Dynamic Boundary Conditions

Silvin Willemsen

June 2020

1 Introduction

This document shows the work done and documentation on dynamic boundary conditions.

2 Motivation

Let's take the 1D wave equation in discrete time (see Figure 1a as an example):

$$\rho A \delta_{tt} u_l^n = T \delta_{xx} u_l^n, \tag{1}$$

parameterised using material density ρ, cross-sectional area A and tension T, and with simply supported boundary conditions such that

$$u_l^n = \delta_{xx} u_l^n = 0 \quad \text{at} \quad l = 0, N, \tag{2}$$

which states that both the state of the boundaries and the curvature at the boundaries should be 0. If we expand (2) at, say, the left boundary, we can introduce a virtual grid point u_{-1} that (as we will see later on) needs to be exactly $-u_1$ to satisfy this condition (see Figures 1b and 1c).

Through stability analysis one can arrive at a condition for the grid spacing that needs to be satisfied in order for the implementation to be stable. In this case this is

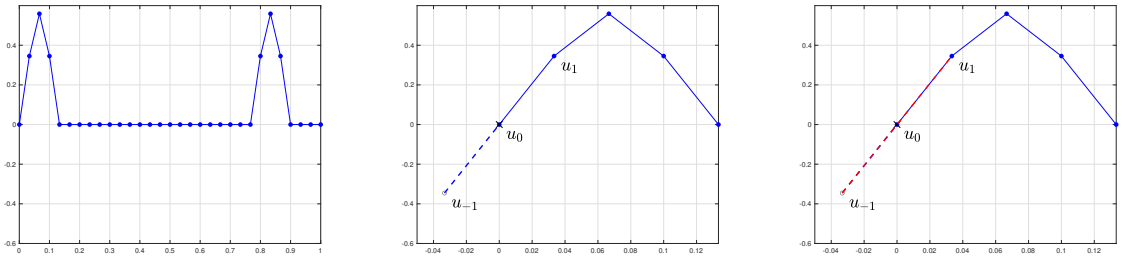
$$h \geq ck, \tag{3}$$

where grid spacing h is the distance between two neighbouring points, wave speed $c = \sqrt{T/\rho A}$ and time step $k = 1/f_\text{s}$ with sample rate f_s. The closer h is to this condition, the more accurate the scheme will be and the less bandwidth we lose. The reason we can't always satisfy condition (3) with equality (and consequently utilise the full bandwidth) is due to the fact that we require an integer number of grid points. In other words, we can't use "fractional grid points". Usually, the following steps are followed to calculate h [1, Section 6.2.10]:

$$N := \text{floor}(1/ck) \qquad h := 1/N. \tag{4}$$

There are cases where we can satisfy condition (3) with equality, i.e., when $1/ck$ is an integer ($1/ck = \text{floor}(1/ck)$). For example, when using a wave speed of $c = 1470$ m/s and sample rate $f_\text{s} = 44100$ Hz we can satisfy the stability condition with equality $h = 1/30$ (see Figure 1a). However, this is a special case, and if we want to change the wave speed dynamically we need to come up with something smarter. I would like to propose, *interpolated boundary conditions*, the possibility of which has briefly been mentioned by Stefan Bilbao in a footnote [1, p. 145], but never elaborated on[1].

[1] ...as "Footnotes are usually written by people who are pretending to know something but actually don't!" – Bilbao, 2020.



(a) String $N = 30$ (or 31?).

(b) At the left boundary.

(c) Curvature at the boundary (in this case at u_0) should be 0.

Figure 1: Case of grid point on boundary.

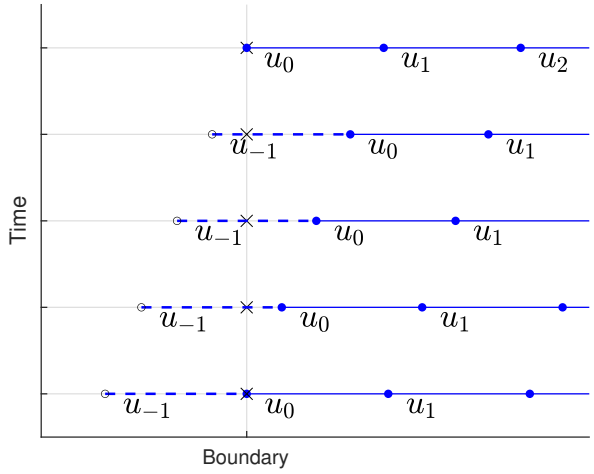


Figure 2: Grid changing over time.

2.1 Analogy with a real-life string

Imagine detuning a real-life string by turning the tuning knob, we essentially change the value for tension T in Eq. (1). If we imagine the tuning knob being at the nut (left boundary), more material will appear on that side when T decreases, and vice versa. To make the transition to the finite-difference setting easier, imagine equidistant points drawn on this string in such a way that there is a point exactly at the nut and the bridge (i.e. at each boundary). Then, when decreasing the tension, the point at the nut will start moving towards the bridge. As a matter of fact, all points (except for the one at the bridge) will start moving towards the bridge! Effectively, we slowly decrease the space between the points which – in a finite-difference setting – is analogous to decreasing the grid spacing h (see Figure 2 with time going from bottom to top). If we do this according to condition (3), we get the great side-effect of always satisfying this condition with equality, which means that we don't lose accuracy (or bandwidth). The issue left to solve now is to figure out what to do around the boundary. We continue by keeping in mind boundary condition (2).

3 The Interpolated Boundary

In this section, the location of a grid point u_l along the string will be given by x_l. In the normal case, i.e., when the first grid point lays on the boundary ($x_0 = 0$), we can satisfy (2) as follows:

$$\begin{aligned} & \delta_{xx} u_0 = 0 \\ & \frac{1}{h^2}(u_1 - 2u_0 + u_{-1}) = 0 \\ \xleftrightarrow{u_0=0} \quad & u_{-1} = -u_1. \end{aligned} \tag{5}$$

Applying this to an expanded version of Eq. (1) is unnecessary, as we know that $u_0 = 0$ at all times, and therefore does not need to be updated. If u_0 doesn't lay on the boundary, however, we need to define and satisfy the boundary condition differently. Let's change the boundary condition to something more general to apply to a point u_B which may or may not coincide with (or be equal to) u_0:

$$u_\mathrm{B} = \delta_{xx} u_\mathrm{B} = 0, \tag{6}$$

where the location of the boundary along the string $x_\mathrm{B} = 0$ at all times. Again, this condition states that the state as well as the curvature at the boundary needs to be 0.

As u_0 is not necessarily 0 (as it used to according to Eq. (2)) we need to calculate it using the scheme in (1). We expand the scheme at u_0 and solve for u_0^{n+1} as follows

$$u_0^{n+1} = 2u_0^n - u_0^{n-1} + \lambda^2(u_1^n - 2u_0^n + u_{-1}^n), \tag{7}$$

where $\lambda = ck/h$. Now, we need a definition for virtual grid point u_{-1}, which we can obtain using the new boundary condition (6).

In order to obtain a definition for the curvature at the boundary $\delta_{xx} u_\mathrm{B}$, we need two points that are equally distant from the boundary. Starting from the virtual grid point u_{-1} at the left side of the boundary, we can define some interpolated grid point u_I with the same distance from the boundary as u_{-1} (see Figure 3 for a visualisation of this).

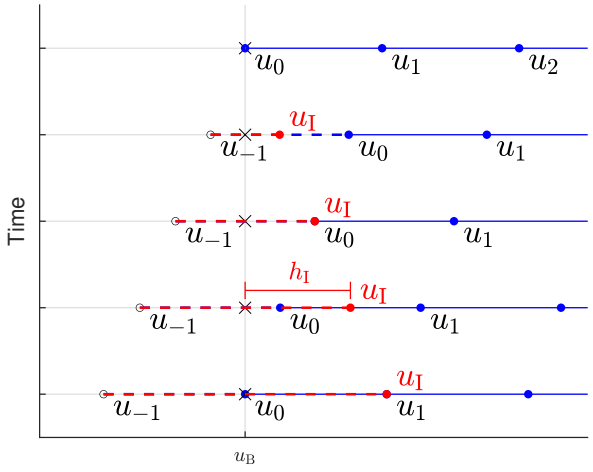


Figure 3: Boundary conditions at different values for h. Important to note is that the distance between u_0 and u_{-1} always remains (the current) h. The distance between u_{-1} and u_B (and also u_I and u_B) is defined as h_I. Then, from the location of u_{-1}, the location of u_I (the interpolated point used to calculate the value of u_{-1}) is determined.

