XL CILAMCE
IBERO-LATIN AMERICAN
CONGRESS ON
COMPUTATIONAL
METHODS IN
ENGINEERING

NOVEMBER
11-14, 2019
Praiamar Natal Hotel & Convention
Natal, RN-BRAZIL

COMPARATIVE STUDY OF PHYSICAL MODELS FOR STRINGED INSTRUMENT SOUND SYNTHESIS

Guilherme N. Fontebasso[1]
Guilherme O. Paiva[2]
Lucas N. Egidio[1]
Rodolfo Thomazelli[2]
José M. C. dos Santos[1]
guilhermenishifonte@gmail.com
pitupaiva@gmail.com
egidio@fem.unicamp.br
rodolfo.thomazelli@gmail.com
zema@fem.unicamp.br

[1]*University of Campinas (UNICAMP)*
Rua Mendeleyev, 200, Cid. Univ. Zeferino Vaz, 13083-970, Campinas/SP, Brazil

[2]*Turbonius R&D*
Rua João Baptista Dalmédico 672, 13082-660, Campinas/SP, Brazil

Abstract. Despite the growing accessibility to computational power, sound synthesis via physical modelling still demands fast algorithms to allow its popularization. This work is concerned with a time-based comparison of two hybrid modal models for sound synthesis implemented in Python. One of them approaches the modal based synthesis problem coupling the string and body in the frequency domain. The other works in the modal domain, that is, it uses a finite difference scheme to integrate directly the modal differential equations governing the string motion coupled to the body. Both models are called "hybrid modal" because the string modal parameters, i.e., natural frequencies, modal shapes and damping factors, are obtained from analytical expressions, and the body modal parameters are extracted from numerical modal analysis. In this paper, a set of time-domain simulations including the planar and nonplanar motions of a monochord string is presented. The measurements of their execution time are obtained for the same computational setup, programming language and simulation parameters, delivering a quantitative comparison about the execution cost of these models. Furthermore, an asymptotic analysis for the proposed algorithms is also presented for the understanding of their scalability.

Keywords: Physical modelling, Algorithm performance, Modal-based methods

1 Introduction

With the constant growth in processing power offered by computers, the sound synthesis of musical instruments via physical modelling has become a topic of great interest in musical acoustics research. Comparatively, this modelling approach allows the realistic representation of the instrument under investigation, leading to a potential clarification of the relation between the physical properties of the instrument and the produced sound. The high computational cost, however, still prevents the design of tools and software applications to be popularized. In that regard, it is important to develop algorithms capable of optimizing time and cost of sound synthesis.

Several works have described computational methods for stringed musical instruments modelling and sound synthesis, *e.g.*, Woodhouse [1], Inácio and Antunes [2] Demoucron [3], Paté et al. [4], Bilbao and Torin [5], Desvages and Bilbao [6], Debut et al. [7] and Issanchou et al. [8]. The frequency domain method (FDM) presented by Woodhouse [1] stands out for two main characteristics: the proposed algorithm does not require space and time domain evaluations since it uses a system modal description combined with a frequency domain solution; and the strings and the instrument body are modelled separately, which leads to useful analysis possibilities and faster execution speed since the algorithm is relatively simple. On the other hand, the FDM does not allow the inclusion of nonlinear phenomena, which have been shown fundamental in the spectral composition of some instruments as shown by Debut et al. [7] and Paiva et al. [9]. In order to investigate the vibroacoustic characteristics of the *viola caipira*, which is a traditional Brazilian stringed instrument, Paiva et al. [9] proposed a modal-based method combined with a finite difference solution which allowed the inclusion of nonlinear strings collisions. Hereinafter such method is referred to as finite difference solution method (FDSM).

Based on the computational particularities and respective advantages of the above-mentioned sound synthesis methods, a comparative study of their performances has been shown to be of great interest. In this paper the FDM and FDSM are used to obtain a set of sound simulations of a monochord, which is a simplified stringed musical instrument comprising a rectangular wooden bar with a single nylon string attached at both ends. At one end the string is coupled to the monochord bridge and at the other end, to the tuner peg machine.

