_c, \\ 0, & \text{otherwise} \end{cases} \tag{7.1}$$

$$I_1(x_c) = \begin{cases} (1-\alpha_c), & \text{if } l = l_c \\ \alpha_c, & \text{if } l = l_c + 1 \\ 0 & \text{otherwise} \end{cases} \tag{7.2}$$

$$I_3(x_c) = \begin{cases} ... \end{cases} \tag{7.3}$$

The following identity is very useful when solving interactions between components:

$$\langle f, J_p(x_c) \rangle_{\mathcal{D}} = I_p(x_c) f \tag{7.4}$$

Chapter 8

Connections

Something about connections

Pointlike $\delta(x - x_\mathrm{c})$ or distributed E_c

8.1 Rigid connection

The simplest connection is Forces should be equal and opposite.

If component a is located 'above' component b, and their relative displacement is defined as $\eta = a - b$, then a positive η is going to have a negative effect on a and a positive effect on b and vice-versa. This is important for the signs when adding the force terms to the schemes.

8.2 Spring-like connections

8.2.1 Connection with rigid barrier (scaled)

Consider the (scaled) 1D wave equation with an additional force term F^n

$$\delta_{tt} u_l^n = \gamma^2 \delta_{xx} u_l^n + J(x_\mathrm{c}) F^n \tag{8.1}$$

where

$$F^n = -\omega_0^2 \mu_{t\cdot} \eta^n - \omega_1^4 (\eta^n)^2 \mu_{t\cdot} \eta^n - 2\sigma_\times \delta_{t\cdot} \eta^n \tag{8.2}$$

and

$$\eta^n = I(x_\mathrm{c}) u_l^n. \tag{8.3}$$

To obtain F^n, an inner product of scheme (8.1) needs to be taken with $J(x_\mathrm{c})$ over domain $\mathcal{D}$ which, using identity (7.4) yields

$$\delta_{tt} I(x_\mathrm{c}) u_l^n = \gamma^2 I(x_\mathrm{c}) \delta_{xx} u_l^n + \underbrace{I(x_\mathrm{c}) J(x_\mathrm{c})}_{\|J(x_\mathrm{c})\|_\mathcal{D}^2} F^n. \tag{8.4}$$

As u is connected to a rigid barrier according to (8.3), a shortcut can be taken and Eqs. (8.2) and (8.3) can be directly substituted into Eq. (8.4) to get

$$\delta_{tt}\eta^n = \gamma^2 I(x_\mathrm{c})\delta_{xx}u_l^n + \|J(x_\mathrm{c})\|_\mathcal{D}^2 \left(-\omega_0^2\mu_{t\cdot}\eta^n - \omega_1^4(\eta^n)^2\mu_{t\cdot}\eta^n - 2\sigma_\times\delta_{t\cdot}\eta^n\right) . \tag{8.5}$$

and solved for η^{n+1}:

$$\begin{aligned} &\Big(1 + \|J(x_\mathrm{c})\|_\mathcal{D}^2 k^2[\omega_0^2/2 + \omega_1^4(\eta^n)^2/2 + \sigma_\times/k]\Big)\eta^{n+1} \\ &\quad = 2\eta^n - \Big(1 + \|J(x_\mathrm{c})\|_\mathcal{D}^2 k^2[\omega_0^2/2 + \omega_1^4(\eta^n)^2/2 - \sigma_\times/k]\Big)\eta^{n-1} \\ &\qquad + \gamma^2 k^2 I(x_\mathrm{c})\delta_{xx}u_l^n \end{aligned} \tag{8.6}$$

This can then be used to calculate F^n in (8.2) and can in turn be used to calculate u_l^{n+1} in (8.1).

8.2.2 String-plate connection

In this example, let's consider a string connected to a plate using a nonlinear damped spring. This could be interpreted as a simplified form of how guitar string would be connected to the body.