This distance can be calculated using

$$h_\mathrm{I} = h - x_0, \tag{8}$$

where x_0 is the location of u_0 along the string length. As u_{-1} and u_I have the same distance h_I to the boundary and are at opposite sides, we can use these when expanding (6) and obtain a different definition for u_{-1}:

$$\begin{aligned} & \delta_{xx} u_\mathrm{B} = 0 \\ & \frac{1}{h_\mathrm{I}^2}(u_\mathrm{I} - 2u_\mathrm{B} + u_{-1}) = 0 \\ \overset{u_\mathrm{B}=0}{\Longleftrightarrow} \quad & u_{-1} = -u_\mathrm{I}, \end{aligned} \tag{9}$$

which is similar to condition (5), but now using the interpolated state (or grid point) u_I. The last thing we need is a definition for this state which is currently done using linear interpolation. We do, however, need different definitions in the cases where $x_\mathrm{I} > x_0$ (fourth instance from the top in Figure 3) and $x_\mathrm{I} \leq x_0$ (second and third instance from the top in Figure 3)

$$\begin{cases} u_\mathrm{I} = (1-\alpha)u_0 + \alpha u_1 \quad \text{where} \quad \alpha = \dfrac{h - 2x_0}{h} & x_\mathrm{I} > x_0 \quad (10a) \\ u_\mathrm{I} = \underbrace{(1-\alpha)u_\mathrm{B}}_{=0} + \alpha u_0 \quad \text{where} \quad \alpha = \dfrac{h_\mathrm{I}}{x_0} & x_\mathrm{I} \leq x_0 \quad (10b) \end{cases}$$

4 Energy

We can get the energy of Eq. (1) by first taking the inner product with respect to $\delta_{t\cdot} u$ like

$$\delta_{t+}\mathfrak{h} = \rho A \langle \delta_{t\cdot} u, \delta_{tt} u \rangle_\mathcal{D} - T \langle \delta_{t\cdot} u, \delta_{xx} u \rangle_\mathcal{D} = 0, \tag{11}$$

using integration by parts to get

$$\delta_{t+}\mathfrak{h} = \rho A \langle \delta_{t\cdot} u, \delta_{tt} u \rangle_\mathcal{D} + T \langle \delta_{t\cdot}\delta_{x+} u, \delta_{x+} u \rangle_{\underline{\mathcal{D}}} = \mathfrak{b}, \tag{12}$$

where boundary term

$$\mathfrak{b} = T(\delta_{t\cdot} u_N)(\delta_{x+} u_N) - T(\delta_{t\cdot} u_0)(\underbrace{\delta_{x+} u_{-1}}_{\delta_{x-} u_0}) \ . \tag{13}$$

Expanding Eq. (12) yields,

$$\delta_{t+}\mathfrak{h} = \rho A \sum_\mathcal{D} h(\delta_{t\cdot} u)(\delta_{tt} u) + T \sum_{\underline{\mathcal{D}}} h(\delta_{t\cdot}\delta_{x+} u)(\delta_{x+} u) \tag{14}$$

Then, using the following identities

$$(\delta_{t\cdot} u)(\delta_{tt} u) = \delta_{t+}\left(\frac{1}{2}(\delta_{t-} u)^2\right) \quad \text{and} \tag{15a}$$

$$(\delta_{t\cdot} u) u = \delta_{t+}\left(\frac{1}{2} u e_{t-} u\right) \tag{15b}$$

we can finally obtain the energy $\mathfrak{h}$:

$$\mathfrak{h} = \mathfrak{t} + \mathfrak{v} \quad \text{where} \tag{16}$$

$$\mathfrak{t} = \frac{\rho A}{2} \sum_\mathcal{D} h(\delta_{t-} u_l^n)^2 \quad \text{and} \quad \mathfrak{v} = \frac{T}{2} \sum_{\underline{\mathcal{D}}} h(\delta_{x+} u_l^n)(\delta_{x+} u_l^{n-1})$$

In the case when $x_{\mathrm{I}} \leq x_0$ (case (10b)) we can find an energy definition for the left boundary:

$$\begin{aligned}
\delta_{t+}\mathfrak{h}_{\mathrm{l}} &= T(\delta_{t\cdot}u_0)(\delta_{x-}u_0) \\
&= \frac{T}{h}(\delta_{t\cdot}u_0)(u_0 - u_{-1}) \\
\xLeftrightarrow{u_{-1}=-\alpha u_0} \quad &= \frac{T}{h}(\delta_{t\cdot}u_0)((1+\alpha)u_0) \\
&= \frac{T(1+\alpha)}{h}(\delta_{t\cdot}u_0)u_0 \\
\xLeftrightarrow{\text{Eq. (15b)}} \quad &= \frac{T(1+\alpha)}{h}\delta_{t+}\left(\frac{1}{2}u_0^n u_0^{n-1}\right) \\
\mathfrak{h}_l &= \frac{T(1+\alpha)}{2h}u_0^n u_0^{n-1}
\end{aligned} \tag{17}$$

The next step will be to find the energy for case (10a) (if it exists....). Here follow the first steps:

$$\begin{aligned}
\delta_{t+}\mathfrak{h}_{\mathrm{l}} &= T(\delta_{t\cdot}u_0)(\delta_{x-}u_0) \\
&= \frac{T}{h}(\delta_{t\cdot}u_0)(u_0 - u_{-1}) \\
\xLeftrightarrow{-u_{-1}=u_{\mathrm{I}} \text{ \& Eq. (10a)}} \quad &= \frac{T}{h}(\delta_{t\cdot}u_0)(u_0 + (1+\alpha)u_0 + \alpha u_1) \\
&= \frac{T(2+\alpha)}{h}(\delta_{t\cdot}u_0)u_0 + \frac{T}{h}(\delta_{t\cdot}u_0)(\alpha u_1) \\
\xLeftrightarrow{\text{Eq. (15b)}} \quad &= \frac{T(2+\alpha)}{h}\delta_{t+}\left(\frac{1}{2}u_0^n u_0^{n-1}\right) + \underbrace{\frac{T}{h}(\delta_{t\cdot}u_0)(\alpha u_1)}_{\text{The issue}}
\end{aligned} \tag{18}$$

References

[1] Stefan Bilbao. *Numerical Sound Synthesis*. John Wiley & Sons, 2009.

A Note regarding issue when increasing T

If we decrease T, the points smoothly enter the scheme without issues. However, in the opposite case when we increase T (moving from top to bottom in Figure 2), the string will already be moving at u_0 and continue moving due to its inertia (the $\rho A\delta_{tt}u$ term). In other words, even though we can satisfy the $\delta_{xx}u_0 = 0$ part of the boundary condition, $u_0 = 0$ is harder to satisfy. Something needs to be figured out for this..

B Alternative Approach: 1D-waves with connected free (Neumann) ends

Another approach to changing the grid dynamically is to add / remove points in the center of the string rather than the boundaries (see Figure 4). I expect that doing this, rather than adding / removing points at the boundary, will allow for smoother changes when increasing T, i.e, when the amount of points decreases (see Appendix A). Consider a string, u with $M_u = \mathrm{ceil}(0.5/ck)$ (simply M below for brevity) and w with $M_w = \mathrm{floor}(0.5/ck)$ points, i.e., half the number of points allowed by the stability

condition N, plus one for overlap due to the combined ceil floor operations (will elaborate below). Then the following boundary conditions are imposed:

$$\begin{aligned} u_0 = w_{M_w} = 0, \quad &\text{(Dirichlet)} \\ \delta_{x\cdot} u_M = \delta_{x\cdot} w_0 = 0 \quad &\text{(Neumann).} \end{aligned} \tag{19}$$

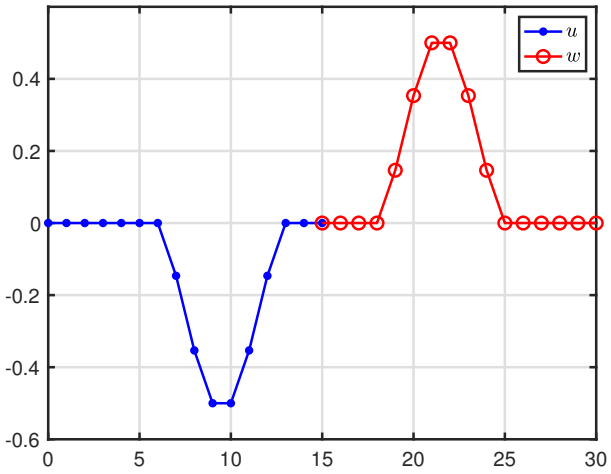


Figure 4: Two strings connected at one of their boundaries.

Then we can connect u_M and w_0 using a rigid connection, i.e.,

$$u_M^n = w_0^n \tag{20}$$

for all n. The system will now be

$$\begin{cases} \delta_{tt} u_l^n = c^2 \delta_{xx} u_l^n + J(x_{u_M}) F \\ \delta_{tt} w_l^n = c^2 \delta_{xx} w_l^n - J(x_{w_0}) F \end{cases} \tag{21}$$

where

$$J(x_i) = \begin{cases} \frac{1}{h}, & l = l_i \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

If we expand the spatial derivative operators in (21) at u_M and w_0 we get, recalling (19)

$$\begin{cases} \delta_{tt} u_M^n = \frac{c^2}{h^2}(2u_{M-1}^n - 2u_M^n) + \frac{1}{h} F \\ \delta_{tt} w_0^n = \frac{c^2}{h^2}(2w_1^n - 2w_0^n) - \frac{1}{h} F. \end{cases} \tag{23}$$

Because (20) is true we also know that $\delta_{tt} u_M^n = \delta_{tt} w_0^n$ for all n. We can then calculate F, by setting the equations in (21) equal to each other:

$$\frac{c^2}{h^2}(2u_{M-1}^n - 2u_M^n) + \frac{1}{h} F = \frac{c^2}{h^2}(2w_1^n - 2w_0^n) - \frac{1}{h} F \tag{24}$$

$$\frac{2}{h} F = \frac{c^2}{h^2}(2w_1^n - 2u_{M-1}^n)$$

$$F = h \frac{c^2}{h^2}(w_1^n - u_{M-1}^n) \tag{25}$$

Filling this into (23) after expansion of the second-time derivative yields

$$\begin{cases} u_M^{n+1} = 2u_M^n - u_n^{n-1} + \lambda^2(u_{M-1}^n - 2u_M^n + w_1^n) & \text{(26a)} \\ w_0^{n+1} = 2w_0^n - w_0^{n-1} + \lambda^2(u_{M-1}^n - 2w_0^n + w_1^n) & \text{(26b)} \end{cases}$$

which, (again, recalling (20)) are indeed equivalent expressions for the connected point. Here, w_1 in the first expression acts as virtual grid point u_{M+1}^n and u_{M-1}^n as virtual grid point w_{-1}^n. So essentially, to connect the two strings we add a state of one string to the update of the other.

B.1 Changing the grid

The previous is an exact solution to the problem when the stability condition is satisfied with equality, i.e, when $1/ck$ is an integer. If we then want to change the grid spacing according to $h = ck$ we leave the locations of the outer boundaries (u_0 and w_{M_w}) fixed and move the rest of the points towards their respective outer boundary (see Figure 5).