This paper is organized as follows: in Section 2 brief descriptions of the FDM and FDSM are presented. In Section 3, the simulation methodology details are explained as well as the model parameters determination and, finally, a set of monochord simulations using the FDM and FDSM are obtained, compared and discussed.

2 Sound synthesis methods

Brief descriptions of the FDM and FDSM for the nonplanar string motion case are presented below. For sake of concision, several formulation details are omitted in this paper. Nevertheless, the reader is invited to refer to Woodhouse [1] and Paiva et al. [9] papers for a thorough formulation of the FDM and FDSM, respectively. Both FDM and FDSM start from the description of the string and body properties as follows:

String. A small-amplitude vibration string is assumed and the following properties are assigned: length L, mass per unit length μ , tension T and bending stiness B. The string is simply supported at one end located at $x = 0$ and allowed to move at the other end located at $x_c = L$. The string transverse nonplanar motion is expressed in terms of the orthogonal components $y^s(x,t)$ and $z^s(x,t)$, oriented normal and parallel to the monochord soundboard plane, respectively. Axial and torsional string motions are neglected as well as its intrinsic geometric nonlinearities. Modal frequencies, mode shapes and damping factors are described as in Paiva et al. [9] paper.

Body. Accordingly, the body transverse nonplanar motion at the string coupling point $\mathbf{p_c} = (x_c, y_c, z_c)$ is expressed in terms of its orthogonal components $y^b(\mathbf{p_c}, t)$ and $z^b(\mathbf{p_c}, t)$, oriented normal and parallel to the monochord soundboard plane, respectively. A finite element analysis of the monochord body using Salome-Meca platform with Code-Aster solver is performed to extract modal frequencies and mode shapes at $\mathbf{p_c}$. Modal damping factor values are globally set to a 2.0%. The monochord body structure is modelled using Slash pine wood (*Pinus elliottii*) whose orthotropic mechanical properties are given in Green et al. [10] work. A mesh density of at least six quadratic elements per wavelength is used. Further information about the finite element model is displayed in Table 1.

Table 1. Finite element model information

Structure type	**Element features**			
	Geometry	Type	Modelling	Quantity
Plate	TRIA3	COQUE	DKT	3226
Solid	TETRA4	MASSIF	3D	10239

Finite difference solution method (FDSM). A generic orthogonal component w^s of the string displacement (with $w \in \{y, z\}$) is described as the sum of N_s modes associated to the string with simply supported ends to which is added an interface mode contribution $w_0^s(x,t)$ corresponding to the string static response when it is simply supported at $x = 0$ and loaded at $x_c = L$ so that

$$w^s(x,t) = \sum_{j=1}^{N_s} a_{j,w}(t) \sin\left(\frac{j\pi x}{L}\right) + w_0^s(x,t), \tag{1}$$

where $a_{j,w}(t)$ are the respective modal amplitudes. Additionally, the body transverse displacements at the coupling point $\mathbf{p_c}$ is described as the sum of N_b body modes:

$$w^b(x,t) = \sum_{k=1}^{N_b} b_k(t)\phi_{k,w}(\mathbf{p_c}), \tag{2}$$

where $\phi_{k,w}$ is the k^{th} body mode shape in the w direction, with $w \in \{y, z\}$. Within a modal framework, the motion equations for a forced response are formulated as a set of secondary-order ordinary differential equations which are written in the matrix form as follows:

$$\mathbf{M}^s\ddot{\mathbf{a}}(t) + \mathbf{C}^s\dot{\mathbf{a}}(t) + \mathbf{K}^s\mathbf{a}(t) = \mathbf{f}^e(t) - \mathbf{f}^c(t), \tag{3}$$