Continuous

The systems in isolation are as in (3.4) and (**??**), but with an added force term:

$$\partial_t^2 u = c^2\partial_x^2 u - \kappa_\mathrm{s}^2\partial_x^4 u - 2\sigma_{0,\mathrm{s}}\partial_t u + 2\sigma_{1,\mathrm{s}}\partial_t\partial_x^2 u - \delta(x - x_\mathrm{c})\frac{f}{\rho_\mathrm{s}A} \tag{8.7a}$$

$$\partial_t^2 w = -\kappa_\mathrm{p}^2\Delta\Delta w - 2\sigma_{0,\mathrm{p}}\partial_t w + 2\sigma_{1,\mathrm{p}}\partial_t\partial_x^2 w + \delta(x - x_\mathrm{c}, y - y_\mathrm{c})\frac{f}{\rho_\mathrm{p}H} \tag{8.7b}$$

where

$$f = f(t) = K_1\eta + K_3\eta^3 + R\dot{\eta} \tag{8.8}$$

and

$$\eta = \eta(t) = u(x_\mathrm{c}, t) - w(x_\mathrm{c}, y_\mathrm{c}, t) \tag{8.9}$$

Discrete

System (8.7) can then be discretised as

$$\delta_{tt}u_l^n = c^2\delta_{xx}u_l^n - \kappa_\mathrm{s}^2\delta_{xxxx}u_l^n - 2\sigma_{0,\mathrm{s}}\delta_{t\cdot}u_l^n + 2\sigma_{1,\mathrm{s}}\delta_{t-}\delta_{xx}u_l^n - J_\mathrm{s}(x_\mathrm{c})\frac{f^n}{\rho_\mathrm{s}A} , \tag{8.10}$$

$$\delta_{tt}w_l^n = -\kappa_\mathrm{p}^2\delta_\Delta\delta_\Delta w_l^n - 2\sigma_{0,\mathrm{p}}\delta_{t\cdot}w_l^n + 2\sigma_{1,\mathrm{p}}\delta_{t-}\delta_{xx}w_l^n + J_\mathrm{p}(x_\mathrm{c}, y_\mathrm{c})\frac{f^n}{\rho_\mathrm{p}H} , \tag{8.11}$$

where

$$f^n = K_1 \mu_{tt} \eta^n + K_3 (\eta^n)^2 \mu_{t\cdot} \eta^n + R \delta_{t\cdot} \eta^n, \tag{8.12}$$

and

$$\eta^n = I(x_\mathrm{c}) u_l^n - I(x_\mathrm{c}, y_\mathrm{c}) w_l^n. \tag{8.13}$$

Expansion

System (8.10) can be expanded at the connection location x_c by taking an inner product of the schemes with their respective spreading operators.

8.2.3 Solving for f

8.2.4 Non-dimensional

The scaled system can be written as:

$$\partial_t^2 u = \gamma^2 \partial_x^2 u - \kappa_\mathrm{s}^2 \partial_x^4 u - 2\sigma_{0,\mathrm{s}} \partial_t u + 2\sigma_{1,\mathrm{s}} \partial_t \partial_x^2 u - \delta(x - x_\mathrm{c}) F \tag{8.14}$$

$$\partial_t w = -\kappa_\mathrm{p}^2 \Delta\Delta w - 2\sigma_{0,\mathrm{p}} \partial_t w + 2\sigma_{1,\mathrm{p}} \partial_t \partial_x^2 w + \delta(x - x_\mathrm{c}, y - y_\mathrm{c}) F \tag{8.15}$$

where

$$F = F(t) = \omega_1^2 \eta + \omega_3^4 \eta^3 + \sigma_\mathrm{c} \dot{\eta} \tag{8.16}$$

and

$$\eta = \eta(t) = u(x_\mathrm{c}, t) - w(x_\mathrm{c}, y_\mathrm{c}, t) \tag{8.17}$$

$$I(x_\mathrm{c}) \delta_{tt} u_l^n = c^2 \left(I(x_\mathrm{c}) \delta_{xx} u_l^n \right) + I(x_\mathrm{c}) J(x_\mathrm{c}) F \tag{8.18}$$

Chapter 9

Collisions

Something about collisions

9.1 Classic models

Note that when using Eq. (7.37) in [2]