We now continue with the idea of adding a state of one string in the update of the other. When the stability condition is not satisfied with equality – and thus the points of the strings don't overlap – we use linear interpolator I_1. This results in,

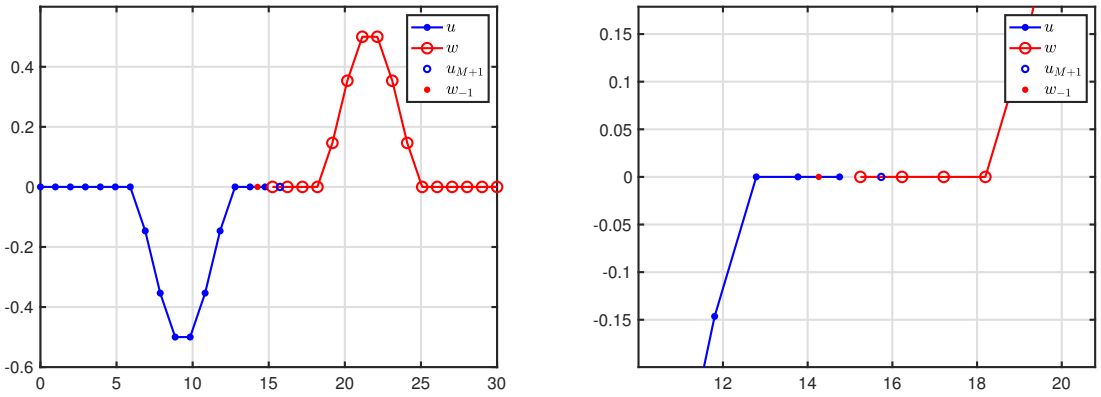
$$u_{M+1}^n = I_1(x_{u_{M+1}})w_l^n = (1-\alpha)w_1^n + \alpha w_0^n \quad (27)$$

$$w_{-1}^n = I_1(x_{w_{-1}})u_l^n = (1-\alpha)u_{M-1}^n + \alpha u_M^n \quad (28)$$

where

$$\alpha = \frac{x_{w_0} - x_{u_M}}{h}, \quad (29)$$

and grid-point locations $x_{u_{M+1}}$ and w_{-1}. Note that when h changes the connected points start to move away from each other. Then, when the boundary points surpass with the virtual points (i.e.



(a) When h changes.

(b) Zoomed

Figure 5

$x_{u_M} \leq x_{w_{-1}}$ and $x_{u_{M+1}} \leq x_{w_0}$) we alternate between adding / removing a point to the right side of u and the left side of w.

$$\begin{cases} \begin{cases} \mathbf{u}^n = [\mathbf{u}^n, \mathcal{I}(x_{u_{M+1}})\mathbf{v}^n]^T & \text{if } N^n \text{ is odd} \\ \mathbf{w}^n = [\mathcal{I}(x_{w_{-1}})\mathbf{v}_\star^n, \mathbf{w}^n]^T & \text{if } N^n \text{ is even} \end{cases} & \text{if } N^n > N^{n-1} \text{ (adding a point)} \\ \begin{cases} \mathbf{u}^n = [u_0^n, u_1^n ..., u_{M-1}^n]^T & \text{if } N^n \text{ is even} \\ \mathbf{w}^n = [w_1^n, w_2^n ..., w_{M_w}^n]^T & \text{if } N^n \text{ is odd} \end{cases} & \text{if } N^n < N^{n-1} \text{ (removing a point)} \end{cases} \quad (30)$$

where

$$\mathbf{v}^n = [u^n_{M-1}, u^n_M, w^n_0, w^n_1]^T \quad \text{and} \tag{31}$$

$$\mathbf{v}^n_\star = [w^n_1, w^n_0, u^n_M, u^n_{M-1}]^T \tag{32}$$

Figure 6 shows a point being added to the grid

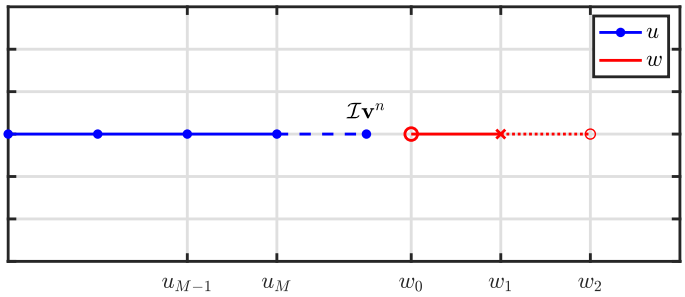


Figure 6: Point added to the grid according to (30) (at the boundary though..)

The interpolator used is (cubic Lagrange)

$$\mathcal{I} = \left[-\tfrac{\alpha(\alpha+1)}{(\alpha+2)(\alpha+3)} \quad \tfrac{2\alpha}{\alpha+2} \quad \tfrac{2}{\alpha+2} \quad -\tfrac{2\alpha}{(\alpha+3)(\alpha+2)}\right], \tag{33}$$

with

$$\alpha = \frac{x_{w_0} - (x_{u_M} + h)}{h}\,. \tag{34}$$

See section B.1.1 for a derivation of Eq. (33).

Even though subjective listening confirmed that the sound is smooth when adding / removing points using linear interpolation, the expected fundamental frequency ($f_0 \approx c/2$) was slightly too high when interpolation needed to happen. Instead of a linear, the cubic interpolator I_3 can be used according to

$$\begin{aligned} u^n_{M+1} = I_3(x_{u_{M+1}})w^n_l \quad &= \alpha_\mathrm{I} w^n_2 + \beta_\mathrm{I} w^n_1 + \gamma_\mathrm{I} w^n_0 + \delta_\mathrm{I} w^n_{-1} \\ w^n_{-1} = I_3(x_{w_{-1}})u^n_l \quad &= \alpha_\mathrm{I} u^n_{M-2} + \beta_\mathrm{I} u^n_{M-1} + \gamma_\mathrm{I} u^n_M + \delta_\mathrm{I} u^n_{M+1}. \end{aligned} \tag{35}$$

where

$$\alpha_\mathrm{I} = \frac{\alpha(\alpha-1)(\alpha-2)}{-6}, \quad \beta_\mathrm{I} = \frac{(\alpha-1)(\alpha+1)(\alpha-2)}{2},$$
$$\gamma_\mathrm{I} = \frac{\alpha(\alpha+1)(\alpha-2)}{-2}, \quad \text{and} \quad \delta_\mathrm{I} = \frac{\alpha(\alpha+1)(\alpha-1)}{6}\,.$$

We can solve for u^n_{M+1} and w^n_{-1} by treating (35) as a linear system of equations

$$\begin{bmatrix} u^n_{M+1} \\ w^n_{-1} \end{bmatrix} = \mathbf{A}^{-1}\mathbf{v}, \tag{36}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & -\delta_\mathrm{I} \\ -\delta_\mathrm{I} & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{v} = \begin{bmatrix} \alpha_\mathrm{I} w^n_2 + \beta_\mathrm{I} w^n_1 + \gamma_\mathrm{I} w^n_0 \\ \alpha_\mathrm{I} u^n_{M-2} + \beta_\mathrm{I} u^n_{M-1} + \gamma_\mathrm{I} u^n_M \end{bmatrix}. \tag{37}$$

Using cubic interpolation gives us the expected fundamental frequency for any wave speed. This can be proven using modal analysis, described in B.3.

B.1.1 Derivation of the LaGrange interpolator

This subsection shows a derivation of how to get to Eq. (33). We use Lagrange interpolation defined as

$$\mathcal{I}_i = \prod_{k=0,k\neq i}^{m} \frac{x - x_k}{x_i - x_k}. \tag{38}$$

We then need the locations of the points we are using to interpolate. If we let (normalised by h as all of the terms contain it)

$$\begin{aligned} x_0 &= x_{u_{M-1}} = 0, \quad \text{then} \\ x_1 &= x_{u_M} = 1, \\ x_2 &= x_{w_0} = 2 + \alpha, \quad \text{and} \\ x_3 &= x_{w_1} = 3 + \alpha. \end{aligned} \tag{39}$$

The x that we're solving for is

$$x = x_{u_M+1} = 2\ . \tag{40}$$

As all of these contain a multiplication with h, these can be divided out. The interpolator can then be built as follows:

$$\begin{aligned} \mathcal{I}_0 &= \left(\frac{2-1}{0-1}\right)\left(\frac{2-(\alpha+2)}{0-(\alpha+2)}\right)\left(\frac{2-(\alpha+3)}{0-(\alpha+3)}\right) = (-1)\left(\frac{-\alpha}{-(\alpha+2)}\right)\left(\frac{-(\alpha+1)}{-(\alpha+3)}\right) \\ &= -\frac{\alpha(\alpha+1)}{(\alpha+2)(\alpha+3)} && (41) \\ \mathcal{I}_1 &= \left(\frac{2-0}{1-0}\right)\left(\frac{2-(\alpha+2)}{1-(\alpha+2)}\right)\left(\frac{2-(\alpha+3)}{1-(\alpha+3)}\right) = (2)\left(\frac{-\alpha}{-(\alpha+1)}\right)\left(\frac{-(\alpha+1)}{-(\alpha+2)}\right) && (42) \\ &= \frac{2\alpha}{\alpha+2} && (43) \\ \mathcal{I}_2 &= \left(\frac{2-0}{(\alpha+2)-0}\right)\left(\frac{2-1}{(\alpha+2)-1}\right)\left(\frac{2-(\alpha+3)}{(\alpha+2)-(\alpha+3)}\right) = \left(\frac{2}{\alpha+2}\right)\left(\frac{1}{\alpha+1}\right)\left(\frac{-(\alpha+1)}{-1}\right) && (44) \\ &= \frac{2}{\alpha+2} && (45) \\ \mathcal{I}_3 &= \left(\frac{2-0}{(\alpha+3)-0}\right)\left(\frac{2-1}{(\alpha+3)-1}\right)\left(\frac{2-(\alpha+2)}{(\alpha+3)-(\alpha+2)}\right) = \left(\frac{2}{\alpha+3}\right)\left(\frac{1}{\alpha+2}\right)\left(\frac{-\alpha}{1}\right) && (46) \\ &= -\frac{2\alpha}{(\alpha+3)(\alpha+2)} && (47) \end{aligned}$$

B.2 Sinc interpolation

An even better result is obtained using *sinc interpolation*.

