

# Connected Physical Models controlled with the Sensel Morph

Silvin Willemsen, Nikolaj Andersson and Stefania Serafin

Multisensory Experience Lab, CREATE, Aalborg University Copenhagen  
{sil, nsa, sts}@smcnetwork.org

## ABSTRACT

Lorum Ipsum

## 1. INTRODUCTION

The behaviour of musical instruments can be well defined by partial differential equations (PDEs).

Finite-difference schemes (FDSs)

The physical models (PMs) used as a case study in this project are the stiff string and the plate.

This paper is structured as follows: Section 2 will present the PMs used in our implementation, Section 3 will show the how to implement the PMs, Section 4

## 2. MODELS

In this section, the partial differential equations for the damped stiff string and the plate will be presented.

### 2.1 Stiff string

The state  $u = u(x, t)$  describes the transverse displacement of the string. The subscript for  $u$  denotes a single derivative with respect to time  $t$  or space  $x$  respectively. The partial differential equation for the damped stiff string is defined as [1]

$$u_{tt} = \gamma^2 u_{xx} - \kappa^2 u_{xxxx} - 2\sigma_0 u_t + 2\sigma_1 u_{txx}, \quad (1)$$

where  $\gamma$  is wave-speed [m/s],  $\kappa$  is stiffness [?] and  $\sigma_0 \geq 0$  and  $\sigma_1 \geq 0$  are frequency-dependent and frequency-independent damping respectively.

We can add an excitation term to extend Equation (1) to a bowed string [1]

$$u_{tt} = \dots - \delta(x - x_B) F_B \phi(v_{\text{rel}}) \quad \text{where} \quad (2)$$

$$v_{\text{rel}} = u_t(x_B) - v_B, \quad (3)$$

where  $F_B$  is the bowing force [N],  $v_B$  is the bowing velocity [m/s],  $v_{\text{rel}}$  is the relative velocity, defined as the difference between the velocity of the string at bowing point  $x_B$  and the bowing velocity  $v_B$  [m/s] and  $\phi$  is a friction characteristic, which has been chosen to be [1]

$$\phi(v_{\text{rel}}) = \sqrt{2} a v_{\text{rel}} \exp(-a v_{\text{rel}}^2 + 1/2). \quad (4)$$

Copyright: © 2018 Silvin Willemsen et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

### 2.2 Plate

In the case of a plate, the state  $u = u(x, y, t)$  is now defined over two spatial dimensions. The PDE for a damped plate is [1]

$$u_{tt} = -\kappa^2 \Delta \Delta u - 2\sigma_0 u_t + 2\sigma_1 \Delta u_t, \quad (5)$$

where  $\Delta$  represents the 2D Laplacian (also see Equation (11)). Just like in the case of the string, an extra term can be added as an input:

$$u_{tt} = \dots + F_e E_e, \quad (6)$$

where  $F_e = f_e / \rho A L$  [m/s<sup>2</sup>], with bowing force  $f_e$  [N], density  $\rho$  [kg/m<sup>3</sup>], cross-sectional area  $A$  [m<sup>2</sup>] and string length  $L$  [m], and  $E_e$  are an excitation function and the excitation area respectively.

## 3. FINITE-DIFFERENCE SCHEMES

To be able to digitally implement the continuous equations mentioned in the previous section, they need to be approximated. The models can be discretised at times  $t = nk$ , where  $n \in \mathbb{W}$  and  $k = 1/f_s$  is the time-step with sample-rate  $f_s$  and locations  $x = lh$ , where  $l \in [0, N]$  with  $N$  being the total number of points and  $h$  is the grid-spacing of the model which is calculated differently for each model (see sub-sections below). The discretised variable  $u_l^n$  is  $u(x, t)$  at the  $n$ th time step and the  $l$ th point on the string. In the case of a plate, the second spatial dimension is discretised using  $x = lh$  where  $l \in [0, N_x]$  with  $N_x$  being the total horizontal number of points and  $y = mh$  where  $m \in [0, N_y]$  with  $N_y$  being the total vertical number of points. Approximations for the derivatives in the equations found in Section 2 are described in the following way:

When approximating the PDEs shown in Section 2, we use operators

$$\delta_{xx} u_l^n = \frac{1}{h^2} (u_{l+1}^n - 2u_l^n + u_{l-1}^n), \quad (7)$$

$$\delta_t u_l^n = \frac{1}{k} (u_l^n - u_l^{n-1}), \quad (8)$$

$$\delta_t u_l^n = \frac{1}{2k} (u_l^{n+1} - u_l^{n-1}), \quad (9)$$

$$\delta_{tt} u_l^n = \frac{1}{k^2} (u_l^{n+1} - 2u_l^n + u_l^{n-1}), \quad (10)$$

$$\delta_{\Delta} u_{l,m}^n = \frac{1}{h^2} (u_{l,m+1}^n + u_{l,m-1}^n + u_{l+1,m}^n + \quad (11)$$