9.2 Michele's tricks

Part V

Dynamic Grids

Chapter 10

Dynamic Grids

Often in math, you should view the definition not as a starting point, but as a target. Contrary to the structure of textbooks, mathematicians do not start by making definitions and then listing a lot of theorems, and proving them, and showing some examples. The process of discovering math typically goes the other way around. They start by chewing on specific problems, and then generalising those problems, then coming up with constructs that might be helpful in those general cases,and only then you write down a new definition (or extend an old one). - Grant Sanderson (AKA 3Blue1Brown) https://youtu.be/O85OWBJ2ayo?t=359

10.1 Background and Motivation

Simulating musical instruments using physical modelling – as mentioned in Part I– allows for manipulations of the instrument that are impossible in the physical world. Examples of this are changes in material density or stiffness, cross-sectional area (1D), thickness (2D) and size. Apart from being potentially sonically interesting, there are examples in the physical world where certain aspects of the instrument are manipulated in real-time.

check if is still true

Tension in a string is changed when tuning it

Some artists even use this in their performances [7, 8]

The hammered dulcimer is another example where the strings are tensioned over a bridge where one can play the string at one side of the bridge, while pushing down on the same string on the other side [6].
1D:

- Trombone
- Slide whistle
- Guitar strings

  - Fretting finger pitch bend
  - Above the nut [8]
  - Tuning pegs directly [7]

- Hammered dulcimer [6]
- Erhu?

2D:

- Timpani
- Bodhrán: https://youtu.be/b9HyB5yNS1A?t=146
- Talking drum (hourglass drum): https://youtu.be/B4oQJZ2TEVI?t=9
- Flex-a-tone (could also be 1D tbh..): https://www.youtube.com/watch?v=HEW1aG8XJQ

A more relevant example is that of the trombone, where the size of the instrument is changed in order to play different pitches. Modelling this using FDTD methods would require

In his thesis, Harrison points out that in order to model the trombone, grid points need to be introduced

Something about time-dependent variable coefficient Stokes flow: https://arxiv.org/abs/10

Time-varying propagation speed in waveguides: https://quod.lib.umich.edu/cgi/p/pod/idx/fractional-delay-application-time-varying-propagation-speed.pdf?c=icmc;idno=bbp2372.

Special boundary conditions (look at!): Modeling of Complex Geometries and Boundary Conditions in Finite Difference/Finite Volume Time Domain Room Acoustics Simulation (`https://www.researchgate.net/publication/260701231_Modeling_of_Complex_Geometries_and_Boundary_Conditions_in_Finite_DifferenceFinite_Volume_Time_Domain_Room_Acoustics_Simulation`)

10.2 Method

Iterations have been:

- Interpolated boundary conditions
- Linear interpolation

These sections are taken from the JASA appendix

In this appendix, some iterations done over the course of this project will be shown in more detail. In the following, the 1D wave equation with a wave speed of $c = 1470$ m/s, a length of $L = 1$ m, Dirichlet boundary conditions and a sample rate of $f_s = 44100$ Hz is considered, and – through Eq. (**??**) – satisfies the CFL condition with equality. These values result in $N = 30$, or a grid of 31 points including the boundaries. Then, the wave speed is decreased to $c \approx 1422.6$ m/s, i.e., the wave speed that results in $N = 31$ and satisfies the stability condition with equality again.

10.2.1 Full-Grid Interpolation

One way to go from one grid to another is performing a full-grid interpolation [?, Chap. 5]. If the number of points changes according to Eq. (??), i.e., if $N^n \neq N^{n-1}$ the full state of the system $(u_l^n, u_l^{n-1}\ \forall l)$ can be interpolated to the new state. See Figure 10.1.

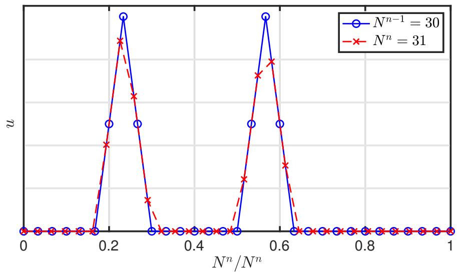


Fig. 10.1: Upsampling u (with an arbitrary state) using (linear) full-grid interpolation with $N^{n-1} = 30$ and $N^n = 31$. The horizontal axis is normalised with respect to N^n.