B.2.1 Introduction

One can retrieve a continuous-time signal $y(t)$ from a discrete time-series $y[n]$ using

$$y(t) = \sum_{n=-\infty}^{\infty} y[n] \cdot \operatorname{sinc}\left(\frac{t - nT}{T}\right). \tag{48}$$

where the normalised sinc function is defined as

$$\operatorname{sinc}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x} & x \neq 0 \\ 1 & x = 0. \end{cases} \tag{49}$$

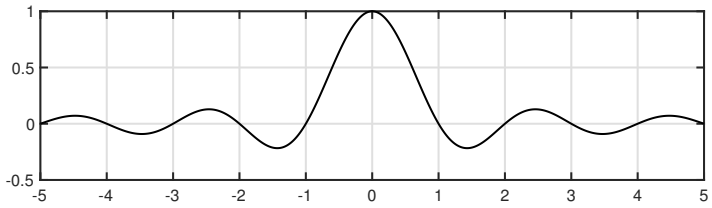


Figure 7: Sinc function in Eq. (49) truncated at $x = -5$ and $x = 5$.

Also see Figure 7.

Naturally, (48) can not be implemented "as is", due to the infinite sum. The continuous-time signal y can, however, be quite well approximated even with a truncated version of Eq. (48).

Going from DSP back to FDTD methods, (48) can be rewritten in terms we already know and applied to grid function u in space (rather than time)

$$u(x^\star) = \sum_{l=\mathcal{D}_\mathrm{i}} u_l \cdot \mathrm{sinc}\left(\frac{x^\star - lh}{h}\right). \tag{50}$$

where $\mathcal{D}_\mathrm{i}$ is the range for interpolation. Ideally, one uses the entire known domain of

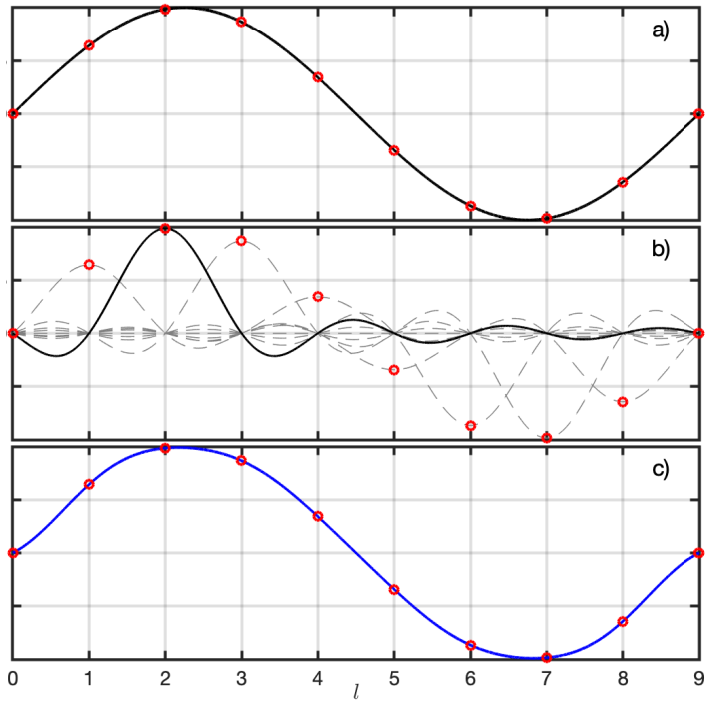


Figure 8: Explanation of sinc interpolation in Eq. (50). a) The "true" state of the system is sinusoidal and is sampled at $l \in [0, 9]$ marked in red. b) Sinc functions with their centers at the sampled points scaled by the value of u_l are drawn in grey with the one centered at $l = 2$ highlighted in black. Notice that exactly at a sample point l, all functions are 0 except for the one centered at l. c) The interpolated system is retrieved by adding all the functions in b) and plotted in blue. The interpolated state is almost identical to the true state shown in a).

B.3 Modal Analysis

In order to analyse the behaviour of the interpolator, one can perform a modal analysis. It is possible to write the complete system in matrix form as

$$\mathbf{U}^{n+1} = \mathbf{B}\mathbf{U}^n - \mathbf{U}^{n-1} \tag{51}$$

where

$$\mathbf{U}^n = \left[u_1^n, \ldots, u_M^n, w_0^n, \ldots, w_{M_w-1}^n\right]^T, \tag{52}$$

(so $\mathbf{u}^n$ and $\mathbf{w}^n$ concatenated excluding the outer boundaries). The modal frequencies (eigenfrequencies) of the system can then be calculated using the following formula [1, p. 174]

$$f_p = \frac{1}{2\pi k}\cos^{-1}\left(\frac{1}{2}\mathrm{eig}_p(\mathbf{B})\right). \tag{53}$$

If $\alpha = 0$, i.e., $x_{u_M} = x_{w_0}$ $\mathbf{B}$ looks like

$$\mathbf{B} = 2\mathbf{I} + \lambda^2 \left[\begin{array}{ccccc|ccccc} -2 & 1 & & & & & & & & \\ 1 & -2 & 1 & & & & & & & \\ & \ddots & \ddots & \ddots & & & & 0 & & \\ & & 1 & -2 & 1 & & & & & \\ & & & 1 & -2 & 0 & 1 & & & \\ \hline & & & 1 & 0 & -2 & 1 & & & \\ & & & & & 1 & -2 & 1 & & \\ & & 0 & & & & \ddots & \ddots & \ddots & \\ & & & & & & & 1 & -2 & 1 \\ & & & & & & & & 1 & -2 \end{array}\right] \tag{54}$$

As $\lambda = 1$ at all times we can write (54) as

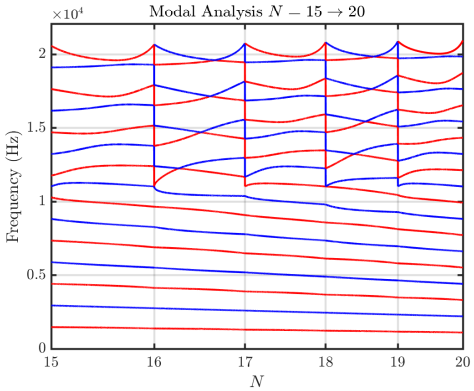
$$\mathbf{B} = \left[\begin{array}{ccccc|ccccc} 0 & 1 & & & & & & & & \\ 1 & 0 & 1 & & & & & & & \\ & \ddots & \ddots & \ddots & & & & 0 & & \\ & & 1 & 0 & 1 & & & & & \\ & & & 1 & 0 & 0 & 1 & & & \\ \hline & & & 1 & 0 & 0 & 1 & & & \\ & & & & & 1 & 0 & 1 & & \\ & & 0 & & & & \ddots & \ddots & \ddots & \\ & & & & & & & 1 & 0 & 1 \\ & & & & & & & & 1 & 0 \end{array}\right] \tag{55}$$

The 1's in the top-right and bottom-left quadrant show the interaction between $\mathbf{u}$ and $\mathbf{w}$ as shown in (26). In other words, u_M^n is calculated using w_1^n and w_0^n using u_{M-1}^n. The following subsections show the experiments done with the three different types of interpolation mentioned in the previous section, i.e., linear, cubic and sinc interpolation.

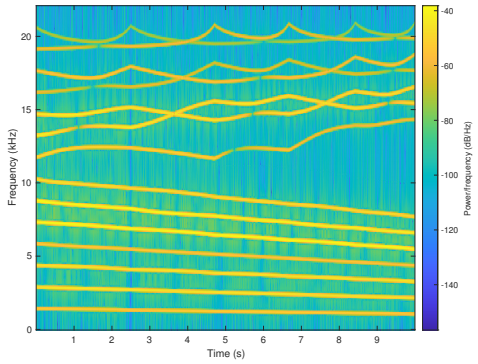
B.3.1 Linear interpolation

We can change $\mathbf{B}$ to include linear interpolation using (focusing only on the middle of the matrix)

$$\mathbf{B}_1 = \left[\begin{array}{ccc|ccc} \ddots & \ddots & & & & 0 \\ 1 & 0 & 1 & & & \\ & 1 & 0 & \alpha & (1-\alpha) & \\ \hline & (1-\alpha) & \alpha & 0 & 1 & \\ & & & 1 & 0 & 1 \\ 0 & & & & \ddots & \ddots \end{array}\right] \tag{56}$$



(a) Modal analysis for linear interpolation.



(b) Spectrogram of system wave excited with raised cosine when $N = 15$.

Figure 9: 1D wave going linearly from $c = 2940$ m/s ($N = 15$) to $c = 2205$ m/s ($N = 20$). Points are added in the middle.