$$\mathbf{M}^b\ddot{\mathbf{b}}(t) + \mathbf{C}^b\dot{\mathbf{b}}(t) + \mathbf{K}^b\mathbf{b}(t) = \mathbf{f}^c(t), \tag{4}$$

where $\mathbf{a}(t)$ and $\mathbf{b}(t)$ are the string and body modal amplitude vectors, respectively. The matrices $\mathbf{M}^l$, $\mathbf{C}^l$ and $\mathbf{K}^l$ are the mass, damping and stiffness modal matrices associated to the string, for $l = s$, and to the body, for $l = b$, respectively. The plucking force is expressed in modal terms by the vector $\mathbf{f}^e$ in Eq. 3. String and body modal equations are coupled through the unknown string/body coupling force vector $\mathbf{f}^c$. Such vector is computed using a finite difference scheme described by Paiva et al. [9], which finally allows the string motion calculation. It is important to emphasise that this method requires a sample frequency five times bigger than the sample frequency of the resultant synthesized signal, accordingly to the convergence evaluation described by Paiva et al. [9].

Frequency domain method (FDM). This method allows to calculate the string motion resulting from a pluck taking into account that at the coupling point the string and the monochord body velocities are identical and the total force exerted is the sum of the forces applied to the two separate subsystems. Thus, it follows the relation

$$\mathbf{Y}^{sb}(\omega)^{-1} = \underbrace{\begin{bmatrix} Y^b_{yy}(\omega) & Y^b_{yz}(\omega) \\ Y^b_{zy}(\omega) & Y^b_{zz}(\omega) \end{bmatrix}^{-1}}_{\mathbf{Y^b}(\omega)^{-1}} + \underbrace{Y^s(\omega)^{-1} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}}_{\mathbf{Y^s}(\omega)^{-1}}, \tag{5}$$

where $\mathbf{Y}^{sb}$ represents the admittance matrix of the string/body coupled system. $\mathbf{Y}^b$ and $\mathbf{Y}^s$ represent, respectively, the body admittance and the string admittance at the coupling point. The expressions used to calculate such matrices are well described by Woodhouse [1]. The velocity response $G_w(\omega)$ at the coupling point (for $w \in \{y, z\}$) are calculated as follows:

$$\begin{bmatrix} G_y(\omega) \\ G_z(\omega) \end{bmatrix} = \mathbf{Y}^{sb}(\omega) \begin{bmatrix} H(\omega) & 0 \\ 0 & H(\omega) \end{bmatrix} \begin{bmatrix} \sin\beta \\ \cos\beta \end{bmatrix}, \tag{6}$$

where $H(\omega)$ is the transfer function from the plucking point to the end of the string as described by Woodhouse [1] and β is the plucking angle measured from the y direction. The velocities at the coupling point are finally obtained:

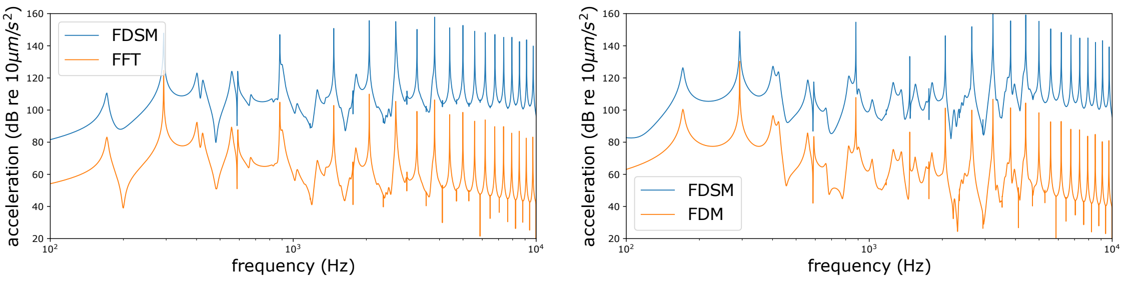
$$\dot{w}^s(x_c, t) = \dot{w}^b(\mathbf{p_c}, t) = \text{real}\{\mathscr{F}^{-1}\{G_w(\omega)\}\}. \tag{7}$$

3 Numerical simulations

The models were implemented in Python using the NumPy library. Diverging from the computational implementation presented by Woodhouse [1] and Paiva et al. [9], some optimizations were made in both methods.