An issue that arises using this method is that the Courant number λ will slightly deviate from the CFL condition as c changes. Using Eq. (??) with L/ck approaching 31 (from below), the minimum value of $\lambda \approx 30/31 \approx 0.9677$. This, employing Eq. (??), has a maximum frequency output of $f_{\text{max}} \approx 18,475$ Hz. The Courant number will deviate more for higher values of c and thus lower values for N – for instance, if N approaches 11 (from below), $\lambda \approx 10/11 \approx 0.9091$ and $f_{\text{max}} \approx 16,018$ Hz.

Another problem with full-grid interpolation, is that it has a low-passing effect on the system state, and thus on the output sound. Furthermore, this state-interpolation causes artefacts or 'clicks' in the output sound as the method causes sudden variations in the states.

All the aforementioned issues could be solved by using a (much) higher sample rate and thus more grid points, but this would render this method impossible to work in real time.

10.2.2 Adding and removing Points at the Boundary

To solve the issues exhibited by a full-grid interpolation, points can be added and removed at a single location and leave most points unaffected by the parameter changes. A good candidate for a location to do this is at a fixed

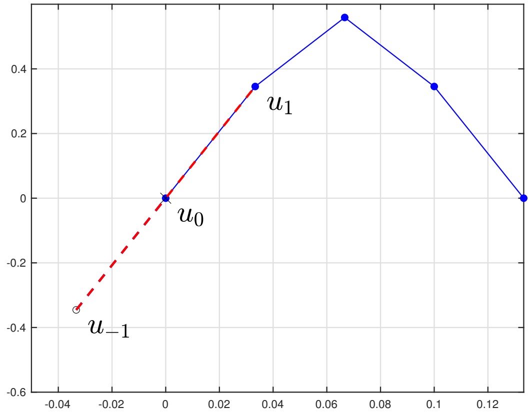


Fig. 10.2: The simply supported boundary condition: both the state and the curvature at the boundary – at $l = 0$ – should be 0.

(Dirichlet) boundary. The state u at this location is always 0 so points can be added smoothly.

As c decreases, h can be calculated according to Eq. (**??**) and decreases as well.

This has a physical analogy with tuning a guitar string. Material enters and exits the neck (playable part of the string) at the nut, which in discrete time means grid points appearing and disappearing at one boundary.

To yield smooth changes between grid configurations, an interpolated boundary has been developed, the possibility of which has been briefly mentioned in [**?**, p. 145]. The Dirichlet condition in Eq. (2.14a) can be extended to be the simply supported boundary condition:

$$u(x,t) = \frac{\partial^2}{\partial x^2} u(x,t) = 0 \quad \text{where} \quad x = 0, L, \tag{10.1}$$

or, when discretised,

$$u_l^n = \delta_{xx} u_l^n = 0, \quad \text{where} \quad l = 0, N. \tag{10.2}$$

This means that on top of that the state of the boundary should be 0, the curvature around it should also be 0. One can again solve for the virtual grid points at the boundary locations, yielding

$$u_{-1}^n = -u_1^n \quad \text{and} \quad u_{N+1}^n = -u_{N-1}^n. \tag{10.3}$$

This is visualised in Figure 10.2.

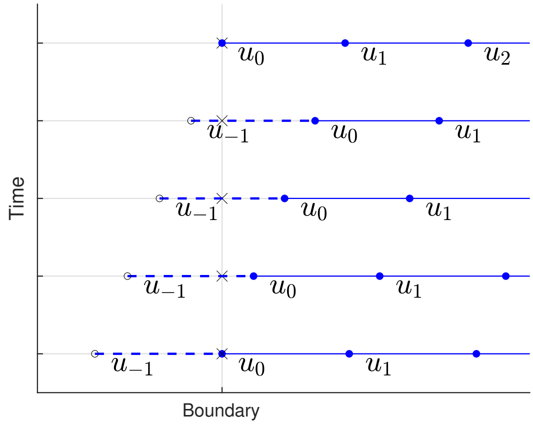


Fig. 10.3: The grid changing over time

If the flooring operation in Eq. (**??**) is removed this introduces a fractional number of grid points.