$$u_{l-1,m}^n - 4u_{l,m}^n), \quad (12)$$

### 3.1 Stiff String

Equation (2) can be approximated using

$$\delta_{tt}u_l^n = \gamma^2\delta_{xx}u_l^n - \kappa^2\delta_{xx}\delta_{xx}u_l^n - 2\sigma_0\delta_t.u_l^n + 2\sigma_1\delta_t-\delta_{xx}u_l^n - \delta_{x_B}F_B\phi(v_{rel}), \quad (13)$$

where  $\delta_{x_B} = \delta(x - x_B)$ . For our implementation, we used clamped boundary conditions, defined as:

$$u = u_x = 0 \quad \text{where} \quad l = \{0, N\}. \quad (14)$$

For stability reasons, the grid-spacing needs to abide the following condition

$$h \geq \sqrt{\frac{\gamma^2k^2 + 4\sigma_1k + \sqrt{(\gamma^2k^2 + 4\sigma_1k)^2 + 16\kappa^2k^2}}{2}}. \quad (15)$$

The smaller  $h$  is, (i.e. the closer  $h$  is to this limit), the higher the quality of the implementation. The number of points  $N$  can then be calculated using

$$N = h^{-1}. \quad (16)$$

### 3.2 Plate

Equation (6) can be approximated using

$$\delta_{tt}u_{l,m}^n = -\kappa^2\delta_{\Delta}\delta_{\Delta}u_{l,m}^n - 2\sigma_0\delta_t.u_{l,m}^n - 2\sigma_1\delta_t\delta_{\Delta}u_{l,m}^n + F_eE_e \quad (17)$$

In the case of the plate, we set the number of horizontal and vertical points and calculate grid spacing  $h$  from that using

$$h = \frac{\sqrt{N_x/N_y}}{N_x} \quad (18)$$

## 4. CONNECTIONS

Adding connections between different PMs, further referred to as elements, adds another term to Equation (2) or (6)

$$u_{tt} = \dots + F_{\alpha}E_{\alpha}, \quad (19)$$

$$u_{tt} = \dots + F_{\beta}E_{\beta}, \quad (20)$$

where  $F_{\alpha}$  and  $F_{\beta}$  are the forces of the connection at connection areas  $E_{\alpha}$  and  $E_{\beta}$  respectively. If the a connection area consists of only one point,  $E$  reduces to  $\delta(x - x_c)$  where  $x_c$  is the point of connection. We use the implementation as presented in [2] where the connection between two elements is a non-linear spring. The forces it imposes on the elements it connects - denoted by  $\alpha$  and  $\beta$  - are defined as

$$F_{\alpha} = -\omega_0^2\eta - \omega_1^4\eta^3 - 2\sigma_{\times}\dot{\eta}, \quad (21)$$

$$F_{\beta} = -M_{\alpha/\beta}F_{\alpha}, \quad (22)$$

where  $\omega_0$  and  $\omega_1$  are the linear and non-linear spring coefficients respectively,  $\sigma_{\times}$  is the damping factor and  $M_{\alpha/\beta}$

is the mass ratio between the two elements. The relative displacement  $\eta$  between  $\alpha$  and  $\beta$  can be calculated as

$$\eta^n = h_{\alpha}u_{\alpha,x_{\alpha}}^n - h_{\beta}u_{\beta,x_{\beta}}^n, \quad (23)$$

which, in other words, is the difference between the state of element  $\alpha$  at connection point  $x_{\alpha}$  and the state of element  $\beta$  at connection point  $x_{\beta}$  scaled by their respective grid-spacings  $h_{\alpha}$  and  $h_{\beta}$ .

## 5. PITCH

We implemented a damping point in the model that acts as a finger on the (virtual) neck of the instrument controlling pitch. After  $u^{n+1}$  is calculated, the following operation is performed at the point of damping:

$$u_{x_f}^{n+1} = u_{x_f}^{n+1}\sigma_f, \quad (24)$$

where  $\sigma_f \in [0, 1]$  is the damping coefficient of the finger.

## 6. IMPLEMENTATION

For prototyping Matlab was used. We then used C++ along with the JUCE framework for implementing the objects and connections in real-time.

### 6.1 String

In order to implement the FDSs presented in Section 3 they need to be solved for  $u^{n+1}$ . In the case of the string, we obtain

$$\begin{aligned} u_l^{n+1} = & \left[ 2u_l^n - (1 - \sigma_0k)u_l^{n-1} \right. \\ & + \lambda^2(u_{l+1}^n - 2u_l^n + u_{l-1}^n) \\ & - \mu^2(u_{l+2}^n - 4u_{l+1}^n + 6u_l^n - 4u_{l-1}^n + u_{l-2}^n) \\ & + \frac{2\sigma_1k}{h^2}(u_{l+1}^n - 2u_l^n + u_{l-1}^n) \\ & - u_{l+1}^{n-1} + 2u_l^{n-1} - u_{l-1}^{n-1} \\ & \left. - k^2\delta_{l_e}F_e \right] \frac{1}{1 + \sigma_0k}, \end{aligned} \quad (25)$$

where

$$\lambda = \frac{\gamma k}{h}, \quad \mu = \frac{\kappa k}{h^2} \quad \text{and} \quad \delta_{l_e} = \delta(x - x_{l_e})$$

The wave-speed of the string is proportional to the fundamental frequency of the stiff string according to:

$$\gamma = 2f_0. \quad (26)$$

The output is retrieved at  $l = \text{floor}(0.75N)$  for all strings.

