Real-time physical model of the Rhodes Electric Piano

Master Thesis

Tobias Sønderbo

Aalborg University
Architecture, Design and Media Technology

Architecture, Design and Media Technology
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Real-time physical model of the Rhodes Electric Piano

Theme:
Physical Models for Sound Synthesis

Project Period:
Fall Semester 2024

Participant(s):
Tobias Sønderbo

Supervisor(s):
Stefania Serafin

Page Numbers: 32

Date of Completion:
May 23, 2024

Abstract:

The use of computers for music production has become widely popular and many instruments are now available in digital format as audio plugins for use in Digital Audio Workstations. This popularity has paved the way for new technologies to allow for higher levels of expression when interacting with digital instruments. This work focuses on developing a physically modelled Rhodes piano using finite-difference time-domain methods with new layers of expression. The model is implemented in a real-time application with support for MPE control interfaces. The model has issues with tuning and the lack of a physically based pickup model results in a tone production that deviates from the Rhodes. Preliminary feedback on the implementation of MPE controls were positive. A set of improvements and additions are suggested to bring the model closer to the desired output.

Contents

Chapter 1

Introduction

1.1 Rhodes Electric Piano

The Rhodes piano is an electro-acoustical keyboard with a unique tone that has since seen great success in genres like Funk, Jazz, Rock and Pop since the 1960s. Despite its age the Rhodes is still commonly used today and is often found on digital pianos (usually labeled 'E.Piano' or similar for copyright reasons). With the increasing use of computers for music production, many instruments are now available in digital format as audio plugins for use in Digital Audio Workstations (DAW). The Rhodes piano is no exception with an audio plugin sold under the official brand by its current manufacturer [1]. The plugin offers a sampled version of the Rhodes with several presets for the past and current versions of instrument.

1.2 Synthesized instruments

An alternative solution to recording the many samples required for a sampled plugins is synthesizing the sounds directly. A well designed synthesis algorithm can generate the desired output at runtime eliminating the need for the time consuming process of recording the many samples required to represent an instruments full range and dynamics. An additional benefit of a synthesised approach is the significantly reduced disk space required for the plugin. Spectral modelling methods where noise or sinusoids are manipulated can be used to create sounds that are perceptually similar to the desired instrument. One such method is FM (Frequency modulation) synthesis [2], which gained popularity following the release of the Yamaha DX7 in 1983. While the method is computationally simple it is limited in its ability to capture details of nonlinear effects [3].

A different approach to synthesising the sound of an instrument is by modelling the source of the sound also known as physical modelling. The resonators of the instrument is expressed mathematically using differential equations which describe its physical behavior. These equations are discretized and run on a computer to simulate the instrument

from which sound can be generated. One approach of discretizing differential equations is finite-difference time-domain (FDTD) methods [4]. FDTD methods have the advantage of being highly configurable and able to capture details that FM synthesis otherwise lack. Recent additions to the method in the form of the dynamic grid [5, 6] allow for time-varying parameters which broadens the range of expressive possibilities. The main disadvantage of FDTD methods is the computational cost.

1.3 MIDI Polyphonic Expression

Introduced in 2018, the MIDI Polyphonic Expression (MPE) specification is an extension to the widely used MIDI protocol [7]. The purpose of MPE is to add individual channels for per-note control. This allows developers to make midi interfaces with new control surfaces to allow for higher levels of musical expression. The manufacturer ROLI has adopted this technology offering several controllers and software packages that support MPE [8] and in their time as owners of the C++ audio development library JUCE [9] they added interfaces to MPE controls for developers. MPE is also supported by most major DAWs such as Reaper and Ableton.

1.4 Project goal

The combination of the new control surfaces in MPE and the recent additions to FDTD methods presents an opportunity to develop a physically modelled Rhodes piano with new layers of expression. By mapping physical parameters to control surfaces in MPE, techniques that are not physically possible with the Rhodes such as pitch bending can be made available. The aim of this project is to develop a physical model of the Rhodes piano using FDTD methods while adding new layers of expression through the inclusion of the dynamic grid and MPE mappings.

Chapter 2

Finite-difference time-domain methods

Finite-difference time-domain methods describe a set of tools for discretizing differential equations. The method is used on both ordinary difference equations (ODE) and partial difference equations (PDE) and approximates the continuous equation by subdividing it into discrete points in time and space. A discretized ODE or PDE is then referred to as a finite difference (FD) scheme.

This chapter will give a limited introduction to FDTD methods, which will cover the functions and operators relevant for this work. The theory for this chapter is from [10, 4]. Notation used is subscript notation for discrete differential equations and Euler's notation for PDEs.

2.1 Domains and grid functions

To discretize a PDE, one must first define a discrete grid. The grid consists of a series of grid points which serve as a discrete sampling of the spatial dimensions of the continuous system. The grid is then updated for each time-step in the defined temporal range. In the case of an ODE, the grid is a single point.

The temporal range is sampled according to a manually defined sample-rate, which in the case of an audio-application is the audio sample-rate (e.g. 44100 Hz). Time step k (in seconds) is defined as $k = 1/f_s$, with f_s being the sample rate (in Hz). The discrete time range is then defined as $n \in \mathbb{N}_0$ steps of k length.

The spatial domains are sampled over a defined finite length of each dimension. In this work, only one-dimensional systems will be considered, meaning we have a single spatial domain. The spatial domain is subdivided into N intervals, each equally spaced with grid spacing h (in meters). The discrete spatial domain is then defined as $l \in \{0, ..., N\}$, for a total of $N + 1$ grid points. As mentioned earlier, FDTD methods are an approximation of

the continuous equation, and its accuracy is largely tied to the grid spacing of the spatial dimension. Ideal grid spacing is defined by the stability condition of FD scheme used.

Once the grid has been defined, the grid points can then be indexed as u_l^n where u is the state variable and n, l are the temporal and spatial indices.

2.2 Finite-difference operators

A PDE can be discretised using FD operators. A set of operators are defined for both first-order time derivatives and first-order spatial derivatives. In both cases, a first-order derivative can be expanded using either a forwards difference, a backwards difference or a centered difference. These are defined as:

$$\partial u_t \approxeq \begin{cases} \delta_{t+} u_l^n = \frac{1}{k} = (u_l^{n+1} - u_l^n), \\ \delta_{t-} u_l^n = \frac{1}{k} = (u_l^n - u_l^{n-1}), \\ \delta_{t\cdot} u_l^n = \frac{1}{2k} = (u_l^{n+1} - u_l^{n-1}). \end{cases} \tag{2.1}$$

$$\partial u_x \approxeq \begin{cases} \delta_{x+} u_l^n = \frac{1}{h} = (u_{l+1}^n - u_l^n), \\ \delta_{x-} u_l^n = \frac{1}{h} = (u_l^n - u_{l-1}^n), \\ \delta_{x\cdot} u_l^n = \frac{1}{2h} = (u_{l+1}^n - u_{l-1}^n). \end{cases} \tag{2.2}$$

Here we see the indices having increments or decrements. This is to denote relative placement. For example, a centered difference is calculated based on the previous and next grid point, either temporally or spatially depending on the derivative.

To get a discrete version of a second order derivative, a first order forwards difference and backwards difference are multiplied together.

$$\partial_t^2 \approxeq \delta_{t+}\delta_{t-} u_l^n = \delta_{tt} u_l^n = \frac{1}{k^2}(u_l^{n+1} - 2u_l^n + u_l^{n+1}) \tag{2.3}$$

$$\partial_x^2 \approxeq \delta_{x+}\delta_{x-} u_l^n = \delta_{xx} u_l^n = \frac{1}{h^2}(u_l^{n+1} - 2u_l^n + u_l^{n+1}) \tag{2.4}$$

Similarly, fourth order spatial derivatives can be discretised through multiplication of two second order difference operators.

$$\partial_x^4 \approxeq \delta_{xx}\delta_{xx} u_l^n = \delta_{xxxx} u_l^n = \frac{1}{h^4}(u_{l+2}^n - 4u_{l+1}^n + 6u_l^n - 4u_{l-1}^n + u_{l-2}^n) \tag{2.5}$$

Finally, temporal and spatial operators can be multiplied together to solve for mixed derivatives. The following mixed derivative is used for a difference equation in later chapters.

$$\partial_t \partial_x^2 \approxeq \delta_{t-}\delta_{xx} u_l^n = \frac{1}{kh^2}(u_{l+1}^n - 2u_l^n + u_{l-1}^n - u_{l+1}^{n-1} + 2u_l^{n-1} + u_{l-1}^{n-1}) \tag{2.6}$$

2.3 Boundary conditions

For any given FD Scheme, the conditions for the boundaries must be defined. As the spatial dimension is finite, the necessary grid points needed for second order and fourth order derivatives, are not present at the end points of the grid. The end points are defined as location 0 and location N. This work makes use of two different boundary conditions which are defined as

$$u_l^n = \delta_{x\pm} u_l^n = 0 \quad (clamped) \tag{2.7}$$

$$\delta_{xx} u_l^n = \delta_x . \delta_{xx} u_l^n = 0 \quad (free) \tag{2.8}$$

The clamped boundary can be solved by simply setting the outer two grid points to 0 at all times for a given end point. So to clamp the boundary at the end point N both grid point $N-1$ and N would be set to 0 at all times. The free boundary can be expanded to solve for a set of virtual grid points, which hold the values necessary to calculate the end grid points.

$$u_{N+1}^n = 2u_N^n - u_{N-1}^n \tag{2.9}$$

$$u_{N+2}^n = u_{N-2}^n - 2u_{N-1}^n + 2u_{N+1}^n \tag{2.10}$$

Equation 2.9 and 2.10 show the calculation of virtual grid points used for a free boundary at the end point located at N. The same approach can be used for a free boundary at the end point 0.

Chapter 3

Physical model

This chapter will cover the design of the physical model used for the real-time implementation. A brief overview of the Rhodes is given followed by more detailed sections showcasing the individual parts and how they are modelled.

3.1 Rhodes overview

The Rhodes piano has gone through several iterations since its the mk1 was released in 1965. It was initially released under the Fender brand as the 'Fender Rhodes' which was later dropped in favor of simply calling it 'Rhodes'. The Rhodes has been released in 88-key, 73-key and 54-key variants. Today the Rhodes piano is produced by Rhodes Music Group Ltd. who is currently producing the mk8 model [11]. While the Rhodes piano has seen many changes over the years, the core design behind its sound production remain the same. This design is extensively documented in the service manual of the mk1 [12] and will be the basis for the design choices of the physical model developed here.

Unlike a traditional piano that uses strings as its resonator, the Rhodes piano uses an original tuning fork design shown in figure 3.1. This tuning fork consists of two parts, the tone bar and the tine. The two parts are of significantly different masses and dimensions but share the same pitch. The purpose of the tone bar is to add timbre to the initial attack and increase the sustain of the vibrations of the tine. The tine, when struck by a hammer, will vibrate in an arc at the tip. The vibrations of the tine are then capture by a pickup (a magnet wrapped in a coil) as shown in 3.2, which is then sent to an external amplifier and played through a speaker. A damper holds the tine in place when the key is not pressed down.