B.3.2 Quadratic Interpolation

One could increase the number of points to take into account in the interpolation. Here we use u_M, w_0 and w_1 to calculate u_{M+1}. The matrix then becomes

$$\mathbf{B}_2 = \left[\begin{array}{ccc|ccc} \ddots & \ddots & & & & 0 \\ 1 & 0 & 1 & & & \\ & 1 & \frac{\alpha-1}{\alpha+1} & 1 & -\frac{\alpha-1}{\alpha+1} & \\ \hline & -\frac{\alpha-1}{\alpha+1} & 1 & \frac{\alpha-1}{\alpha+1} & 1 & \\ & & & 1 & 0 & 1 \\ 0 & & & & \ddots & \ddots \end{array}\right] \tag{57}$$

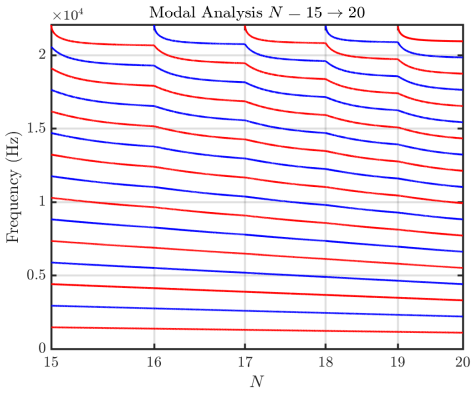
B.3.3 Cubic Interpolation

Equally spaced grid points

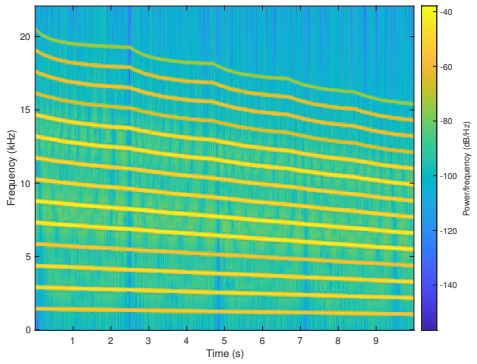
Cubic interpolation (using equally spaced points) is slightly trickier but still possible. Renaming $\mathbf{A}^{-1}$ from Eq. (37) to $\mathbf{A}^{\mathrm{i}}$ and writing out (36) yields

$$u_{M+1}^n = \mathbf{A}_{1,1}^{\mathrm{i}} \begin{bmatrix} \gamma_{\mathrm{I}} & \beta_{\mathrm{I}} & \alpha_{\mathrm{I}} \end{bmatrix} \begin{bmatrix} w_0^n & w_1^n & w_2^n \end{bmatrix}^T + \mathbf{A}_{1,2}^{\mathrm{i}} \begin{bmatrix} \alpha_{\mathrm{I}} & \beta_{\mathrm{I}} & \gamma_{\mathrm{I}} \end{bmatrix} \begin{bmatrix} u_{M-2}^n & u_{M-1}^n & u_M^n \end{bmatrix}^T \tag{58}$$

$$w_{-1}^n = \mathbf{A}_{2,1}^{\mathrm{i}} \begin{bmatrix} \gamma_{\mathrm{I}} & \beta_{\mathrm{I}} & \alpha_{\mathrm{I}} \end{bmatrix} \begin{bmatrix} w_0^n & w_1^n & w_2^n \end{bmatrix}^T + \mathbf{A}_{2,2}^{\mathrm{i}} \begin{bmatrix} \alpha_{\mathrm{I}} & \beta_{\mathrm{I}} & \gamma_{\mathrm{I}} \end{bmatrix} \begin{bmatrix} u_{M-2}^n & u_{M-1}^n & u_M^n \end{bmatrix}^T \tag{59}$$



(a) Modal analysis for linear interpolation.

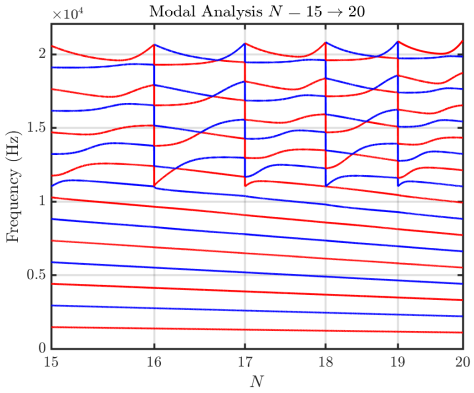


(b) Spectrogram of system wave excited with raised cosine when $N = 15$.

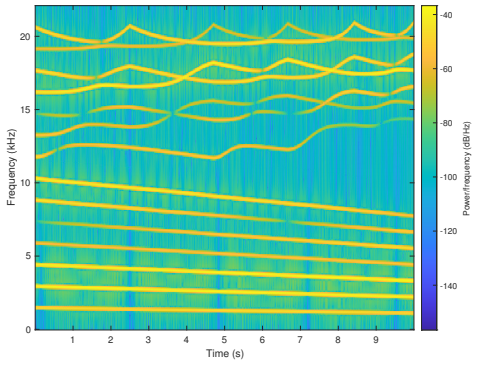
Figure 10: 1D wave going linearly from $c = 2940$ m/s ($N = 15$) to $c = 2205$ m/s ($N = 20$). Points are added in the middle.

Recalling Eq. (26a) where w_1^n acts as virtual grid point u_{M+1}^n and Eq. (26b) we can use this to insert the solution from (36) into $\mathbf{B}$ according to

$$\mathbf{B}_3 = \left[\begin{array}{ccc|ccc} \ddots & \ddots & & & & 0 \\ 1 & 0 & 1 & & & \\ \mathbf{A}^{\mathrm{i}}_{1,2}\alpha_{\mathrm{I}} & \mathbf{A}^{\mathrm{i}}_{1,2}\beta_{\mathrm{I}} + 1 & \mathbf{A}^{\mathrm{i}}_{1,2}\gamma_{\mathrm{I}} & \mathbf{A}^{\mathrm{i}}_{1,1}\gamma_{\mathrm{I}} & \mathbf{A}^{\mathrm{i}}_{1,1}\beta_{\mathrm{I}} & \mathbf{A}^{\mathrm{i}}_{1,1}\alpha_{\mathrm{I}} \\ \mathbf{A}^{\mathrm{i}}_{2,2}\alpha_{\mathrm{I}} & \mathbf{A}^{\mathrm{i}}_{2,2}\beta_{\mathrm{I}} & \mathbf{A}^{\mathrm{i}}_{2,2}\gamma_{\mathrm{I}} & \mathbf{A}^{\mathrm{i}}_{2,1}\gamma_{\mathrm{I}} & \mathbf{A}^{\mathrm{i}}_{2,1}\beta_{\mathrm{I}} + 1 & \mathbf{A}^{\mathrm{i}}_{2,1}\alpha_{\mathrm{I}} \\ & & & 1 & 0 & 1 \\ 0 & & & & \ddots & \ddots \end{array}\right] \tag{60}$$



(a) Modal analysis for cubic interpolation.



(b) Spectrogram.

Figure 11: Cubic Interpolation

Two points on each side

Alternatively, one can create a custom cubic interpolator for unequally spaced gridpoints using (38). If we want the number of points to the left and the right of u_{M+1} to be the same we need to consider u_M and w_0 and right of it are w_1 and w_2 (look at Figure 5b for reference). Similar to section B.1.1

we define the following

$$\begin{aligned} x_0 &= x_{u_M} = 0 \\ x_1 &= x_{w_0} = \alpha \\ x_2 &= x_{w_1} = \alpha + 1 \\ x_3 &= x_{w_2} = \alpha + 2, \end{aligned} \tag{61}$$

and the x that we're solving for is

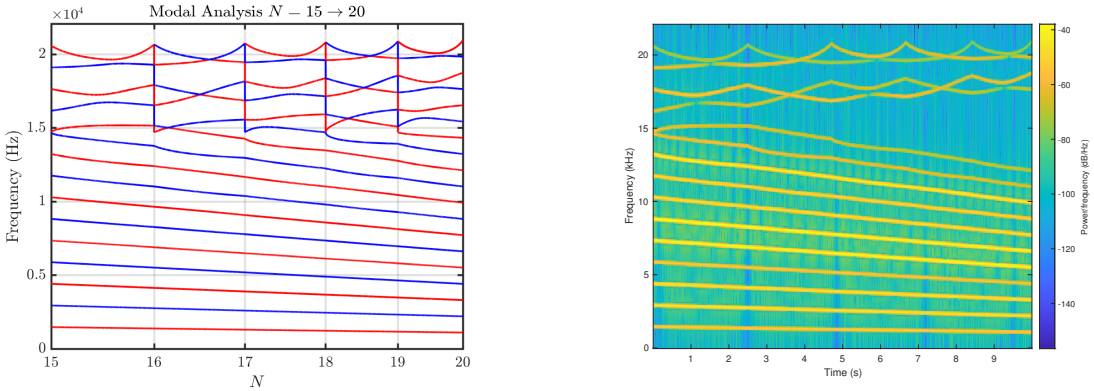
$$x = x_{u_M+1} = 1. \tag{62}$$

This results in the following interpolator:

$$\mathcal{I}_3^\star = \begin{bmatrix} \frac{\alpha-1}{\alpha+2} & \frac{\alpha+1}{2} & (1-\alpha) & \frac{\alpha(\alpha-1)}{2(\alpha+2)} \end{bmatrix} . \tag{63}$$

This results in the following $\mathbf{B}$ matrix:

$$\mathbf{B}_3^\star = \left[\begin{array}{ccc|ccc} \ddots & \ddots & & & & 0 \\ 1 & 0 & 1 & & & \\ & 1 & \frac{\alpha-1}{\alpha+2} & \frac{\alpha+1}{2} & (1-\alpha) & \frac{\alpha(\alpha-1)}{2(\alpha+2)} \\ \hline \frac{\alpha(\alpha-1)}{2(\alpha+2)} & (1-\alpha) & \frac{\alpha+1}{2} & \frac{\alpha-1}{\alpha+2} & 1 & \\ & & & 1 & 0 & 1 \\ 0 & & & & \ddots & \ddots \end{array}\right] \tag{64}$$



(a) Modal analysis for alternative cubic interpolation.

(b) Spectrogram.

Figure 12: Alternative Cubic interpolation.