The by-product of using a fractional N this is that the CFL condition in (2.16) can now always be satisfied with equality no matter what the wave speed is.

An issue with this method is that removing points is much harder than adding.

their interactions change through a change in the grid spacing and wave speed. This interaction, though, is defined by λ which

10.2.3 Cubic interpolation

10.2.4 Sinc interpolation

10.2.5 Displacement correction

The displacement correction can be interpreted as a spring force pulling u_M^n and w_0^n to the average displacement.

$$
\begin{aligned}
u_M^{n+1} &= 2u_M^n + \lambda^2(u_{M-1}^n - 2u_M^n + u_{M+1}^n) - K\left(u_M^n - \frac{u_M^n + w_0^n}{2}\right) \\
w_0^{n+1} &= 2u_M^n + \lambda^2(w_{-1}^n - 2w_0^n + w_1^n) - K\left(w_0^n - \frac{u_M^n + w_0^n}{2}\right)
\end{aligned} \tag{10.4}
$$

$$
\begin{aligned}
u_M^{n+1} &= 2u_M^n + \lambda^2(u_{M-1}^n - 2u_M^n + u_{M+1}^n) + \frac{K}{2}(w_0^n - u_M^n) \\
w_0^{n+1} &= 2u_M^n + \lambda^2(w_{-1}^n - 2w_0^n + w_1^n) - \frac{K}{2}(w_0^n - u_M^n)
\end{aligned} \tag{10.5}
$$

with $K = K(\alpha)$

$$
K = (1 - \alpha)^\epsilon. \tag{10.6}
$$

10.3 Analysis and Experiments

10.3.1 Interpolation technique

10.3.2 Interpolation range

10.3.3 Location

... where to add and remove points

Using the whole range, we can still add/remove points at the sides.

10.4 Discussion and Conclusion

Part VI

Real-Time Implementation and Control

Real-Time Implementation and Control

It’s all fun and dandy with all these physical models, but what use are they if you can’t control them in real time?!

Chapter 11

Real-Time Implementation

JUCE Give overall structure of code

Implementation of the physical models using FDTD methods

As mentioned in Chapter 1, FDTD methods are used for high-quality and accurate simulations, rather than for real-time applications. This is due to their lack of simplifications.

Usually, `MATLAB` is used for simulating

Here, an interactive application is considered real-time when

Control of the application generates or manipulates audio with no noticeable latency.

Also the application needs to be controlled continuously

Chapter 12

Control

12.1 Sensel Morph

150 Hz

12.2 Phantom OMNI

Part VII

Complete Instruments

Complete Instruments

This part will give several examples of full instrument models that have been developed during the PhD. Chapter **??** shows a three instrument-inspired case-studies using a large-scale modular environment, Chapter 14 describes the implementation of the tromba marina and Chapter 15 that of the trombone.

Chapter 13

Large Scale Modular Physical models

In the paper "Real-Time Control of Large-Scale Modular Physical Models using the Sensel Morph" [?] we presented a modular physical modelling environment using three instruments as case studies.

13.1 Bowed Sitar

- Stiff String
- Thin Plate
- Pluck
- Bow
- Non-linear spring connections

13.2 Dulcimer

- Stiff String
- Thin Plate
- Hammer (simple)
- Non-linear spring connections

13.3 Hurdy Gurdy

- Stiff String
- Thin Plate
- Pluck
- Bow
- Non-linear spring connections

Chapter 14

Tromba Marina

14.1 Introduction

14.2 Physical Model

14.2.1 Continuous

14.2.2 Discrete

14.3 Real-Time Implementation

14.3.1 Control using Sensel Morph

14.3.2 VR Application

Chapter 15

Trombone

Published in [H]