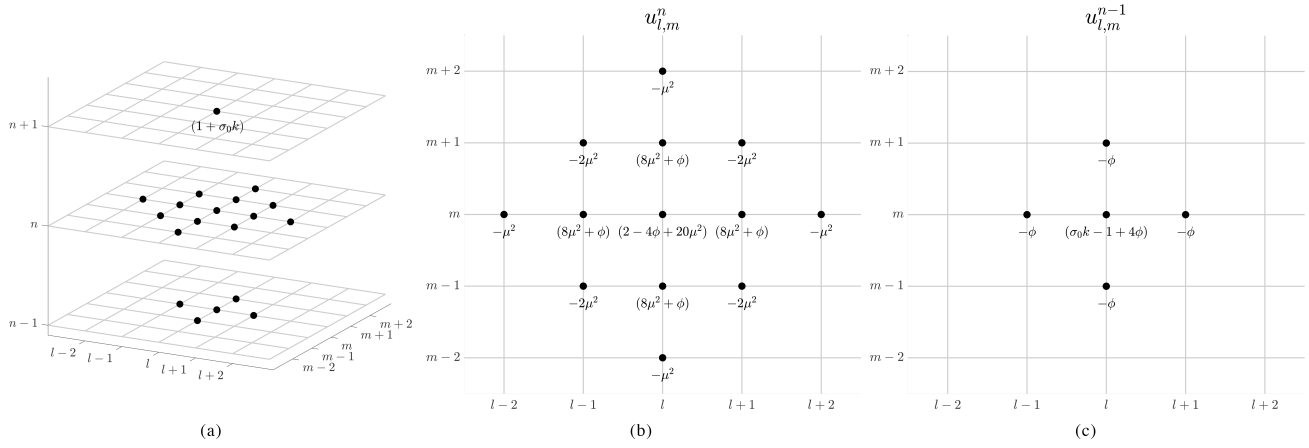


Figure 1. A visualisation of how to calculate  $u_{l,m}^{n+1}$  (Equation (27)). The dots represent the locations  $(l, m)$  and the equations represent with what these locations need to be multiplied. (a) An overview. (b) The current time-step  $n$ . (c) The previous time-step  $n - 1$ .

## 6.2 Plate

$$\begin{aligned}
 (1 + \sigma_0 k) u_{l,m}^{n+1} = & (2 - 4\phi + 20\mu^2) u_{l,m}^n \\
 & + (\sigma_0 k - 1 + 4\phi) u_{l,m}^{n-1} \\
 & - \mu^2 (u_{l,m+2}^n + u_{l,m-2}^n + u_{l+2,m}^n + u_{l-2,m}^n) \\
 & - 2\mu^2 (u_{l+1,m+1}^n + u_{l+1,m-1}^n + u_{l-1,m+1}^n + u_{l-1,m-1}^n) \\
 & + (8\mu^2 + \phi) (u_{l,m+1}^n + u_{l,m-1}^n + u_{l+1,m}^n + u_{l-1,m}^n) \\
 & - \phi (u_{l,m+1}^{n-1} + u_{l,m-1}^{n-1} + u_{l+1,m}^{n-1} + u_{l-1,m}^{n-1}) \\
 & + k^2 \delta_{l_e, m_e} F_e,
 \end{aligned} \tag{27}$$

where

$$\mu = \frac{\kappa k}{h^2}, \quad \phi = \frac{2\sigma_1 k}{h^2} \quad \text{and} \quad \delta_{l_e, m_e} = \delta(x - x_{l_e}, y - y_{m_e})$$

## 7. USER INTERACTION

User-controlled variables:

- Bowing position
- Bow force
- Bow velocity
- Connection points
- Finger position (pitch)

The vertical velocity of the finger is linked to the bow velocity with a maximum of  $V_b = 0.2$  m/s and the finger force is linked to the excitation function with a maximum of  $100 \text{ m/s}^2$ .

### 7.1 Sensel Morph

Something about the sensel morph

#### 7.1.1 Mapping strategies

Something about the different prototype mappings, and the "final" mapping

## 8. DISCUSSION

## 9. CONCLUSION AND FUTURE WORK

### Acknowledgments

We would like to thank...

## 10. REFERENCES

- [1] S. Bilbao, *Numerical Sound Synthesis, Finite Difference Schemes and Simulation in Musical Acoustics*. John Wiley and Sons, Ltd, 2009.
- [2] B. Bil, *Numerical Sound Synthesis, Finite Difference Schemes and Simulation in Musical Acoustics*. Proceedings of the 12th International Conference on Digital Audio Effects (DAFx-09), 2009.