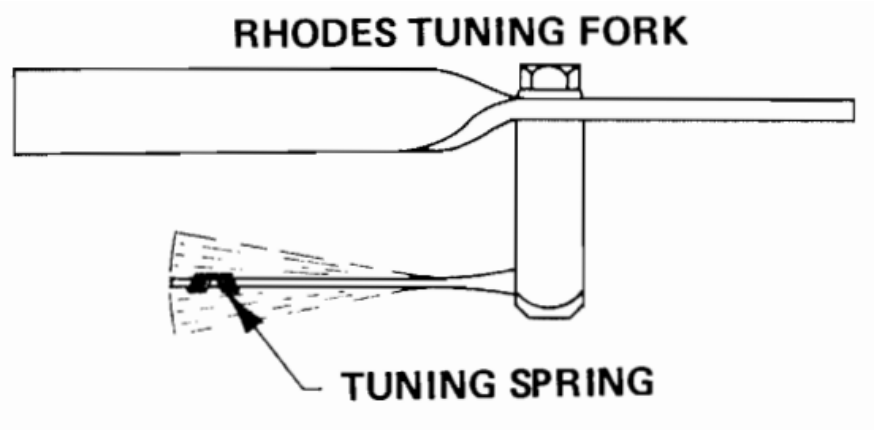


Figure 3.1: The tuning fork design used in the Rhodes piano as shown in the Rhodes service manual [12]. The tuning fork consists of an upper part called the tone bar, to which a metal block with a protruding bar, called the tine, is mounted. A small spring that can slide along the tine is used for fine tuning the fundamental frequency of the tuning fork assembly.

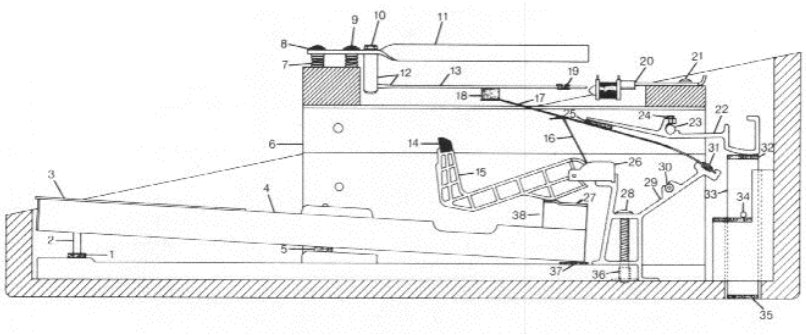


Figure 3.2: The full mechanism for sound production of a single key in the Rhodes piano as shown in the Rhodes service manual [12].

3.2 Tine

The tuning fork assembly used in the Rhodes can be described as two metal bars[1] sharing a rigid connection. Each bar is clamped on one end with the other end free. As shown in figure 3.1, the tine has a slight increase in circumference near its base. This addition of mass is there for durability reasons. For purposes of this model the tine is assumed to be uniform across its length. The transverse vibrations of such a bar can be described with a PDE [13, 4] known as the Euler-Bernoulli model and is used to model the tine of the Rhodes in [14]. In another work on modelling the Rhodes [15], the tine is modelled using the shear beam model. The shear beam model is an extension of the Euler-Bernoulli model, and is shown to more accurately approximate the fundamental frequency by taking into account the shearing effect of the beam [16]. The advantage of the Euler-Bernoulli model is that its corresponding FD scheme can be implemented explicitly, making it suitable for a real-time implementation. The FD scheme for the Euler-Bernoulli bar has the form

$$\rho A \delta_{tt} u_l^n = -EI \delta_{xxxx} u_l^n \tag{3.1}$$

Here E is Young's modulus, I is moment of inertia, ρ is material density and A is cross-sectional area of the bar. For a round bar, its moment of inertia I and cross-sectional area A are calculated as

$$A = \pi r \quad and \quad I = \frac{\pi r^4}{4}$$

The FD Scheme in equation 3.1 is lossless. To add losses, the scheme is extended by two terms

$$\rho A \delta_{tt} u_l^n = -EI \delta_{xxxx} u_l^n - 2\sigma_0 \rho A \delta_{t\cdot} u_l^n + 2\sigma_1 \rho A \delta_{t-} \delta_{xx} u_l^n \tag{3.2}$$

Here σ_0 will add frequency independent damping and σ_1 will add frequency dependent damping. Using Von Neumann Analysis [4], equation 3.2 can be shown to be stable under the condition

$$h \geq \sqrt{\frac{-4\sigma_1 k + \sqrt{(-4\sigma_1 k)^2 16\kappa^2 k^2}}{2}} \tag{3.3}$$

where κ is a stiffness coefficient defined as

$$\kappa = \sqrt{\frac{EI}{\rho A}} \tag{3.4}$$

A proof of this is given in Appendix A. Ideal grid spacing is achieved when h equals the stability condition.

Under the assumption that the tone bar is uniform, the FD scheme in equation 3.2 could be used to model its behaviour. The two schemes would then share a rigid connection, transferring energy back and forth between the two. Given that the two damping

[1] The terms "bar", "beam" and "rod" will be used interchangeably as different models use different terms.

Name	Symbol (unit)	Value
Length	$L\ (m)$	
Radius	$r\ (m)$	0.0005
Material Density	$\rho\ (kg/m^3)$	7850
Young's Modulus	$E\ (Pa)$	$2 * 10^{11}$
Frequency independent damping	$\sigma_0\ (s^{-1})$	0.0001
Frequency dependent damping	$\sigma_1\ (m^2/s)$	0.005

Table 3.1: Material parameters overview for the tine model. As length is calculated dynamically its value is not listed, instead see section 3.2.1

parameters allow for control over the sustain of the tine and its attack (through damping of high frequencies), the model can instead be simplified by using these parameters to model the influence of the tone bar. This reduces the amount of grid points that needs to be updated significantly and reduces the computational complexity of the overall model. Values and units for all bar parameters are presented in table 3.1.

Update equation

The FD Scheme in equation 3.2 can be expanded using the FD operators presented in chapter 2.2. Expanding all the derivatives and collecting the terms yields

$$\begin{aligned}(1+\sigma_0 k)u_l^{n+1} = &\Big(2-6\mu^2-\frac{4\sigma_1 k}{h^2}\Big)u_l^n \\ &+\Big(4\mu^2+\frac{2\sigma k}{h^2}\Big)(u_{l+1}^n+u_{l-1}^n) \\ &-\mu^2(u_{l+2}^n+u_{l-2}^n) \\ &+\Big(-1+\sigma_0 k+\frac{4\sigma_1 k}{h^2}\Big)u_l^{n-1} \\ &-\frac{2\sigma_1 k}{h^2}(u_{l+1}^{n-1}+u_{l-1}^{n-1})\end{aligned} \tag{3.5}$$

where

$$\mu = \frac{\kappa k}{h^2}$$

is a term used for compactness. Following a simple division, this equation can be used calculate a grid point at the next time step. The number of grid points needed for a given tine is calculated as

$$N = \lfloor L/h \rfloor \tag{3.6}$$

Where L is the length of the tine and h is defined by equation 3.3. The brackets in equation 3.6 indicate a flooring operation. As a flooring operation is used, h must be recalculated once N is calculated as

$$h = L/N \tag{3.7}$$

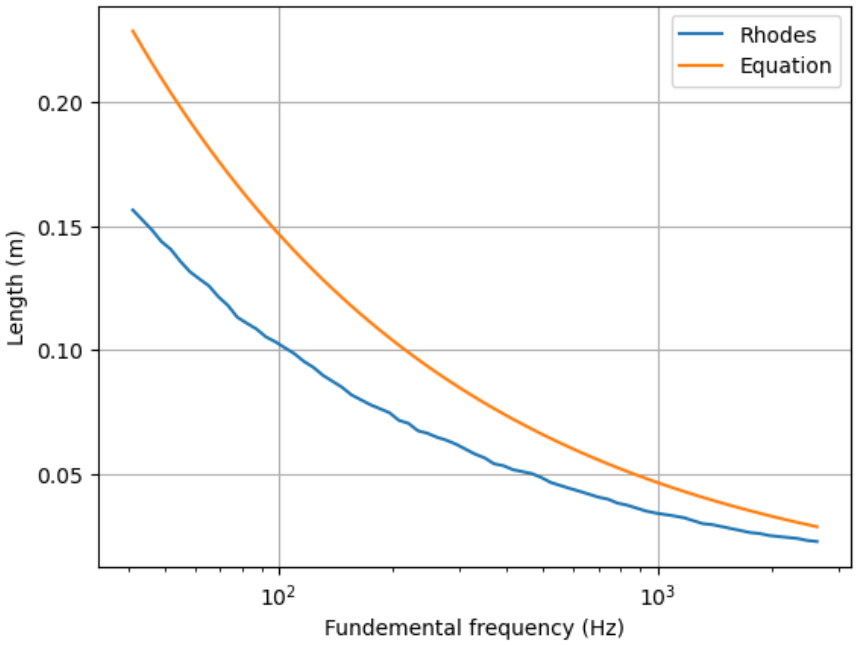


Figure 3.3: A comparison of the lengths of tines as a function of fundamental frequency using the measurements from the Rhodes manual and lengths calculated using equation 3.8. The x-axis is logarithmic.

3.2.1 Tuning

The Rhodes service manual [12] details the individual lengths of all the tines through an included cutting chart. From lowest to highest note of the 73-key model, the tines have a length in the range $\{0.1564, ..., 0.0226\}$ (in meters). The fundamental frequency f_1 of a bar with circular cross-section and clamped/free boundaries can be calculated [13, 17] as

$$f_1 = 1.426 \frac{\pi(r/2)}{8L^2} \sqrt{\frac{E}{\rho}} \tag{3.8}$$

Using this equation it can be shown that the length of the tines in the Rhodes are too short to achieve their target fundamental frequency (see figure 3.3). This highlights the importance of the additional mass added to the tip of the tine by the tuning spring. In [15], this additional mass from the tuning spring is modeled as a varying cross section. A varying cross section can be implemented by having A and I be spatially varying across the FD Scheme [4]. This needs to be accounted for in the grid spacing, where h is now determined by the points of the bar that produces the largest value according to the stability condition. This grid spacing will be less accurate for the remaining scheme, which will produce a low-passing effect. This can be solved for by the use of spatially stretched coordinates [4]. A different and much simpler approach which is used in [14] is to increase the length of

the tines to bring them in tune. In either case, manual fine-tuning is likely required for the individual tines. Due to its simplicity, the latter approach of increasing the lengths of the tines will be used for this model.