15.1 Introduction

Interesting read: https://newt.phys.unsw.edu.au/jw/brassacoustics.html

15.2 Physical Model

Most has been described in Chapter 5

15.2.1 Continuous

Just to save the conversation with Stefan about Webster's equation:

Using operators ∂_t and ∂_x denoting partial derivatives with respect to time t and spatial coordinate x, respectively, a system of first-order PDEs describing the wave propagation in an acoustic tube can then be written as

$$\frac{S}{\rho_0 c^2}\partial_t p = -\partial_x(Sv) \tag{15.1a}$$

$$\rho_0 \partial_t v = -\partial_x p \tag{15.1b}$$

with acoustic pressure $p = p(x,t)$ (in N/m^2), particle velocity $v = v(x,t)$ (in m/s) and (circular) cross-sectional area $S(x)$ (in m^2). Furthermore, ρ_0 is the density of air (in kg/m^3) and c is the speed of sound in air (in m/s). System (15.1) can be condensed into a second-order equation in p alone, often referred to as Webster's equation [**?**]. Interesting! In NSS it is the acoustic potential right? Can you go from that to a second-order PDE in p? There is a time-derivative hidden there somewhere right? (Just wondering :))Yes, the form in p alone is the one you usually see. You get it by differentiating the first equation, giving you a $\dot{v}$ on the RHS, and then you can substitute the second equation in...I used the velocity potential one because it has direct energy balance properties. Right. So Webster's eq. in p and Ψ are identical (will exhibit identical behaviour), except for the unit of the state variable..?yes that's right...using the velocity potential allows you to do all the energy analysis easily, in terms of physical impedances. But the scheme you get to in the end is the same, just one derivative down. Alright cool! Thanks for the explanation :) For simplicity, effects of viscothermal losses have been neglected in (15.1). For a full time domain model of such effects in an acoustic tube, see, e.g. [**?**].

15.2.2 Discrete

15.3 Real-Time Implementation

Unity??

15.4 Discussion

more for your info, don't think I want to include this: To combat the drift, experiments have been done involving different ways of connecting the left and right tube. One involved alternating between applying the connection to the pressures and the velocity. Here, rather than adding points to the left and right system in alternating fashion, points were added to pressures p and q and velocities v and w in an alternating fashion. Another experiment involved a "staggered" version of the connection where (fx.) for one system (either left or right), a virtual grid point of the velocity was created from known values according to (**??**), rather than both from pressures. This, however, showed unstable behaviour. No conclusory statements can be made about these experiments at this point. $\leftarrow$ which is exactly why I don't want to include this section

As the geometry varies it matters a lot where points are added and removed as this might influence the way that the method is implemented. speculative section coming up The middle of the slide crook was chosen, both because it would be reasonable for the air on the tube to "go away from" or "go towards" that point as the slide is extended or contracted, and because the geometry does not vary there. Experiments with adding / removing grid points where the geometry varies have been left for future work. even more speculative.. $\rightarrow$ It could be argued that it makes more sense to add points at the ends of the inner slides as "tube material" is also added there. This would mean that the system should be split in three parts: "inner slide", "outer slide" and "rest", and would complicate things even more.

check whether all references are used

References

[1] J. Bensa, S. Bilbao, R. Kronland-Martinet, and J. O. Smith, "The simulation of piano string vibration: From physical models to finite difference schemes and digital waveguides," *Journal of the Acoustical Society of America (JASA)*, vol. 114, no. 2, pp. 1095—-1107, 2003.

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

[3] S. Bilbao, C. Desvages, M. Ducceschi, B. Hamilton, R. Harrison-Harsley, A. Torin, and C. Webb, "Physical modeling, algorithms, and sound synthesis: The ness project," *Computer Music Journal*, vol. 43, no. 2-3, pp. 15–30, 2019.

[4] S. Bilbao, J. Perry, P. Graham, A. Gray, K. Kavoussanakis, G. Delap, T. Mudd, G. Sassoon, T. Wishart, and S. Young, "Large-scale physical modeling synthesis, parallel computing, and musical experimentation: The ness project in practice," *Computer Music Journal*, vol. 43, no. 2-3, pp. 31–47, 2019.