Same points for both virtual grid points

As the results were still slightly unsatisfactory, a shifted version
where for both virtual grid points, u_{M-1}, u_M, w_0 and w_1 were used.

$$\mathbf{B}_3' = \left[\begin{array}{ccc|ccc} \ddots & \ddots & & & & 0 \\ 1 & 0 & 1 & & & \\ & 1 - \frac{\alpha(\alpha-1)}{(\alpha+1)(\alpha+2)} & \frac{2(\alpha-1)}{\alpha+1} & \frac{2}{\alpha+1} & -\frac{2(\alpha-1)}{(\alpha+1)(\alpha+2)} & \\ \hline & -\frac{2(\alpha-1)}{(\alpha+1)(\alpha+2)} & \frac{2}{\alpha+1} & \frac{2(\alpha-1)}{\alpha+1} & 1 - \frac{\alpha(\alpha-1)}{(\alpha+1)(\alpha+2)} & \\ & & & 1 & 0 & 1 \\ 0 & & & & \ddots & \ddots \end{array}\right] \tag{65}$$

B.3.4 Quartic interpolation

Although with the alternative cubic interpolation approach there are two points on either side of the point we want to calculate, and in that sense symmetric, the distance from that point is not. We can take one extra point and create a quartic interpolator:

$$\begin{aligned} x_0 &= x_{u_{M-1}} = 0 \\ x_1 &= x_{u_M} = 1 \\ x_2 &= x_{w_0} = \alpha + 1 \\ x_3 &= x_{w_1} = \alpha + 2 \\ x_4 &= x_{w_2} = \alpha + 3, \end{aligned} \tag{66}$$

and the x that we're solving for is

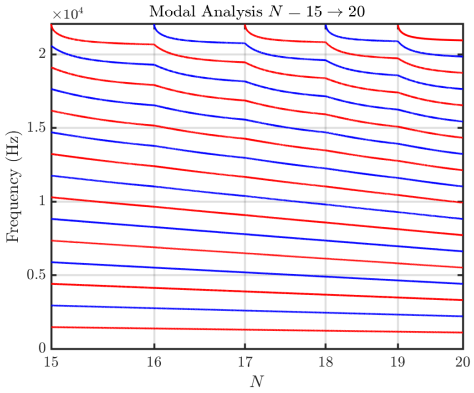
$$x = x_{u_M+1} = 2. \tag{67}$$

This results in the following interpolator:

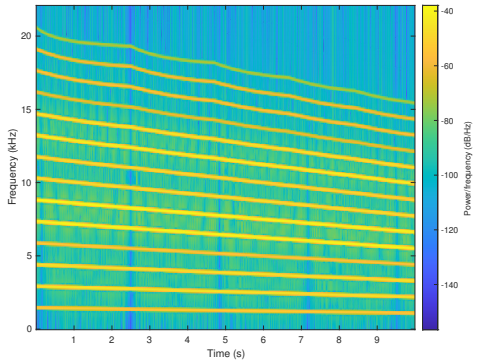
$$\mathcal{I}_4 = \begin{bmatrix} \frac{-\alpha(\alpha-1)}{(\alpha+2)(\alpha+3)} & \frac{2(\alpha-1)}{\alpha+2} & 1 & \frac{-2(\alpha-1)}{\alpha+2} & \frac{\alpha(\alpha-1)}{(\alpha+2)(\alpha+3)} \end{bmatrix}. \tag{68}$$

and the following matrix

$$\mathbf{B}_4 = \left[\begin{array}{ccc|ccc} \ddots & \ddots & & & & 0 \\ 1 & 0 & 1 & & & \\ & 1 - \frac{\alpha(\alpha-1)}{(\alpha+2)(\alpha+3)} & \frac{2(\alpha-1)}{\alpha+2} & 1 & -\frac{2(\alpha-1)}{\alpha+2} & \frac{\alpha(\alpha-1)}{(\alpha+2)(\alpha+3)} \\ \frac{\alpha(\alpha-1)}{(\alpha+2)(\alpha+3)} & -\frac{2(\alpha-1)}{\alpha+2} & 1 & \frac{2(\alpha-1)}{\alpha+2} & 1 - \frac{\alpha(\alpha-1)}{(\alpha+2)(\alpha+3)} & \\ & & & 1 & 0 & 1 \\ 0 & & & & \ddots & \ddots \end{array}\right] \tag{69}$$



(a) Modal analysis for quartic interpolation.



(b) Spectrogram of system wave excited with raised cosine when $N = 15$.

Figure 13: Quartic interpolation.

The results for a system going linearly from $c = 2940$ m/s ($N = 15$) to $c = 2205$ m/s ($N = 20$), can be found in Figure **??**. Figure **??** shows a number of interesting things

- At every integer N, there are $N - 1$ modes (related to the number of moving points at that point) that are integer multiples of the fundamental. This is expected and desired. However, due to the extra point used for overlap, an extra mode arises between the lower and the upper half of the harmonics. If the system is excited when the system has an integer N (and $\alpha = 0$ in Eq. (35)), this extra mode will not be excited due to the constraint of a rigid connection.

- Modes lower than and equal to half the number of total modes follow a linear pattern down, whereas higher modes follow a "squiggly" pattern upwards.
- Modal crossings can be seen when moving from an even to an odd N.

The spectrogram in Figure 14b shows that excited modes follow the "paths of least resistance" looking at Figure **??**. The latter two points are an undesired behaviour.

Notes about what the expected / desired behaviour should be. – or – System Requirements

- Smooth between different number-of-points-configurations
- Linearly down / up

The following section goes into how to get closer to a desired behaviour.

B.3.5 Deciding where to add / remove points

Until now we have only looked at adding / removing points at the center of the string as in Eq. (30), i.e, alternating between the left and the right half. To get decrease of some of the undesired behaviour described above, such as "squiggly" patterns for high modes and modal crossings. Trial and error showed that the closer the point add / remove location is to a boundary, the less this undesired behaviour shows up. The closest we can get to a boundary is to have u be $N-1$ points (including the boundary) and w two points, one moving, one boundary. If we then want to calculate the interpolated points using (37) we can extend the Dirichlet boundary condition in Eq. (19) to the simply supported one

$$w_{M_w} = \delta_{xx} w_{M_w} = 0, \quad \text{(Simply Supported)} \tag{70}$$

to find the definition for w_2^n

$$w_2^n = -w_0^n. \tag{71}$$

This condition can be exploited to have less points interacting with each other (which probably caused the undesired behaviour described above). As w_1 is the boundary and thus excluded from the calculation we can build matrix B_3'

$$\mathbf{B}_3' = 2\mathbf{I} + \lambda^2 \left[\begin{array}{ccc|c} \ddots & \ddots & & 0 \\ 1 & -2 & 1 & \\ \mathbf{A}^{\mathrm{i}}_{1,2}\alpha_{\mathrm{I}} & \mathbf{A}^{\mathrm{i}}_{1,2}\beta_{\mathrm{I}} + 1 & \mathbf{A}^{\mathrm{i}}_{1,2}\gamma_{\mathrm{I}} - 2 & \mathbf{A}^{\mathrm{i}}_{1,1}(\gamma_{\mathrm{I}} - \alpha_{\mathrm{I}}) \\ \hline \mathbf{A}^{\mathrm{i}}_{2,2}\alpha_{\mathrm{I}} & \mathbf{A}^{\mathrm{i}}_{2,2}\beta_{\mathrm{I}} & \mathbf{A}^{\mathrm{i}}_{2,2}\gamma_{\mathrm{I}} & \mathbf{A}^{\mathrm{i}}_{2,1}(\gamma_{\mathrm{I}} - \alpha_{\mathrm{I}}) - 2 \end{array}\right] \tag{72}$$

Analysing this using Eq. (53) yields Figure 14a.

B.4 Stiff string

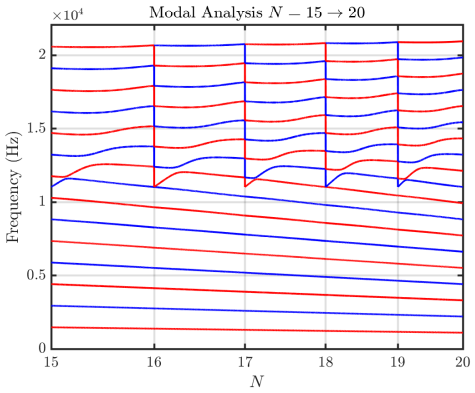
To prove that this technique also works with more complicated schemes, we can apply the above to the stiff string with the following scheme

$$\delta_{tt} u = c^2 \delta_{xx} u - \kappa^2 \delta_{xxxx} u \tag{73}$$

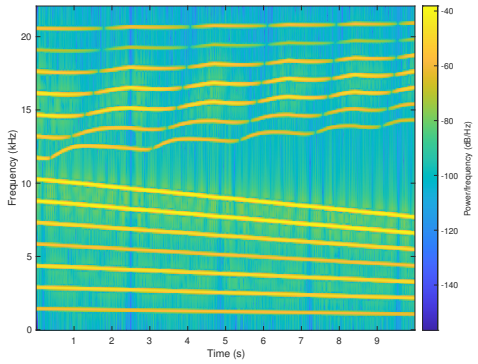
where $\kappa = \sqrt{EI/\rho A}$ with Young's modulus E and moment of inertia I. The difference with the 1D-wave equation is the stiffness term containing a fourth order spatial derivative. This requires an extra virtual grid point to be calculated at the boundaries. For the outer boundaries we choose simply supported conditions according to

$$u_0 = w_{M_w} = \delta_{xx} u_0 = \delta_{xx} w_{M_w} = 0. \tag{74}$$

The points needed at the other boundaries are calculated below.



(a) Modal analysis when points are added close at the boundary



(b) Spectrogram of system wave excited with raised cosine when $N = 15$.

Figure 14: 1D wave going linearly from $c = 2940$ m/s ($N = 15$) to $c = 2205$ m/s ($N = 20$). Points are added at the boundary.