3.2.2 Boundary conditions

As mentioned earlier, the boundary conditions for the tine can be described with a clamped and free boundary. Using these boundary conditions the update equation shown in equation 3.5 would run for the grid points in the range $\{2, ..., N\}$. Due to the 4th order spatial derivative, the scheme needs an additional 2 grid points outside of this range. These grid points can be accessed using virtual grid points defined in equation 2.9 and 2.10.

The mixed derivative for the frequency dependent damping requires an additional grid point at the previous time step for grid point N. It can be shown that accessing this grid point using virtual grid point from equation 2.9 causes the scheme to become unstable. To solve this, a different approach for calculating the free boundary is used. The FD operators presented in 2.2 can also be written in matrix form [10]. FD operators in matrix form have not been discussed until now, as they are computationally slower than recursive update equations, making them less suitable for real-time implementations. They are however useful for deriving a different solution to calculating the update equation for the free boundary, as they do not rely on virtual grid points. Using matrix operators to find coefficients for update equation 3.5 at grid points $N-1$ and N yields.

$$\begin{aligned}
(1+\sigma_0 k)u_{N-1}^{n+1} &= (2-5\mu^2-\frac{4\sigma_1 k}{h^2})u_{N-1}^n \\
&\quad + (\frac{2\sigma_1 k}{h^2}2\mu^2)u_N^n \\
&\quad + (4\mu^2+\frac{2\sigma_1 k}{h^2})u_{N-2}^n \\
&\quad - \mu^2 u_{N-3}^n \\
&\quad + (-1+\sigma_0 k+\frac{4\sigma_1 k}{h^2})u_{N-1}^{n-1} \\
&\quad - \frac{2\sigma_1 k}{h^2}(u_{N-2}^{n-1}+u_N^{n-1})
\end{aligned} \tag{3.9}$$

$$\begin{aligned}
(1+\sigma_0 k)u_N^{n+1} &= (2-2\mu^2-\frac{4\sigma_1 k}{h^2})u_N^n \\
&\quad + (4\mu^2+\frac{4\sigma_1 k}{h^2})u_{N-1}^n \\
&\quad - 2\mu^2 u_{N-2}^n \\
&\quad + (-1+\sigma_0 k+\frac{4\sigma_1 k}{h^2})u_N^{n-1} \\
&\quad - \frac{4\sigma_1 k}{h^2}u_{N-1}^{n-1}
\end{aligned} \tag{3.10}$$

A more detailed derivation of these boundaries is shown in appendix B.

3.2.3 Dynamic grid

Initially presented in [5] and later expanded upon in [6], the dynamic grid is a method that allows for smooth sub-audio rate parameter changes to a FD scheme. This is highly useful in the context of MPE as new layers of expression can be added by mapping controls to material parameters of the model. The method involves splitting the grid into two separate grids.

$$\delta_{tt} u_l^n = -(\kappa^2)^n \delta_{xxxx} u_l^n - 2\sigma_0^n \delta_{t\cdot} u_l^n + 2\sigma_1^n \delta_{t-} \delta_{xx} u_l^n \tag{3.11a}$$

$$\delta_{tt} w_l^n = -(\kappa^2)^n \delta_{xxxx} w_l^n - 2\sigma_0^n \delta_{t\cdot} w_l^n + 2\sigma_1^n \delta_{t-} \delta_{xx} w_l^n \tag{3.11b}$$

Superscript ^{n} has been added to material parameters to show that they are time-varying. These new grids are defined as $l_u \in \{0, ..., M_u^n\}$ and $l_w \in \{0, ..., M_w^n\}$ where

$$M_u^n = \lceil 0.5N^n \rceil \quad and \quad M_w^n = \lfloor 0.5N^n \rfloor \tag{3.12}$$

are the number of grid intervals for each grid. This gives $M_u^n + M_w^n + 2$ grid points, which is one more than what is defined in chapter 2.1. The purpose of this additional grid point is in the connection between the two grids. The two grids will be connected at $u_{M_u}^n$ and w_0^n also refereed to as the inner boundaries. The distance between these to grid points is then defined by the fractional part of N before the flooring operation defined as

$$\alpha = \frac{L}{h} - N^n \tag{3.13}$$

In the case that $\alpha = 0$ the grid points $u_{M_u}^n$ and w_0^n will have identical values and function as a rigid connection between the two grids.

Addition and removal of grid points

When a parameter is changed such that $N^n \neq N^{n-1}$ the grid must be adjusted. The original grid can be split in two at any point along the grid, as longs as there are at least one moving grid point in the smaller subset. The approach used in [6], splits the original grid such that second grid only contains the minimum moving grid points needed. This simplifies addition and removal of grid points as these can be done exclusively to the first grid. For some tines, the Rhodes has a hammer contact location location is near the boundary. To avoid conflicts with these locations, this model will have the split in the center of the original grid as done in [5].

The addition of grid of grid points is done alternating between the two grids and is done when $N^n > N^{n-1}$. A new grid point is calculated by interpolating of the grid points at the inner boundaries. This is done using cubic Lagrangian interpolation vector

$$\mathbf{I} = \Big[-\frac{\alpha(\alpha+1)}{(\alpha+2)(\alpha+3)} \quad \frac{2\alpha}{\alpha+2} \quad \frac{2}{\alpha+2} \quad -\frac{2\alpha}{(\alpha+3)(\alpha+2)} \Big] \tag{3.14}$$

and transposed vector

$$\mathbf{v}^n = [u^n_{M^n_u-1} \quad u^n_{M^n_u} \quad w^n_0 \quad w^n_1]^T \tag{3.15}$$

Using definitions for the two grids

$$\mathbf{u}^n = [u^n_0, ..., u^n_{M^n_u}], \quad and \quad \mathbf{w}^n = [w_0, ..., w_{M^n_w}] \tag{3.16}$$

Additions of a new point can then be described as

$$\mathbf{u}^n = [\mathbf{u}^n, \mathbf{I}\mathbf{v}^n] \tag{3.17a}$$

$$\mathbf{w}^n = [\mathbf{I}^{\leftarrow}\mathbf{v}^n, \mathbf{w}^n] \tag{3.17b}$$

Here $\mathbf{I}^{\leftarrow}$ is a flipped version of the vector in equation 3.14.

Removal of grid is relatively straight forward and is done when $N^n < N^{n-1}$. Similar to the addition of grid points, this is done alternating between the two grids and is defined as

$$\mathbf{u}^n = [u^n_0, ..., u^n_{M^n_u-1}] \tag{3.18a}$$

$$\mathbf{w}^n = [w^n_1, ..., w_{M^n_w}] \tag{3.18b}$$

Update equations

Presented in [6] are two FD operators in matrix form, that contain quadratic interpolation values for calculating the spatial derivatives at the inner boundaries. Using the same methodology as for the free boundary in chapter 3.2.2, one can arrive at update equations for the individual grid points at the inner boundaries. As these equations are rather large, they will not be included here, but are instead shown in appendix C along with a detailed derivation.

3.3 Hammer

Early designs of the Rhodes used traditional 'tear drop' piano hammers [12]. This type of hammer compresses during contact with the resonator. The compression rate of the hammer is dependent on hammer velocity, which introduces hysteresis. The dynamic of such a hammer can be modelled with the Stulov model [18, 19]. The 'tear drop' design was dropped due to the hammers deforming over time and was replaced by a simpler hammer design using neoprene hammer tips (see figure 3.2). An impact model introduced by Hunt & Crosley in [20] and later used for modelling hammer-resonator impacts [21, 22], offers an alternative to the Stulov model and has been shown to yield good results for modelling the neoprene hammer impacts in the Rhodes [15]. This model also includes a term for the compression rate of the hammer material which introduces hysteresis. For a model of the early hammer design the Stulov model may be more suitable, but as this model will use of the neoprene hammers and the Hunt & Crosley model is chosen here.

Name	Symbol (unit)	Value
Hammer mass	M_h (Kg)	0.011
Impact coefficient	α_h	2.8
Stiffness	k_h (N/m^{α_h})	$1.5 \cdot 10^{11}$
Max initial velocity	v_{in} (m/s)	4
Force damping weight	λ $(N \cdot s/m^{5/2})$	$9 \cdot 10^{10}$

Table 3.2: Material parameters overview for the hammer model.

The Hunt & Crosley model can be implemented as a mass with some initial velocity dependent on user input, that when colliding with the tine exerts a force. This collision uses a penalising method, where the hammer and tine will inter-penetrate and force is calculated based on difference in displacement during this penetration. This force is then applied to the tine and subtracted from the hammer. The hammer displacement is calculated as

$$\delta_{tt} u_h = -F \tag{3.19}$$

where F is force exerted and is defined by

$$F = \begin{cases} k_h x^{\alpha_h}(1 + \mu_h \delta_{t-} x) & x > 0 \\ 0 & x \leq 0 \end{cases} \tag{3.20}$$

with x being the difference in displacement of the hammer and the tine. k_h is the stiffness of the hammer tip, $\mu_h = \lambda/k$ is a mathematically convenient term, λ is force damping weight and α_h is a coefficient for the local geometry of the impact. Subscript $_h$ is used here to avoid confusion with previously mentioned material parameters. The scheme in equation 3.2 then has force applied as

$$\delta_{tt} u_l^n = -\kappa^2 \delta_{xxxx} u_l^n - 2\sigma_0 \delta_{t\cdot} u_l^n + 2\sigma_1 \delta_{t-} \delta_{xx} u_l^n + \epsilon_h \mathcal{M} F \tag{3.21}$$

where ϵ_h is a contact distribution over the grid and $\mathcal{M} = M_h/\rho AL$ is the mass ratio of the hammer and tine. The contact distribution here is simply modelled as a single grid point. The contact location varies from tine to tine as seen in [12] where lower notes have a contact location closer to the center of the tine and higher notes closer to the base of the tine. Additionally, the stiffness of the hammer tip increases for higher tines. Values and units for all hammer parameters are presented in table 3.2.

3.4 Damper

The damping mechanism in the Rhodes consists of a leaf spring with a small piece of felt at the end [12]. The damping mechanism is engaged when a key is released, unless the sostenuto pedal is held. One way of modelling the damping effect is through a collision applied over a subset of the grid points. This approach is similar to the hammer

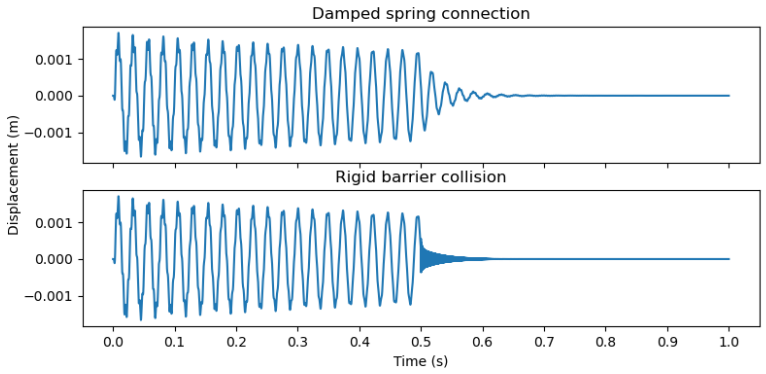


Figure 3.4: A comparison of using a damped spring connection and rigid barrier collision for damping of the tine. The plot shows the displacement of the tip of the tine over time. Damping is applied after 0.5 seconds.

in that it imposes a penalty when two objects inter-penetrate. The mass-spring collision [10] is suitable for modelling the spring behaviour of the damper, but the model requires matrix inversions, making it a computationally expensive solution ill-suited for real-time implementations. The simpler rigid-barrier collision [10] can be solved for explicitly, but produces audible artifacts when applied (see figure 3.4).