[5] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*. Springer, 1998.

[6] T. L. Glenn, "Amazing hammered dulcimer musician - joshua messick," 2014. [Online]. Available: https://youtu.be/veuGTnzgNRU?t=215

[7] J. Gomm, "Passionflower," 2011. [Online]. Available: https://www.youtube.com/watch?v=nY7GnAq6Znw

[8] J. Mayer, "Gravity (live in l.a.)," 2008. [Online]. Available: https://youtu.be/dBFW8OvciIU?t=284

[9] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, pp. 114—117, 1965.

Part VIII

Appendix

Appendix A

List of Symbols

Symbol	Description	Unit
A	Cross-sectional area	m^2
c	Wave speed	m/s
E	Young's Modulus	Pa ($\mathrm{kg{\cdot}m^{-1}{\cdot}s^{-2}}$)
f_s	Sample rate	s^{-1}
F_α	Connection force string	N ($\mathrm{kg{\cdot}m{\cdot}s^{-2}}$)
F_β	Connection force plate	N
h	Grid spacing	m
H	Plate thickness	m
I	Area moment of inertia **[?]**	m^4
l	Spatial index to grid function	-
L	String length	m
k	Time step ($1/f_\mathrm{s}$)	s
K_1	Linear spring coefficient	N/m
K_3	Non-linear spring coefficient	$\mathrm{N{\cdot}m^{-3}}$
n	Sample index to grid function	-
N	Number of points string	-
T	String tension	N
u	State variable ($u(x,t)$ or $u(x,y,t)$)	m (displacement)
γ	Scaled wave speed	s^{-1}
κ	Stiffness coefficient	m^2/s (string) $\mathrm{m^4{\cdot}s^{-2}}$ (plate)
λ	Courant number for the wave equation	m/s
μ	Similar to courant number but for plate	m/s

Symbol	Description	Unit
ν	Poisson's ratio	-
η	Relative displacement spring	m
ρ	Material density	kg·m^{-3}

Appendix B

List of Abbreviations

Abbreviation	Definition
FDS	Finite-difference scheme
FDTD	Finite-difference time-domain

Part IX

Papers

Paper Errata

Here, some errors in the published papers will be listed:
Real-Time Tromba [D]:

- The minus sign in Eq. (28) (and thus Eqs. (31) and (35)) should be a plus sign.
- $\sigma_{1,\mathrm{s}}$ in Eq. (21) should obviously be $\sigma_{1,\mathrm{p}}$
- the unit of the spatial Dirac delta function δ should be m^{-1}

DigiDrum [F]:

- σ_0 and σ_1 should be multiplied by ρH in order for the stability condition to hold.
- stability condition is wrong. Should be:

$$h \geq \sqrt{c^2k^2 + 4\sigma_1 k + \sqrt{(c^2k^2 + 4\sigma_1 k)^2 + 16\kappa^2 k^2}} \tag{1}$$

- Unit for membrane tension is N/m.

Dynamic grids [G]

- Reference in intro for 'recently gained popularity' should go to [3]
  Note: not really an error, but should be changed before resubmission

Paper A

Real-Time Control of Large-Scale Modular Physical Models using the Sensel Morph

Paper B

Physical Models and Real-Time Control with the Sensel Morph

Paper C

Real-Time Implementation of an Elasto-Plastic Friction Model applied to Stiff Strings using Finite Difference Schemes

Paper D

Real-time Implementation of a Physical Model of the Tromba Marina

Paper E

Resurrecting the Tromba Marina: A Bowed Virtual Reality Instrument using Haptic Feedback and Accurate Physical Modelling

Paper F

DigiDrum: A Haptic-based Virtual Reality Musical Instrument and a Case Study

Paper G

Dynamic Grids for Finite-Difference Schemes in Musical Instrument Simulations

Paper H

A Physical Model of the Trombone using Dynamic Grids for Finite-Difference Schemes