We start by (again) connecting u_M^n and w_0^n using a rigid connection using (20) yielding the following system:

$$\begin{cases} \delta_{tt} u_l^n = c^2 \delta_{xx} u_l^n - \kappa^2 \delta_{xxxx} u_l^n + J(x_{u_M}) F \\ \delta_{tt} w_l^n = c^2 \delta_{xx} w_l^n - \kappa^2 \delta_{xxxx} w_l^n - J(x_{w_0}) F. \end{cases} \tag{75}$$

If we expand the spatial derivative operators and using the concept of weight ratio due to the overlap (see Appendix C) we get

$$\begin{cases} \delta_{tt} u_M^n = \frac{c^2}{h^2}(2u_{M-1}^n - u_M^n) - \frac{\kappa^2}{h^4}(2u_{M-2}^n - 8u_{M-1}^2 + 6u_M^n) + \frac{1}{h} F \\ \delta_{tt} w_0^n = \frac{c^2}{h^2}(2w_1^n - w_0^n) - \frac{\kappa^2}{h^4}(2w_2^n - 8w_1^2 + 6w_0^n) - \frac{1}{h} F. \end{cases} \tag{76}$$

Again, because (20) is true, $\delta_{tt} u_M = \delta_{tt} w_0$ and by setting the equations in (79) equal to each other we can solve for F

$$\begin{aligned} \frac{c^2}{h^2}(2u_{M-1}^n - 2u_M^n) - \frac{\kappa^2}{h^4}(2u_{M-2}^n - 8u_{M-1}^n + 6u_M^n) + \frac{1}{h} F = \\ \frac{c^2}{h^2}(2w_1^n - 2w_0^n) - \frac{\kappa^2}{h^4}(2w_2^n - 8w_1^n + 6w_0^n) - \frac{1}{h} F \end{aligned}$$

$$\frac{2}{h} F = \frac{c^2}{h^2}(2w_1^n - 2w_0^n - 2u_{M-1}^n + 2u_M^n) - \frac{\kappa^2}{h^4}(2w_2^n - 8w_1^n + 6w_0^n - 2u_{M-2}^n + 8u_{M-1}^n - 6u_M^n)$$

$$F = h\frac{c^2}{h^2}(w_1^n - u_{M-1}^n) - h\frac{\kappa^2}{h^4}(w_2^n - 4w_1^n + 4u_{M-1}^n - u_{M-2}^n). \tag{77}$$

If we add this to the respective equations in (79) and expand the acceleration term to get the following updates

$$\begin{cases} u_M^{n+1} = 2u_M^n - u_M^{n-1} + \lambda^2(u_{M-1}^n - 2u_M^n + w_1^n) - \mu^2(u_{M-2}^n - 4u_{M-1}^2 + 6u_M^n - 4w_1^n + w_2^n) \\ w_0^{n+1} = 2w_0^n - w_0^{n-1} + \lambda^2(u_{M-1}^n - 2w_0^n + w_1^n) - \mu^2(u_{M-2}^n - 4u_{M-1}^2 + 6w_0^n - 4w_1^2 + w_2^n), \end{cases} \tag{78}$$

which are (again) equivalent expressions for the connected point.

Along the same lines[2] we can find updates for u_{M-1}^n and w_1^n, which – through the fourth-order spatial derivative – are now also affected by the state of the other string:

$$\begin{cases} u_{M-1}^{n+1} = 2u_{M-1}^n - u_{M-1}^{n-1} + \lambda^2(u_{M-2}^n - 2u_{M-1}^n + u_M^n) - \mu^2(u_{M-3}^n - 4u_{M-2}^n + 6u_{M-1}^n - 4u_M^n + w_1^n) \\ w_1^{n+1} = 2w_1^n - w_1^{n-1} + \lambda^2(w_0^n - 2w_1^n + w_2^n) - \mu^2(u_{M-1}^n - 4w_0^n + 6w_1^n - 4w_2^n + w_3^n). \end{cases} \tag{79}$$

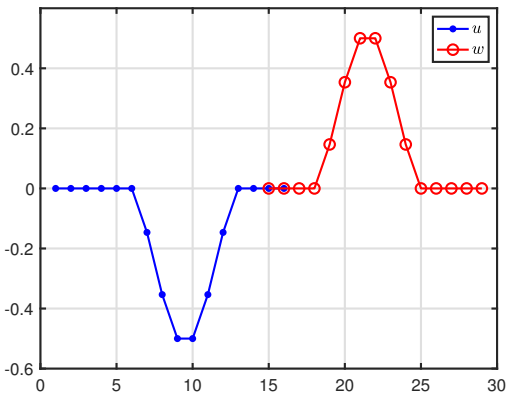
[2] might have to prove something here

C Adding an overlap

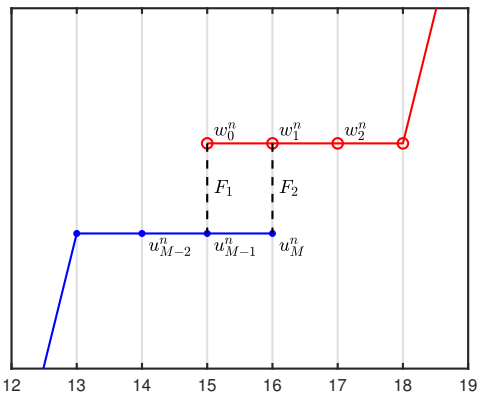
Another, slightly trickier, setup could be that the left string has an extra point at its right side to cause some overlap. This might come in handy later if we want to interpolate the forces between the strings for better quality. We can then apply two rigid connections to the overlapping points

$$\begin{aligned} u_{M-1}^n &= w_0^n, \\ u_M^n &= w_1^n. \end{aligned} \tag{80}$$

See Figure 15 for clarification.



(a) Overlapped points.



(b) Zoomed. Note that the overlapping points should always have the same state, but they have been pulled apart for clarity.

Figure 15

As an overlap happens, the string is "heavier" at this point if we keep the same physical parameters. However, in reality, the string is equally heavy everywhere. As shown in Figure 15b, the overlapping points (u_{M-1}^n, u_M^n, w_0^n and w_1^n) have to be made twice as light as their non-overlapping counterparts. From their perspective, u_{M-2}^n and w_2^n are twice as heavy when compared to themselves. Imagine changing the material density of these to double of what they usually are. This will change the wave speed according to:

$$c^2 = T/\rho A \quad \xRightarrow{2\rho} \quad c^2 = T/2\rho A \quad \Rightarrow \quad 2c^2 = T/\rho A. \tag{81}$$

For the overlapping points, we will now have the following update equations

$$\begin{cases} u_{M-1}^{n+1} = 2u_{M-1}^n - u_{M-1}^{n-1} + \lambda^2(2u_{M-2}^n - 2u_{M-1}^n + u_M^n) + \frac{k^2}{h}F_1 \\ u_M^{n+1} = 2u_M^n - u_M^{n-1} + \lambda^2(u_{M-1}^n - 2u_M^n) + \frac{k^2}{h}F_2 \\ w_0^{n+1} = 2w_0^n - w_0^{n-1} + \lambda^2(w_1^n - 2w_0^n) - \frac{k^2}{h}F_1 \\ w_1^{n+1} = 2w_1^n - w_1^{n-1} + \lambda^2(w_0^n - 2w_1^n + 2w_2^n) - \frac{k^2}{h}F_2 \end{cases} \tag{82}$$

One would think that some scaling also needs to happen for u_{M-1}^n in the update equation for u_{M-2}^{n+1} and w_1^n in w_2^n, but as we will see shortly, this is not necessary. This concept of a weight ratio also automatically happened in the previous case because of the Neumann boundary condition.

Recalling (80), we can then solve for F_1 and F_2 in the same way as before.

$$2u_{M-1}^n - u_{M-1}^{n-1} + \lambda^2(2u_{M-2}^n - 2u_{M-1}^n + u_M^n) + \frac{k^2}{h}F_1 = 2w_0^n - w_0^{n-1} + \lambda^2(w_1^n - 2w_0^n) - \frac{k^2}{h}F_1$$
$$\frac{2k^2}{h}F_1 = \lambda^2(-2u_{M-2}^n)$$
$$F_1 = -h\frac{c^2}{h^2}(u_{M-2}^n) \tag{83}$$

and

$$2u_M^n - u_M^{n-1} + \lambda^2(u_{M-1}^n - 2u_M^n) + \frac{k^2}{h}F_2 = 2w_1^n - w_1^{n-1} + \lambda^2(w_0^n - 2w_1^n + 2w_2^n) - \frac{k^2}{h}F_2$$
$$\frac{2k^2}{h}F_2 = \lambda^2(2w_2^n)$$
$$F_2 = h\frac{c^2}{h^2}(w_2^n). \tag{84}$$

When filled into the update equations in (85) we get

$$\begin{cases} u_{M-1}^{n+1} = 2u_{M-1}^n - u_{M-1}^{n-1} + \lambda^2(u_{M-2}^n - 2u_{M-1}^n + u_M^n) \\ u_M^{n+1} = 2u_M^n - u_M^{n-1} + \lambda^2(u_{M-1}^n - 2u_M^n + w_2^n) \\ w_0^{n+1} = 2w_0^n - w_0^{n-1} + \lambda^2(w_1^n - 2w_0^n + u_{M-2}^n) \\ w_1^{n+1} = 2w_1^n - w_1^{n-1} + \lambda^2(w_0^n - 2w_1^n + w_2^n). \end{cases} \tag{85}$$

Here, similarly to (26), w_2^n and u_{M-2}^n in the updates for u_M^{n+1} and w_0^{n+1} respectively act as virtual grid points. Now we can also see why as scaling for overlapping points u_{N-1}^n and w_1^n in the updates for u_{M-2}^{n+1} and w_2^{n+1} are not necessary.

C.0.1 Preliminary tests..

...show instability...