Due to these factors a different approach for modelling the damper is used. Instead of modelling a collision between the tine and the damper, the damper will be modelled as a connection between the tine and its equilibrium (i.e. 0) using a damped spring connection [10]. The connection is added to the FD scheme in equation 3.2 at the connection location as

$$\delta_{tt} u_l^n = -\kappa^2 \delta_{xxxx} u_l^n - 2\sigma_0 \delta_{t\cdot} u_l^n + 2\sigma_1 \delta_{t-} \delta_{xx} u_l^n - \frac{f^n}{h\rho A} \tag{3.22}$$

The damped spring connection is normally used to model a connection between two FD schemes, but in this case the second scheme will be replaced by the tines equilibrium. Force f^n is then calculated as

$$f^n = \frac{1}{2} \cdot \frac{u^\star + \frac{K_1}{2r_+^n}\eta^n + \frac{r_-^n}{r_+^n}\eta^{n-1}}{\frac{1}{r_+^n} + \frac{k^2}{h\rho A(1+\sigma_0 k)}} \tag{3.23}$$

with

$$r_+^n = \frac{K_1}{4} + \frac{K_3(\eta^n)^2}{2} + \frac{R}{2k}$$
$$r_l^n = \frac{K_1}{4} + \frac{K_3(\eta^n)^2}{2} - \frac{R}{2k}$$

where $u^\star$ is the tine calculated for the next time step without the connection force (equation 3.5). Relative displacement η is simply calculated as the displacement of the tine at

Name	Symbol (unit)	Value
Linear spring coefficient	K_1 (N/m)	100
Non-linear spring coefficient	K_3 (N/m^3)	100000
Damping coefficient	R (Kg/s)	0.5

Table 3.3: Material parameters overview for damped spring connection

the connection location. Linear and nonlinear spring coefficients K_1 and K_3 and damping coefficient R are parameters for the spring. Values and units for all damped spring connection parameters are presented in table 3.3.

3.5 Pickup

The Rhodes has individual pickups for each tine. The pickups are identical, but adjustment of tine position relative to the pickup will vary from key to key. The Rhodes manual [12] shows that the ideal position is at an offset from the center of the pickup and correct spacing is required for ideal volume. The non-linear behaviour of pickups is complex and various methods are proposed for a physically based modelling of it (see section 3.5.1). Due to the complexity, a physically based model of the pickup is left for future work. Instead, this section will describe a different approach using a set of filters and nonlinear distortion based on [23].

Audio input for the pickup is read from grid point at the tip of the tine. This is then filtered through a low-pass filter. The filtered output is gained and clipped using the hyperbolic tangent function. The resulting clipped signal the goes through a wave-shaping process before being filtered with a high-pass filter.

3.5.1 Physically based alternatives

This section will propose some physically based alternatives to the pickup described in the previous section. The non-linear behaviour of pickups is complex and the models suggested here work under certain assumptions.

Work has been done modelling the Rhodes model specifically. A model first introduced in [24] and later expanded upon in [15] models the magnetic field of the pickup using Finite Element Methods (FEM). The induction of the tine is then formulated as a three-part integral over a simplified geometry of the bell-shaped Rhodes pickup. This model takes into account the offset of the tine relative to the magnet in the pickup, allowing for timbre adjustments as described in the Rhodes manual [12]. This modelling approach was shown to be accurate enough given proper parameters in [25] when compared against pickup recordings.

A more commonly modelled pickup is that of the electric guitar. There are many parallels between the pickup of the electric guitar and that of the Rhodes. As such some of the methods may be applicable for modelling the Rhodes pickup as shown in [14]. This

model is developed using Port-Hamiltonian modeling. Unlike the previously mentioned model, this does not account for the shape of the magnet. Instead the derived magnetic field is based off of the magnet and the coil winding it. The tip of the tine is then a sphere immersed in the magnetic field.

Of the two the first model is arguably better suited for this project as it is also shown to run in real-time.

Chapter 4

Real-time implementation

The physical model described in chapter 3 is implemented in a real-time application using the C++ framework JUCE [9] and is available online at [26]. The application functions as a VST plugin with support for MPE and MIDI input and can be used with a Digital Audio Workstation (DAW). This chapter will give a design overview providing details on key technical parts of the implementation, describe how MPE is mapped and highlight important optimizations.

4.1 Design overview

The physical model is implemented using the *MPESynthesiser* class from JUCE where the tines are considered the voices of the synth. By default the synth is configured to use 5 voices, which is chosen as a balance between playability and computational requirements. Audio is rendered block by block where each block is split into sub blocks separated by midi input. In each sub block samples are rendered from the physical model sample by sample.

Tine

The grid of for tine is implemented using vectors with space allocated according to the amount of required grid points. This is done for each needed time step for both halves of the grid leading to 6 state vectors. On user input, the length of the tine is determined by fundamental frequency of the MIDI note based on equation 3.8. The grid is then resized accordingly. As the grid size will wary on every new note and change in parameters, sufficient memory for each grid vector is reserved to prevent re-allocation of memory. Once all grid points have been updates, a pointer swap is utilized to reuse the allocated vectors.

The FD Scheme in equation 3.2 with clamped/free boundaries and dynamic grid requires grid to have at least 8 grid points. To get a grid size sufficient for this requirement

for the full range of notes, it is necessary to run a minimum of 16 times oversampling when using a standard sample-rate of $44100Hz$. This is implemented by having the system run at standard sample-rate but run the update equation for the scheme j times between each sample, where j is the oversampling multiplier. Time step k is then defined as $k = 1/f_s j$.

Pickup

The filters for the pickup are implemented as *StateVariableTPTFilter* from JUCE. These filters allow of user control at runtime. The initial low-pass filters cutoff frequency can be changed by the user to control the brightness of the attack. The cutoff frequency of high-pass filter at the end is set to the current notes fundamental frequency.

User controlled parameters

User controlled parameters are mapped to the Graphical User Interace (GUI) as relevant input controls (knobs, toggles, sliders etc.). This is attached to a *ValueTreeState* object which tracks the changes made by the user. The changes in state is checked at a rate of 60Hz with changes applied at the start of the next rendered audio block. In cases where changes could result in audio artifacts, the change is smoothed on a sample by sample basis over defined smoothing period. The smoothing is done either as a linear or multiplicative smoothing depending on the parameter.

MIDI Polyphonic Expression

The current implementation supports per key pitch bend through the use of the dynamic grid. This is mapped to the length of the tine and allows the user to bend each key either up or down one semi-tone. The length is calculated as in the beginning of a note by using equation 3.8. As no material parameters used in the stability condition are changed, the grid spacing does not need to be recalculated.

4.2 Optimizations

While it is possible to run the model described in chapter 3 in a real-time application as it is, optimizations can be made to allow for more voices and wider compatibility with computers with less computational power. The systems bottleneck will be at the CPU, so in many cases computations can be saved at the cost of additional memory. The additional memory used is at a scale where it is trivial on a modern computer. Key optimizations are highlighted here.

Pre-computed coefficients

It is beneficial to collect the terms of an update equation as done in equation 3.5. These can then be pre-computed outside of the update loop saving multiple mathematical operations. Equation 3.5 can be rewritten as

$$u_l^{n+1} = \Big(C_0 u_l^n + C_1(u_{l+1}^n + u_{l-1}^n) - M_0(u_{l+2}^n + u_{l-2}^n) + B_0 u_l^{n-1} - S(u_{l+1}^{n-1} + u_{l-1}^{n-1})\Big)D \quad (4.1)$$

Where the pre-computed coefficients are defined as

$$\begin{aligned} C_0 &= \Big(2 - 6\mu^2 - \frac{4\sigma_1 k}{h^2}\Big) \\ C_1 &= \Big(4\mu^2 + \frac{2\sigma k}{h^2}\Big) \\ M_0 &= \mu^2 \\ B_0 &= \Big(-1 + \sigma_0 k + \frac{4\sigma_1 k}{h^2}\Big) \\ S &= \frac{2\sigma_1 k}{h^2} \\ D &= \frac{1}{(1 + \sigma_0 k)} \end{aligned}$$

While this may initially seem like a trivial amount of mathematical operations saved, once scaled by the sample-rate, grid points and potential oversampling, this will become several thousands of operations saved per second. This approach is still valid even with the time-varying parameters introduced with the Dynamic grid, as these changes are sub audio rate and thus still benefit from being pre-computed outside the update loop. This approach is also used for the update equations for the free boundary and inner boundary as well as other mathmatical operations that may be pre-computed outside of the main audio loop.

Dynamic grid

As shown in equation 3.17 the addition of grid points will either be at the back of the state vector or in front, depending on which half is added to. While data structures that allow for efficient insertion at the front exist, a simpler approach would be to flip the indexes of the $\mathbf{w}^n$ vector, such that the addition logic is the same as with the $\mathbf{u}^n$ vector. The two vectors are now connected at the back. As update equation 3.5 is spatially symmetric it remains valid. The addition and removal logic for both state vectors can now be described as that of the data structure Stack [27] where grid points are added and removed at the back of the vector. The addition of grid points for either vector now follows

$$\mathbf{u}^n = [\mathbf{u}^n, \mathbf{I}\mathbf{v}^n] \quad (4.2a)$$

$$\mathbf{w}^n = [\mathbf{w}^n, \mathbf{I}(\mathbf{v}^n)^{\leftarrow}] \quad (4.2b)$$

with

$$\mathbf{v}^n = [u^n_{M^n_u-1} \quad u^n_{M^n_u} \quad w^n_{M^n_w} \quad w^n_{M^n_w-1}]^T \tag{4.3}$$

and removal is defined as

$$\mathbf{u}^n = [u^n_0, ..., u^n_{M^n_u-1}] \tag{4.4a}$$

$$\mathbf{w}^n = [w^n_0, ..., w^n_{M^n_w-1}] \tag{4.4b}$$

This approach allows for all state vectors to be implemented with the same data structure. If sufficient memory is reserved, memory for state vectors do not need to be reallocated, saving valuable computations. The inner boundary condition for the $\mathbf{w}^n$ vector and free boundary are spatially mirrored for this approach.

Sample-rate

As described in the design overview, the system needs to apply oversampling to achieve a grid spacing short enough to be able to model the full range of the tines in the Rhodes. While high oversampling is necessary for the shortest tines, the longer tines are able to run with lower or no oversampling. To achieve a relatively consistent level of grid points across the different notes and thereby avoid sudden increases in computational complexity, the system uses various levels of oversampling depending on the note played. This approach is valid for oversampling multipliers that are integer values.

Chapter 5

Evaluation

This chapter presents the evaluation methodology and results of the physical model and real-time implementation developed. The evaluation is done in three parts covering the sound production, tone production and addition of MPE. Sound production and tone production are evaluated through output comparisons and analysis and MPE is evaluated based on interviews with users. The results of the evaluation will be discussed in chapter 6.

5.1 Sound production

The purpose of this evaluation is to investigate how accurate the physical model of the tine and hammer are in reproducing the behavior of the Rhodes sound production. The time-domain output measured at the tip of the tine will be compared against the results found in [28], where measurements of the tine of a Rhodes mk1 were made using a piezo and a high speed camera. The frequency-domain output will be compared against that of [25] where analysis of the frequency contents of the tine were measured using a SLDV technique. The note F1 with fundamental frequency 43.65 is chosen to match the one measured in [25]. The note C4 (middle C) will additionally be shown for insight into how center frequency notes behave in comparison to lower notes. Both notes use $2(m/s)$ hammer input velocity.

Figure 5.1 shows the F1 tine over a period of 5 seconds. As noted in [28] the output of a Rhodes tine is approximately sinusoidal shortly after excitation, which is not the case here. Looking at the output frequencies as shown in figure 5.2a reveals the presence of 2 overtones, which last for the entire duration. The presence of two overtones matches that of [25] where overtones are present at ratios 7.11 and 20.25. However in the case of this model the overtones are at a ratio of 6.37 and 35.27. Comparing to a higher frequency note shown in figure 5.2b the presence of the second overtone is significantly reduced. Both notes are also shown to be significantly out of tune indicating that equation 3.8 is not valid for this model and a different methodology for estimating length is required.

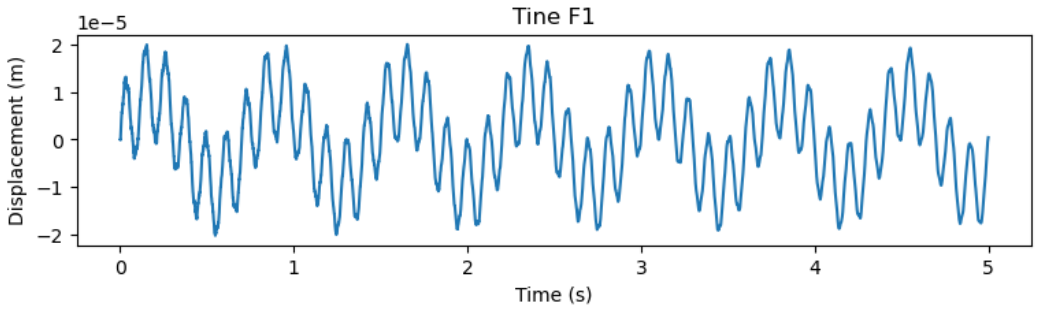
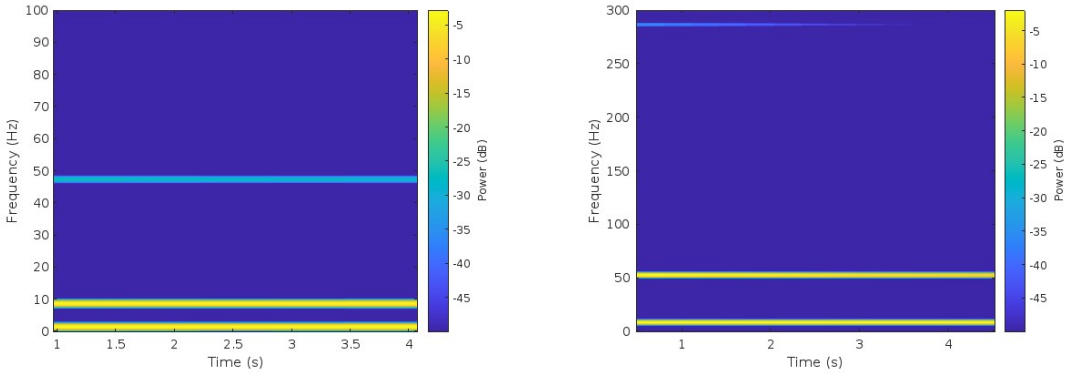


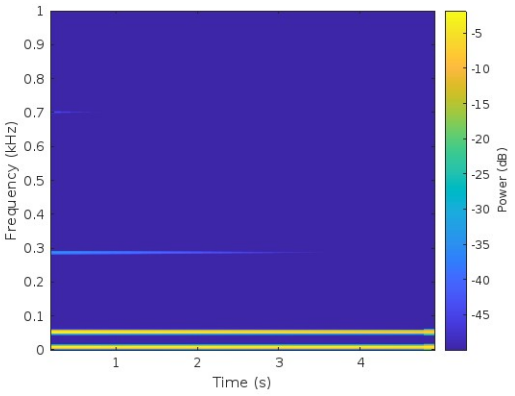
Figure 5.1: Time domain representation of the displacement of the tip of the tine corresponding to note F1 (43.65Hz) following hammer excitation.



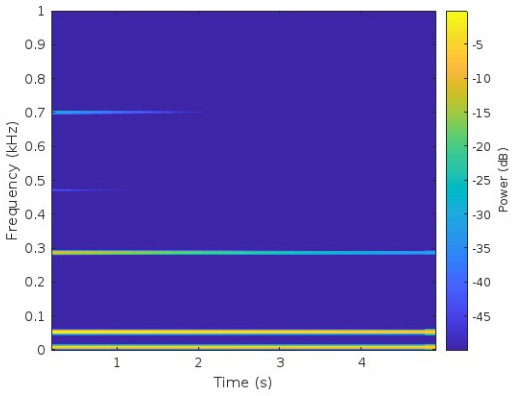
(a) Tine F1 power spectrum

(b) Tine C4 power spectrum

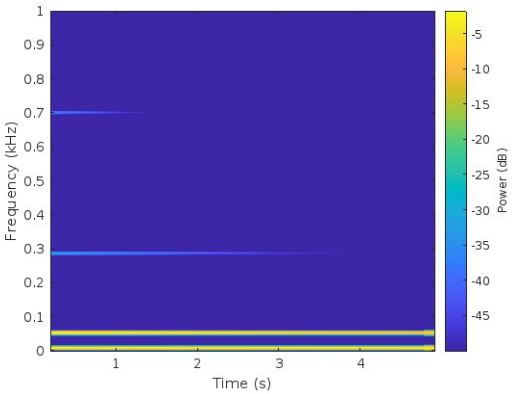
Figure 5.2: Power spectrum of tine F1 (43.65Hz) and tine C4 (261.63Hz).



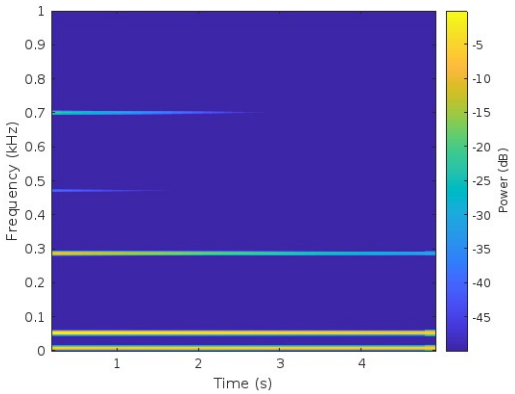
(a) $k = 1.5 \cdot 10^{11}$ at $0.4L$



(b) $k = 1.5 \cdot 10^{11}$ at $0.2L$



(c) $k = 1.5 \cdot 10^{12}$ at $0.4L$



(d) $k = 1.5 \cdot 10^{12}$ at $0.2L$

Figure 5.3: Power spectrum of tine C4 using different hammer tip stiffness and contact locations.

Hammer parameters

As noted in the Rhodes manual [12], the hammer impact location and hammer tip stiffness has great impact on the sound production. To investigate this, a comparison of 4 different hammer configurations for the note C4 are simulated.

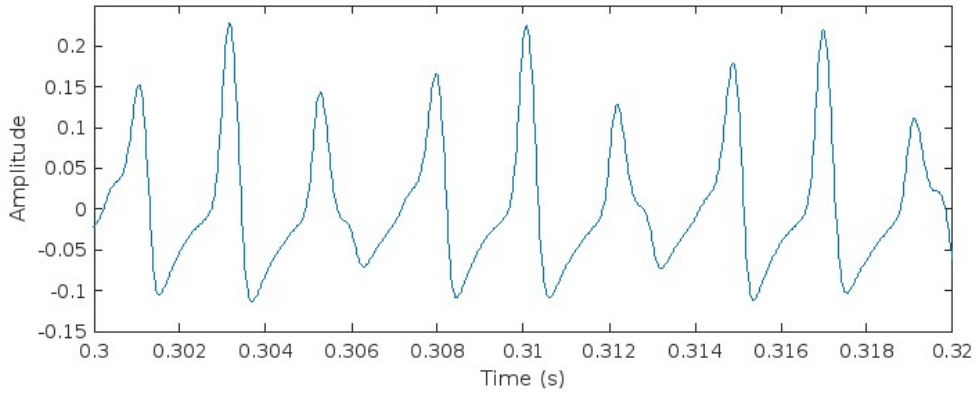
The timbre most notably changes with hammer contact location as seen in figure 5.3. Both number and power of overtones produced from the hammer impact increase as the contact location moves closer to the base of the tine. A similar behavior can be seen when increasing the hammer tip stiffness though the effect is not as pronounced.

The importance of correct hammer impact location and stiffness is particularly seen on higher notes. For very short tines a contact location close to the base with a significantly stiff hammer tip is required for the tine to produce any output. Improper configuration will lead to too long of a contact time, which results in the hammer both transferring force but also damping the tine.