C.1 Energy

For the kinetic energy, we now have to add a scaling with 0.5, not only to the last point, but also to the rest of the overlapping ones:

$$\mathfrak{t}_u = \sum_{l=0}^{M-2} h(\delta_{t-}u_l^n)^2 + \frac{h}{2}(\delta_{t-}u_{M-1}^n)^2 + \frac{h}{2}(\delta_{t-}u_M^n)^2 \tag{86}$$

$$\mathfrak{t}_w = \sum_{l=2}^{M} h(\delta_{t-}w_l^n)^2 + \frac{h}{2}(\delta_{t-}w_0^n)^2 + \frac{h}{2}(\delta_{t-}w_1^n)^2 \tag{87}$$

$$\mathfrak{v}_u = \frac{c^2}{2}\sum_{l=0}^{M-2} h(\delta_{x+}u_l^n)(\delta_{x+}u_l^{n-1}) + c^2\frac{h}{4}(\delta_{x+}u_{M-1}^n)(\delta_{x+}u_{M-1}^{n-1}) \tag{88}$$

$$\mathfrak{v}_w = \frac{c^2}{2}\sum_{l=1}^{M-1} h(\delta_{x+}w_l^n)(\delta_{x+}w_l^{n-1}) + c^2\frac{h}{4}(\delta_{x+}w_0^n)(\delta_{x+}w_0^{n-1}) \tag{89}$$

D Dirichlet Condition (Stefan)

Consider a domain of length L, and grid spacing $h = ck$ (so, right at the Courant limit). Set $N = floor(L/h)$. Set the grid locations to be

$$x_0 = 0, \quad x_1 = h, \quad \ldots, \quad x_{N-1} = (N-1)h, \quad x_N = L = (N+\alpha)h \tag{90}$$

where $0 \leq \alpha \leq 1$. The final grid point lies directly on the boundary.

Under Dirichlet conditions, we set $u_0 = 0$ and $u_N = 0$. The Laplacian takes the form[3]

$$\delta_{xx} u_1 = \frac{1}{h^2}(u_2 - 2u_1) \tag{91}$$

$$\delta_{xx} u_l = \frac{1}{h^2}(u_{l+1} - 2u_l + u_{l-1}) \qquad 2 \leq l \leq N-2 \tag{92}$$

$$\delta_{xx} u_{N-1} = \frac{1}{h^2}\left(\frac{2}{\alpha+2}u_{N-2} - \frac{2}{\alpha+1}u_{N-1}\right) \tag{93}$$

and thus, in matrix form,

$$\mathbf{D}_{xx} = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & \frac{2}{\alpha+2} & -\frac{2}{\alpha+1} \end{bmatrix} \tag{94}$$

The scheme, as a whole, then looks like

$$\delta_{tt}\mathbf{u}^n = c^2 \mathbf{D}_{xx}\mathbf{u}^n \tag{95}$$

Define the matrix $\mathbf{P}$ as

$$\mathbf{P} = \text{diag}\left(1, 1, \ldots, \frac{\alpha+2}{2}\right) \tag{96}$$

Then $\mathbf{P}\mathbf{D}_{xx}$ is now symmetric.

Left-multiplying the scheme by $\mathbf{P}$ and then taking the inner product with $\delta_t.\mathbf{u}$ then gives a conserved energy. Notice that this reduces to the usual case when $\alpha = 0$.

D.1 Taylor series expansion (Silvin)

To get to the coefficients for u_{N-2} and u_{N-1} in (93) we use a Taylor series expansion around the grid point $N-1$. To start, it is easiest to assume an expansion around 0 and then apply it to a different location.

Suppose we have three grid locations $x_{-1} = -h, x_0 = 0$ and $x_1 = h(\alpha+1)$ and their function values u_{-1}, u_0 and u_1. We then want to find an approximation $L = \frac{\partial^2 u}{\partial x^2} + O(h)$ (where $O(h)$ describes the order of the error) of the form

$$L = a_{-1}u_{-1} + a_0 u_0 + a_1 u_1 \tag{97}$$

where the coefficients a_{-1}, a_0 and a_1 depend on α. I suppose in the regular case ($\alpha = 0$) the coefficients end up being $a_{-1} = 1/h^2$, $a_0 = -2/h^2$ and $a_1 = 1/h^2$. To find these coefficients we perform a Taylor series expansion. First we need to assume for a continuous function $u(x)$ that $u_{-1} = u(x_{-1}), u_0 = u(x_0)$ and $u_1 = u(x_1)$. We can then approximate this function u around location $x_0 (= 0)$ as

$$u(x) = u(0) + \frac{du(0)}{dx}x + \frac{1}{2}\frac{d^2u(0)}{dx^2}x^2 + O(x^3). \tag{98}$$

Here, $O(x^3)$ describes the order of the error (or the collective error term with order of x^3).

[3] $\frac{2}{\alpha+1} = \frac{2}{\alpha+2} + \frac{2}{(\alpha+2)(\alpha+1)}$

We can then fill in the values of x_{-1} and x_1 above ($u(x_0) = u(0)$) to get

$$\begin{aligned} u(-h) &= u(0) + \frac{du(0)}{dx}(-h) + \frac{1}{2}\frac{d^2u(0)}{dx^2}(-h)^2 + O(h^3) \\ &= u(0) - h\frac{du(0)}{dx} + \frac{h^2}{2}\frac{d^2u(0)}{dx^2} + O(h^3), \end{aligned} \tag{99}$$

and

$$\begin{aligned} u(h(\alpha+1)) &= u(0) + \frac{du(0)}{dx}(h(\alpha+1)) + \frac{1}{2}\frac{d^2u(0)}{dx^2}(h(\alpha+1))^2 + O(h^3) \\ &= u(0) + h(\alpha+1)\frac{du(0)}{dx} + \frac{h^2(\alpha+1)^2}{2}\frac{d^2u(0)}{dx^2} + O(h^3). \end{aligned} \tag{100}$$

The goal is then to solve for $\frac{d^2u(0)}{dx^2}$ as this is what we want to approximate in (1). We want to get rid of the first-order derivative term so we can perform the following addition

$$\begin{aligned} u(-h) + \frac{u(h(\alpha+1))}{\alpha+1} &= \left(1 + \frac{1}{\alpha+1}\right)u(0) + \left(-h + \frac{h(\alpha+1)}{\alpha+1}\right)\frac{du(0)}{dx} + \left(\frac{h^2}{2} + \frac{h^2(\alpha+1)^2}{2(\alpha+1)}\right)\frac{d^2u(0)}{dx^2} + O(h^3) \\ \frac{(\alpha+1)u(-h) + u(h(\alpha+1))}{\alpha+1} &= \left(\frac{\alpha+2}{\alpha+1}\right)u(0) + \left(\frac{h^2 + h^2(\alpha+1)}{2}\right)\frac{d^2u(0)}{dx^2} + O(h^3) \\ \frac{(\alpha+1)u(-h) - (\alpha+2)u(0) + u(h(\alpha+1))}{\alpha+1} &= \left(\frac{h^2(\alpha+2)}{2}\right)\frac{d^2u(0)}{dx^2} + O(h^3) \\ L = \frac{d^2u(0)}{dx^2} + O(h) &= \frac{2((\alpha+1)u(-h) - (\alpha+2)u(0) + u(h(\alpha+1)))}{h^2(\alpha+2)(\alpha+1)}. \end{aligned} \tag{101}$$

Note that due to a division with h^2 the order of the error is $O(h)$ rather than $O(h^3)$. The coefficients in (97) are then as follows:

$$a_{-1} = \frac{2}{h^2(\alpha+2)}, \quad a_0 = -\frac{2}{h^2(\alpha+1)}, \quad \text{and} \quad a_1 = \frac{2}{h^2(\alpha+2)(\alpha+1)}. \tag{102}$$

E Quick note on connections

When connecting elements (for now with a spring) we need to use interpolation and spreading operators. We can add a connection to two elements by localising the along the strings using a spreading operator $J(x_c)$:

$$\delta_{tt}u = c^2\delta_{xx}u + J(x_{u,c})F \tag{103}$$

$$\delta_{tt}w = c^2\delta_{xx}u - J(x_{w,c})F, \tag{104}$$

where $x_{u,c}$ and $x_{w,c}$ are the locations of connection along string u and w respectively. Note that the forces are equal and opposite when applied to their respective components.

Call the relative distance between the states at the connection location $\eta = I(x_{u,c})u - I(x_{w,c})w$. Using (103), we can then arrive at a definition for $\delta_{tt}\eta^{n+1}$

$$\delta_{tt}\eta = c^2 I(x_{u,c})\delta_{xx}u + I(x_{u,c})J(x_{u,c})F - c^2 I(x_{w,c})\delta_{xx} + I(x_{w,c})J(x_{w,c})F \tag{105}$$

We can connect the two elements using a rigid connection, i.e. $\eta = 0$. From this we can calculate the force directly:

$$F = \frac{c^2(I(x_{w,c})\delta_{xx}w - I(x_{u,c})\delta_{xx}u)}{I(x_{u,c})J(x_{u,c}) + I(x_{w,c})J(x_{w,c})}. \tag{106}$$

F Interpolants (Stefan)

OK, suppose you have a set of distinct grid locations, $x_l, l = 1, \ldots, N$. We don't assume they are equally spaced, as you will be working at the interface between two grids. Suppose also that x^* is the point at which you wish to find an interpolated value. Normally, x^* is enclosed by the set of points x_l, but it doesn't have to be (in which case you're doing extrapolation). Also assume dependent values y_l, $l = 1, \ldots, N$. Now, define a general interpolant, at a coordinate x^* as

$$y^* = \sum_{l=1}^{n} a_l y_l \tag{107}$$

Now, assume that the original samples are drawn from a single complex exponential of the form

$$y_l = e^{j\beta x_l} \tag{108}$$

for some wavenumber β. WLOG, assume $x^* = 0$. You can then define an error measure as

$$E(x_1, \ldots, x_N) = \int_{-\beta_{max}}^{\beta_{max}} |1 - y^*|^2 d\beta \tag{109}$$

where here, β_{max} is a maximal wavenumber; if the grid points are equally spaced, with spacing h, you might choose $\beta_{max} = \pi/h$, but you don't have to! It could be smaller than this, allowing for optimisation over a smaller range of wavenumbers.

Then...you need to minimise this error with respect to all the parameters a_l. If the points are equally spaced, you end up with a sinc interpolant. If not, well, you end up with something else.