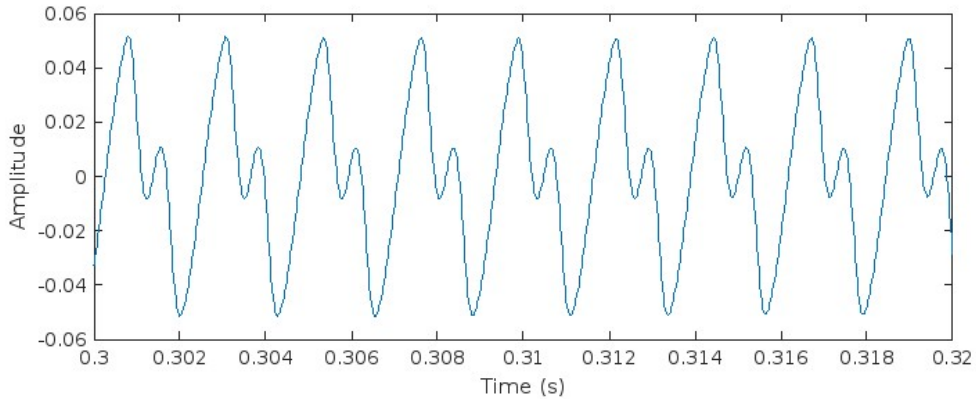
5.2 Tone production

The purpose of this evaluation is to assess how well the model is able to reproduce the tones of a Rhodes piano. The output of the model will be compared against a Rhodes piano using the Rhodes V8 plugin [1]. The Rhodes V8 plugin is set to the *MK8 Factory Default Setup* profile and output is generated using an A4 (440Hz) midi note with input velocity set to 100. This velocity is chosen to produce an output with a notable attack but without the note distorting. A corresponding output from the model developed here is generated using the same midi note as input. The model is tuned manually for this note, due to the tuning issues shown in the previous section.

As seen in figure 5.4 the output waveform of the pysical model differs significantly from that of the Rhodes V8. The Rhodes V8 shows a consistently repeating period while the physical model repeats every three periods. Looking at the frequencies present in the signals as shown in figure 5.5, the Rhodes V8 has a strong fundamental frequency with harmonic overtones. The physical model shows the same harmonic overtones but with longer sustain, but also many in-harmonic overtones not present in the Rhodes V8.

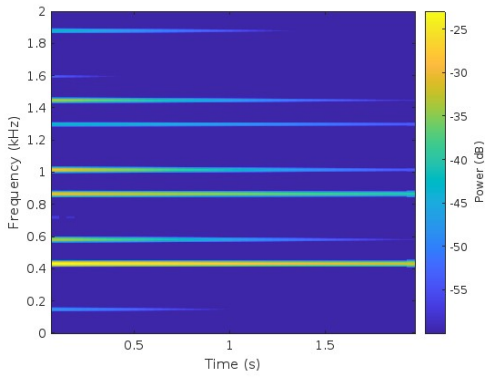


(a) Physical model

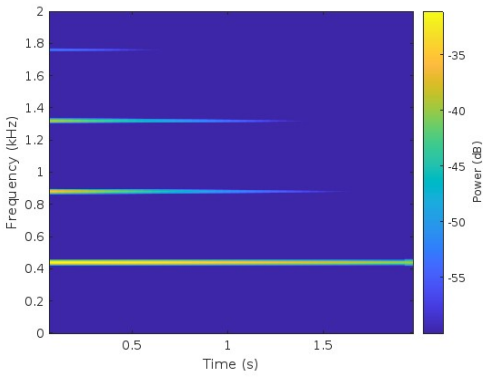


(b) Rhodes V8

Figure 5.4: Waveform of the physical model and the Rhodes V8 plugin for the note A4 (440Hz).



(a) Physical model



(b) Rhodes V8

Figure 5.5: Power spectrum of the A4 note for the physical model and the Rhodes V8 plugin for the note A4 (440Hz).

5.3 User interviews

A preliminary set of interviews were conducted to evaluate the current implementation and mapping of MPE functionality. This was done with 2 users experienced with audio plugins and DAWs. The users were introduced to the plugin and its mappings and afterwards given time to experiment with the plugin. The plugin was running on a laptop using the built in speakers and a Roli Lumi Keys was used as MPE interface.

Both users described the pitch bend as smooth and natural sounding though found the base note and max pitch bend to be out of tune. When using the lumi keys for pitch bend one user found the range inadequate. The other found the range to be decent, but would like further range for when working with a DAW. Both users stated that the addition of pitch bend would be useful to them for music production. When asked about additional input controls one user noted that mapping tone control to after touch would be interesting.

Chapter 6

Discussion

The results of the different evaluations done in chapter 5 show a significant deviation from the Rhodes. While the model shows some similarities, many changes could be made to achieve a better output. This chapter will discuss the results of the evaluation and propose additions and modifications to the physical model developed.

6.1 Sound production

The behaviour of the model is shown to have notably longer sustain than that of the Rhodes. Particularly the overtones present in the tine last for longer than that found in studies of the Rhodes. While this could be solved to some extent by introducing more frequency dependent damping, this could also indicate improper hammer contact location. Hammer contact location and hammer stiffness is shown to correlate to the presence of higher modes, with the former being the most influential. Proper hammer configuration and frequency dependent damping could bring the behaviour of the tine more in line with that of the Rhodes. An addition to the model could be the introduction of a third damping term. Adding this would change equation 3.2 to

$$\rho A \delta_{tt} u_l^n = -EI\delta_{xxxx} u_l^n - 2\sigma_0 \rho A \delta_{t\cdot} u_l^n + 2\sigma_1 \rho A \delta_{t-} \delta_{xx} u_l^n - 2\sigma_2 \rho A \delta_{t-} \delta_{xxxx} u_l^n \qquad (6.1)$$

Where σ_2 is an additional frequency dependent damping term for more control over higher frequencies.

Tuning

As seen in the analysis of the sound production, the tine is significantly out of tune. When checked against a tuner in Ableton Live, this is the case across all notes, where they have a lower fundamental frequency than desired. A new method for determining the length of the tines is required. This could be done through the use of modal analysis [10, 4]. This

form of analysis gives objective data on the resonant frequencies present in the scheme, which can be used for tuning the system.

6.2 Tone production

Aside from the tuning issues described earlier, the pickup model is arguably where the model deviates from the Rhodes the most. The harmonic overtones are similar to that of the Rhodes but with a notable difference how long they are sustained for. The main difference is in the many in harmonic overtones not present in the Rhodes, which leads to a much harsher sound in comparison. An issue with the pickup model is that it is likely dependent on the input being close to a sine-wave as seen with the original implementation [23]. Given that the model for the tine developed here includes overtones, that likely causes issues with this pickup model.

As shown in [28] the pickup is the most important to the sound of the Rhodes. For future work the model developed here would benefit greatly from a physically modelled pickup such as the one developed in [15].

6.3 MPE

The feedback on the implementation of MPE was positive with both users considering the added functionality useful. With the implementation of the dynamic grid the addition of new mappings is fairly straight forward with the main work being in designing the new mappings. With the current implementation of the pickup and the mention of a desire for tone control, one option for a new mapping would be to map after-touch to the cut-off frequency for the low pass filter in the pickup model. The range of the pitch bend could easily be increased through configuration, though this increase may result in audible artifacts. As stated in [5, 6] the removal of grid points can introduce high frequency artifacts. By increasing the pitch bend range it will be possible for the user to remove several grid points which increases the likelihood of artifacts appearing. A solution proposed in [5] is introducing displacement correction when removing a grid point, by calculating a correction force with a damped spring connection. An alternative solution proposed in [6] is to introduce frequency dependent damping term as shown in equation 6.1, to dampen the high frequency content introduced from the grid point removal. Given that the additional damping term would already benefit the model this solution may be more relevant.

Chapter 7

Conclusion

This project has designed and implemented a real-time physical model of the Rhodes piano as an audio plugin for use in Digital Audio Workstations. The physical model uses finite-difference time-domain methods with the dynamic grid and maps physical parameters to MPE which adds pitch bending to the instrument. Preliminary user interviews show interest in the addition of pitch bending and additional mappings are proposed based on the feedback given. The evaluation of the physical model shows issues with tuning and a different behaviour of the model for the tine when compared against other studies. Hammer parameters are shown to play an important role in sound production and the model would benefit from additional tuning of these parameters. Modal analysis and an extension to the model used for the tine is proposed to solve the tuning issues and difference in behaviour. The non-physical model of the pickup produces in-harmonic overtones that are not present in the Rhodes. The model for the pickup would need to be physically modelled to produce better results than the current implementation.

Bibliography

[1] *Rhodes Music Group Ltd.* URL: https://rhodesmusic.com/rhodes-v8-plugin/.

[2] John M. Chowning. "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation". In: (1977).

[3] Julius O. Smith. "Virtual Acoustic Musical Instruments: Review and Update". In: (2010).

[4] Stefan Bilbao. *Numerical Sound Synthesis*. Wiley, 2009.

[5] Silvin Willemsen et al. "Dynamic Grids for Finite-Difference Schemes in Musical Instrument Simulations". In: *24th International Conference on Digital Audio Effects (DAFx20in21)* (2021).

[6] Silvin Willemsen and Stefania Serafin. "Real-Time Implementation of the Dynamic Stiff String using Finite-Difference Time-Domain Methods and the Dynamic Grid". In: *25th International Conference on Digital Audio Effects (DAFx20in22)* (2022).

[7] The MIDI Association and Association of Musical Electronics Industry. *MIDI Polyphonic Expression*. 2022.

[8] *MIDI Polyphonic Expression*. URL: https://roli.com/mpe.

[9] *JUCE*. URL: https://juce.com/.

[10] Silvin Willemsen. "The Emulated Ensemble". PhD thesis. Aalborg University Copenhagen, 2021.

[11] *Rhodes Music Group Ltd.* URL: https://rhodesmusic.com/.

[12] Rhodes Keyboard Instruments. *Rhodes Service Manual*. URL: https://www.vintagevibe.com/pages/service-manuals.

[13] Neville H. Fletcher and Thomas D. Rossing. *The physics of musical instruments*. Springer, 1998.

[14] Antoine Falaize and Thomas Hélie. "Passive simulation of the nonlinear port-Hamiltonian modeling of a Rhodes Piano". In: *Journal of Sound and Vibration* (2017).

[15] Florian Pfeifle. "Real-time physical model of Wurlitzer and Rhodes electric piano". In: *20th International Conference on Digital Audio Effects (DAFx-17)* (2017).

[16] Seon M. Han, Haym Benaroya, and Timothy Wei. "Dynamics of transversely vibrating beams using four engineering theories". In: *Journal of Sound and Vibration 1999* (1999).

[17] Neville H. Fletcher and Thomas D. Rossing. *Principles of Vibration and Sound, 2nd edition*. Springer, 2004.

[18] Anatoli Stulov. "Hysteretic model of the grand piano hammer felt". In: *Journal of the Acoustical Society of America* (1995).

[19] Anatoli Stulov. "Experimental and computational studies of piano hammers". In: *Acta Acustica united with Acustica* (2005).

[20] Kenneth Hunt and Erskine Crossley. "Coefficient of restitution interpreted as damping in vibroimpact". In: *Journal of Applied Mechanics* (1975).

[21] Frederico Avanzini and Davide Rocchesso. "Physical modelling of impacts: Theory and experiments on contact time and spectral centroid". In: *Journées d'informatique musicale, 2004, Paris, France* (2004).

[22] Frederico Avanzini and Davide Rocchesso. "Controlling Material Properties in Physical Models of Sounding Objects". In: (2001).

[23] Mike Moreno DSP. *EP-MK1 Documentation*. URL: `https://mikemorenodsp.github.io/documentation-ep-mk1/`.

[24] Florian Pfeifle and Malte Münster. "Tone Production of the Wurlitzer and Rhodes E-Pianos". In: *Studies in Musical Acoustics and Psychoacoustics*. 2017, pp. 75–108.

[25] Leonardo Gabrielli et al. "The Rhodes electric piano: Analysis and simulation of the inharmonic overtones". In: *Journal of Acoustical Society of America 148* (2020).

[26] *Real-time implementation*. URL: `https://github.com/TSonderbo/EPiano`.

[27] Thomas H. Cormen et al. *Introduction to Algorithms, Third edition*. The MIT Press, 2009.

[28] Malte Muenster and Florian Pfeifle. "Non-Linear Behaviour in Sound Production of the Rhodes Piano". In: *The Journal of the Acoustical Society of America* (2014).

Appendix A

Bar analysis

Update equation

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

Divide by ρA

$$\delta_{tt} u_l^n = -\kappa^2 \delta_{xxxx} u_l^n - 2\sigma_0 \delta_{t\cdot} u_l^n + 2\sigma_1 \delta_{t-} \delta_{xx} u_l^n$$

expanding using FD operators

$$\begin{aligned}\frac{1}{k^2}(u_l^{n+1} - 2u_l^n + u_l^n - 1) = &\frac{\kappa^2}{h^4}(u_{l+2}^n - 4u_{l+1}^n + 6u_l^n - 4u_{l-1}^n + u_{l-2}^n) \\ &- \frac{2\sigma_0}{2k}(u_l^{n+1} - u_l^{n-1}) \\ &+ \frac{2\sigma_1}{kh^2}(u_{l+1}^n - 2u_l^n + u_{l-1}^n - u_{l+1}^{n-1} + 2u_l^{n-1} + u_{l-1}^{n-1})\end{aligned}$$

multiply by k^2

$$\begin{aligned}u_l^{n+1} - 2u_l^n + u_l^{n-1} = &\frac{\kappa^2 k^2}{h^4}(u_{l+2}^n - 4u_{l+1}^n + 6u_l^n - 4u_{l-1}^n + u_{l-2}^n) \\ &- 2\sigma_0 \frac{k}{2}(u_l^{n+1} - u_l^{n-1}) \\ &+ \frac{2\sigma_1 k}{h^2}(u_{l+1}^n - 2u_l^n + u_{l-1}^n - u_{l+1}^{n-1} + 2u_l^{n-1} + u_{l-1}^{n-1})\end{aligned}$$

add $2u_l^n - u_l^{n-1}$

$$\begin{aligned}u_l^{n+1} = &\frac{\kappa^2 k^2}{h^4}(u_{l+2}^n - 4u_{l+1}^n + 6u_l^n - 4u_{l-1}^n + u_{l-2}^n) \\ &- \sigma_0 k(u_l^{n+1} - u_l^{n-1}) \\ &+ \frac{2\sigma_1 k}{h^2}(u_{l+1}^n - 2u_l^n + u_{l-1}^n - u_{l+1}^{n-1} + 2u_l^{n-1} + u_{l-1}^{n-1}) + 2u_l^n - u_l^{n-1}\end{aligned}$$

isolate u_l^{n+1}

$$\begin{aligned}(1+\sigma_0 k)u_l^{n+1} &= \mu^2(u_{l+2}^n - 4 * u_{l+1}^n + 6u_l^n - 4u_l l - 1^n + u_{l-2}^n) \\ &- \sigma_0 k(-u_l^n - 1) \\ &+ \frac{2\sigma_1 k}{h^2}(u_{l+1}^n - 2u_l^n + u_{l-1}^n - u_{l+1}^{n-1} + 2u_l^{n-1} + u_{l-1}^{n-1}) + 2u_l^n - u_l^{n-1}\end{aligned}$$

collecting terms

$$\begin{aligned}(1+\sigma_0 k)u_l^{n+1} &= \Big(2 - 6\mu^2 - \frac{4\sigma_1 k}{h^2}\Big)u_l^n \\ &+ \Big(4\mu^2 + \frac{2\sigma k}{h^2}\Big)(u_{l+1}^n + u_{l-1}^n) \\ &- \mu^2(u_{l+2}^n + u_{l-2}^n) \\ &+ \Big(-1 + \sigma_0 k + \frac{4\sigma_1 k}{h^2}\Big)u_l^{n-1} \\ &- \frac{2\sigma_1 k}{h^2}(u_{l+1}^{n-1} + u_{l-1}^{n-1})\end{aligned}$$

with

$$\kappa = \sqrt{\frac{EI}{\rho A}} \quad \text{and} \quad \mu = \frac{\kappa k}{h^2}$$

Von Neumann stability analysis

Equation numbers from [10]. Using the definitions for spatial and temporal operators found in (eq. 3.23 and 3.22) we can obtain a frequency domain representation of the scheme

$$\frac{1}{k^2}(z - 2 + z^{-1}) = -\frac{16\kappa^2}{h^4}sin^4(\beta h/2) - \frac{2\sigma_0}{2k}(z - z^{-1}) + 2\frac{\sigma_1}{k}(1 - z^{-1})(-\frac{4}{h^2}sin^2(\beta h/2))$$

$$\frac{1}{k^2}(z - 2 + z^{-1}) = -\frac{16\kappa^2}{h^4}sin^4(\beta h/2) - \frac{\sigma_0}{k}z + \frac{\sigma_0}{k}z^{-1} - \frac{8\sigma_1}{kh^2}sin^2(\beta h/2) + \frac{8\sigma_1}{kh^2}sin^2(\beta h/2)z^{-1}$$

Rewriting equation

$$0 = \frac{1}{k^2}(z - 2 + z^{-1}) + \frac{16\kappa^2}{h^4}sin^4(\beta h/2) + \frac{\sigma_0}{k}z - \frac{\sigma_0}{k}z^{-1} + \frac{8\sigma_1}{kh^2}sin^2(\beta h/2) - \frac{8\sigma_1}{kh^2}sin^2(\beta h/2)z^{-1}$$

Multiply all terms by k^2

$$0 = z - 2 + z^{-1} + \frac{16\kappa^2 k^2}{h^4} sin^4(\beta h/2) + (\sigma_0 k)z - (\sigma_0 k)z^{-1} + \frac{8\sigma_1 k}{h^2} sin^2(\beta h/2) - \frac{8\sigma_1 k}{h^2} sin^2(\beta h/2) z^{-1}$$

Collect terms to form characteristic equation

$$0 = (1 + \sigma_0 k)z + (\frac{16\kappa^2 k^2}{h^4} sin^4(\beta h/2) + \frac{8\sigma_1 k}{h^2} sin^2(\beta h/2) - 2) + (1 - \sigma_0 k - \frac{8\sigma_1 k}{h^2} sin^2(\beta h/2))z^{-1}$$

Divide by the term containing σ_0 and multiply by z to get the equation as a second-order polynomial. Using $\mathcal{S} = sin^2(\beta h/2)$ for brevity.

$$0 = z^2 + \Big(\frac{\frac{16\kappa^2 k^2}{h^4}\mathcal{S}^2 + \frac{8\sigma_1 k}{h^2}\mathcal{S} - 2}{1 + \sigma_0 k}\Big)z + \frac{1 - \sigma_0 k - \frac{8\sigma_1 k}{h^2}\mathcal{S}}{1 + \sigma_0 k}$$

Substituting the coefficients into (eq. 3.26) lets us determine the stability condition.

$$\Big|\Big(\frac{\frac{16\kappa^2 k^2}{h^4}\mathcal{S}^2 + \frac{8\sigma_1 k}{h^2}\mathcal{S} - 2}{1 + \sigma_0 k}\Big)\Big| - 1 \leq \frac{1 - \sigma_0 k - \frac{8\sigma_1 k}{h^2}\mathcal{S}}{1 + \sigma_0 k} \leq 1$$

Multiplying by $(1 + \sigma_0 k)$

$$\Big|\frac{16\kappa^2 k^2}{h^4}\mathcal{S}^2 + \frac{8\sigma_1 k}{h^2}\mathcal{S} - 2\Big| - 1 + \sigma_0 k \leq 1 - \sigma_0 k - \frac{8\sigma_1 k}{h^2}\mathcal{S} \leq 1 + \sigma_0 k$$

Adding $1 + \sigma_0 k$

$$\Big|\frac{16\kappa^2 k^2}{h^4}\mathcal{S}^2 + \frac{8\sigma_1 k}{h^2}\mathcal{S} - 2\Big| \leq 2 - \frac{8\sigma_1 k}{h^2}\mathcal{S} \leq 2 + 2\sigma_0 k$$

The second condition will always hold as $\sigma_0, \sigma_1 \geq 0$. We can then rewrite the previous equation as

$$2 - \frac{8\sigma_1 k}{h^2}\mathcal{S} \leq \frac{16\kappa^2 k^2}{h^4}\mathcal{S}^2 + \frac{8\sigma_1 k}{h^2}\mathcal{S} - 2 \leq 2 - \frac{8\sigma_1 k}{h^2}\mathcal{S}$$

Adding $2 - \frac{8\sigma_1 k}{h^2}\mathcal{S}$

$$0 \leq \frac{16\kappa^2 k^2}{h^4}\mathcal{S}^2 \leq 4 - \frac{16\sigma_1 k}{h^2}\mathcal{S}$$

$\frac{16\kappa^2 k^2}{h^4}\mathcal{S}^2$ is non negative, so the first condition is satisfied.
Adding $\frac{16\sigma_1 k}{h^2}\mathcal{S}$

$$\frac{16\kappa^2 k^2}{h^4}\mathcal{S}^2 + \frac{16\sigma_1 k}{h^2}\mathcal{S} \leq 4$$

Dividing by 4

$$\frac{4\kappa^2 k^2}{h^4}\mathcal{S}^2 + \frac{4\sigma_1 k}{h^2}\mathcal{S} \leq 1$$

As the previous equation takes its maximum value when $\mathcal{S} = 1$ it can be rewritten as

$$\frac{4\kappa^2 k^2}{h^4} + \frac{4\sigma_1 k}{h^2} \leq 1$$

Multiply by h^4

$$4\kappa^2 k^2 + 4\sigma_1 k h^2 \leq h^4$$

Add left side to right to form quadratic equation in h^2

$$h^4 + 4\sigma_1 k h^2 4\kappa^2 k^2 \geq 0$$

Then using the quadratic formula, h can be shown to be bound by

$$h \geq \sqrt{\frac{-4\sigma_1 k + \sqrt{(-4\sigma_1 k)^2 16\kappa^2 k^2}}{2}}$$

Appendix B

Free boundary derivation

We are interested in the last 2 rows of the matrices for both D_{xx} and D_{xxxx} or in other words row number $N-1$ and N

For grid u These will have the form:

$$D_{xx}[N-1,:] = \frac{1}{h^2}[0,0,...,0,1,-2,1]$$
$$D_{xx}[N,:] = \frac{1}{h^2}[0,0,...,0,0,2,-2]$$
$$D_{xxxx}[N-1,:] = \frac{1}{h^4}[0,0,...,1,-4,5,-2]$$
$$D_{xxxx}[N,:] = \frac{1}{h^4}[0,0,...,0,2,-4,2]$$

for the range $\{0,1,...,N-3,N-2,N-1,N\}$ Each row containing zeros until the last set of elements

We then substitute this into the following equation

$$\mathbf{u}^{n+1} = \frac{1}{A}(\mathbf{B}\mathbf{u}^n + \mathbf{C}\mathbf{u}^{n-1})$$

where

$$A = (1+\sigma_0 k) \qquad \mathbf{B} = 2\mathbf{I} - \kappa^2 k^2 \mathbf{D}_{xxxx} + 2\sigma_1 k \mathbf{D}_{xx} \quad C = -(1-\sigma_0 k)\mathbf{I} - 2\sigma_1 k \mathbf{D}_{xx}$$

Given that the rows for D_{xx} and D_{xxxx} contain zeroes up until the final set of elements, we can simplify the update equation for u^n_{N-1} as

$$\begin{aligned} u^{n+1}_{N-1} = \Big(&2u^n_{N-1} - \frac{\kappa^2 k^2}{h^4}(u^n_{N-3} - 4u^n_{n-2} + 5u^n_{N-1} - 2u^n_N) \\ &+ \frac{2\sigma_1 k}{h^2}(u^n_{N-2} - 2u^n_{N-1} + u^n_N) - u^{n-1}_{N-1}(1-\sigma_0 k) \\ &- \frac{2\sigma_1 k}{h^2}(u^{n-1}_{N-2} - 2u^{n-1}_{N-1} + u^{n-1}_N)\Big)\frac{1}{1+\sigma_0 k} \end{aligned} \tag{B.1}$$

Collecting the terms and using $\mu = \frac{\kappa k}{h^2}$

$$\begin{aligned} u_{N-1}^{n+1} = \Big(& (2 - 5\mu^2 - \frac{4\sigma_1 k}{h^2}) u_{N-1}^n \\ & + (\frac{2\sigma_1 k}{h^2} + 2\mu^2) u_N^n \\ & + (4\mu^2 + \frac{2\sigma_1 k}{h^2}) u_{N-2}^n \\ & - \mu^2 u_{N-3}^n + (-1 + \sigma_0 k + \frac{4\sigma_1 k}{h^2}) u_{N-1}^{n-1} \\ & - \frac{2\sigma_1 k}{h^2} (u_{N-2}^{n-1} + u_N^{n-1}) \Big) \frac{1}{1 + \sigma_0 k} \end{aligned} \tag{B.2}$$

Following same methodology u_N^{n+1} can be derived as

$$\begin{aligned} u_N^{n+1} = \Big(& 2u_N^n - \frac{\kappa^2 k^2}{h^4} (2u_{N-2}^n - 4u_{N-1}^n + 2u_N^n) \\ & + \frac{2\sigma_1 k}{h^2} (2u_{N-1}^n - 2u_N^n) - (1 - \sigma_0 k) u_N^{n-1} \\ & - \frac{2\sigma_1 k}{h^2} (2u_{N-1}^{n-1} - 2u_N^{n-1}) \Big) \frac{1}{1 + \sigma_0 k} \end{aligned} \tag{B.3}$$

Collecting the terms and using $\mu = \frac{\kappa k}{h^2}$

$$\begin{aligned} u_N^{n+1} = \Big(& (2 - 2\mu^2 - \frac{4\sigma_1 k}{h^2}) u_N^n \\ & + (4\mu^2 + \frac{4\sigma_1 k}{h^2}) u_{N-1}^n \\ & - 2\mu^2 u_{N-2}^n \\ & + (-1 + \sigma_0 k + \frac{4\sigma_1 k}{h^2}) u_N^{n-1} \\ & - \frac{4\sigma_1 k}{h^2} u_{N-1}^{n-1} \Big) \frac{1}{1 + \sigma_0 k} \end{aligned} \tag{B.4}$$

Appendix C

Inner boundaries derivation

We are interested in the center 4 rows of the matrices for both D_{xx} and D_{xxxx} For grid u These will have the form:

$$D_{xx}[M_u - 1, :] = \frac{1}{h^2}[0, 0, ..., 1, -2, 1, 0, 0]$$
$$D_{xx}[M_u, :] = \frac{1}{h^2}[0, 0, ..., 0, 1, \mathcal{I} - 2, 1, -\mathcal{I}]$$

for the range $\{0, 1, ..., M_u - 2, M_u - 1, M_u, w_0, w_1\}$

$$D_{xxxx}[M_u - 1, :] = \frac{1}{h^4}[0, 0, ..., 1, -4, 6, \mathfrak{J}_1, 1, \mathfrak{J}_3, 0]$$
$$D_{xxxx}[M_u, :] = \frac{1}{h^4}[0, 0, ..., 0, 1, -4, \mathfrak{J}_0, \mathfrak{J}_1, \mathfrak{J}_2, \mathfrak{J}_3,]$$

for the range $\{0, 1, ..., M_u - 3, M_u - 2, M_u - 1, M_u, w_0, w_1, w_2\}$

with

$$\mathcal{I} = \frac{\alpha - 1}{\alpha + 1}$$

and

$$\mathfrak{J}_0 = \mathcal{I}^2 - 4\mathcal{I} + 6$$
$$\mathfrak{J}_1 = \mathcal{I} - 4$$
$$\mathfrak{J}_2 = -\mathcal{I}^2 + 4\mathcal{I} + 1$$
$$\mathfrak{J}_3 = -\mathcal{I}$$

We then substitute this into the following equation

$$\mathbf{u}^{n+1} = \frac{1}{A}(\mathbf{B}\mathbf{u}^n + \mathbf{C}\mathbf{u}^{n-1})$$

where

$$A = (1 + \sigma_0 k) \qquad \mathbf{B} = 2\mathbf{I} - \kappa^2 k^2 \mathbf{D}_{xxxx} + 2\sigma_1 k \mathbf{D}_{xx} \qquad C = -(1 - \sigma_0 k)\mathbf{I} - 2\sigma_1 k \mathbf{D}_{xx}$$

Given that the rows for D_{xx} and D_{xxxx} contain zeroes up until the final set of elements, we can simplify the update equation as

$$\begin{aligned} u^{n+1}_{M_u-1} = \Big(&2u^n_{M_u-1} - \frac{\kappa^2 k^2}{h^4}(u^n_{m_u-3} - 4u_{M_u-2} + 6u_{M_u-1} + \mathfrak{J}_1 u_{M_u} + w_0 + \mathfrak{J}_3 w_1) \\ &+ \frac{2\sigma_1 k}{h^2}(u^n_{M_u-2} - 2u^n_{M_u-1} + u^n_{M_u}) \\ &- u^{n-1}_{M_u-1}(1 - \sigma_0 k) \\ &- \frac{2\sigma_1 k}{h^2}(u^{n-1}_{M_u-2} - 2u^{n-1}_{M_u-1} + u^{n-1}_{M_u})\Big)\frac{1}{1 + \sigma_0 k} \end{aligned} \tag{C.1}$$

Collecting the terms and using $\mu = \frac{\kappa k}{h^2}$

$$\begin{aligned} u^{n+1}_{M_u-1} = \Big(&(2 - 6\mu^2 - \frac{4\sigma_1 k}{h^2})u^n_{M_u-1} \\ &+ (-\mathfrak{J}_1\mu^2 + \frac{2\sigma_1 k}{h^2})u^n_{M_u} \\ &+ (4\mu^2 + \frac{2\sigma_1 k}{h^2})u^n_{M_u-2} \\ &- \mu^2 u_{M_u-3} \\ &- \mu^2 w^n_0 \\ &- \mathfrak{J}_3\mu^2 w^n_1 \\ &+ (-1 + \sigma_0 k + \frac{4\sigma_1 k}{h^2})u^{n-1}_{M_u-1} \\ &- \frac{2\sigma_1 k}{h^2}(u^{n-1}_{M_u-2} + u^{n-1}_{M_u})\Big)\frac{1}{1 + \sigma_0 k} \end{aligned} \tag{C.2}$$

Following same methodology $u^{n+1}_{M_u}$ can be derived as

$$\begin{aligned} u^{n+1}_{M_u} = \Big(&2u^n_{M_u} - \frac{\kappa^2 k^2}{h^4}(u^n_{M_u-2} - 4u^n_{M_u-1} + \mathfrak{J}_0 u^n_{M_u} + \mathfrak{J}_1 w^n_0 + \mathfrak{J}_2 w^n_1 + \mathfrak{J}_3 w^n_2) \\ &+ \frac{2\sigma_1 k}{h^2}(u^n_{M_u-1} + (\mathcal{I} - 2)u^n_{M_u} + w^n_0 + -\mathcal{I} w^n_1) \\ &- (1 - \sigma_0 k)u^{n-1}_{M_u} \\ &- \frac{2\sigma_1 k}{h^2}(u^{n-1}_{M_u-1} + (\mathcal{I} - 2)u^{n-1}_{M_u} + w^{n-1}_0 + -\mathcal{I} w^{n-1}_1)\Big)\frac{1}{1 + \sigma_0 k} \end{aligned} \tag{C.3}$$

Collecting the terms and using $\mu = \frac{\kappa k}{h^2}$

$$
\begin{aligned}
u_{M_u}^{n+1} = \Big(& (2 - \mathfrak{J}_0\mu^2 + (\mathcal{I}-2)\frac{2\sigma_1 k}{h^2}) u_{M_u}^n \\
& + (4\mu^2 + \frac{2\sigma_1 k}{h^2}) u_{M_u-1}^n \\
& - \mu^2 u_{M_u-2}^n \\
& + (\frac{2\sigma_1 k}{h^2} - \mathfrak{J}_1\mu^2) w_0^n \\
& + (-\mathcal{I}\frac{2\sigma_1 k}{h^2} - \mathfrak{J}_2\mu^2) w_1^n \\
& - (\mathfrak{J}_3\mu^2) w_2^n \\
& + (-1 + \sigma_0 k - (\mathcal{I}-2)\frac{2\sigma_1 k}{h^2}) u_{M_u}^{n-1} \\
& - \frac{2\sigma_1 k}{h^2}(u_{M_u-1}^{n-1} + w_0^{n-1}) \\
& - (-\mathcal{I}\frac{2\sigma_1 k}{h^2}) w_1^{n-1} \Big) \frac{1}{1+\sigma_0 k}
\end{aligned}
\tag{C.4}
$$

The same approach can be done for the w grid which would result in the same update equations except mirrored spatially.