Model parameters. The string parameters used were $L = 0.582$m, $\mu = 0.527$g/m, $T = 61.57$N. The pluck consists in an impulse of 1N, $\beta = 45°$, and was applied at the middle of the string. The chosen sample frequency was 44100 Hz. The maximal frequency considered for the string was 10kHz, the most significant modes accordingly to Paiva et al. [9], and for the simulated monochord body, 5 kHz, the most significant modes accordingly to Woodhouse [1]. For those frequency ranges, 69 string modes and 89 body modes were used.

Model validation. To validate the implementations, a comparison between the results of the models for the same inputs was made, and their frequency response for the y and z directions were compared as shown for z in the Figure 1a, and for y in the Figures 1b. Therefore, the implementation was validated.



(a) z direction

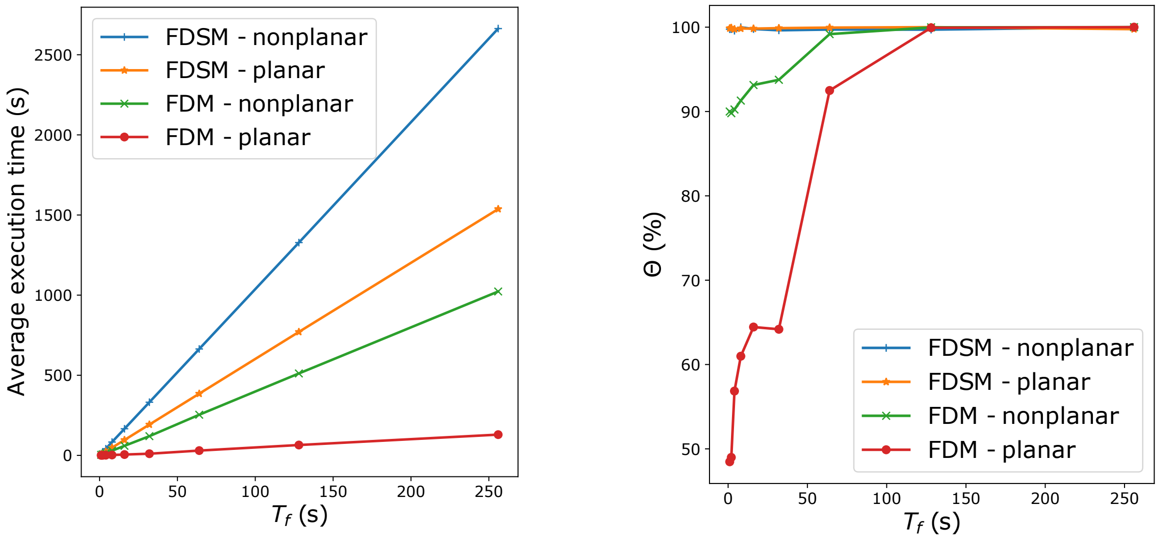
(b) y direction

Figure 1. Comparison between FDM and FDSM frequency response for z and y directions. The FDM curves were shifted -50 dB for better visibility and comparison.

Asymptotic analysis. The finite difference solution method asymptotic complexity is $\mathcal{O}(n)$, considering the linearity of the finite difference method algorithm, and the frequency domain method is $\mathcal{O}(n \log n)$, considering the asymptotic complexity of the FFT algorithm, where $n = T_f F_s$, and T_f is the length in seconds of the synthesized vector at the coupling point, and F_s is the sampling frequency in Hertz, fixed in this study as 44100 Hz.

Setup and methodology. Regarding the simulation, 12 virtual machines (VMs) on Google Cloud servers, with 7.5 Gb of RAM and 1 vCPU, were used, working as a single hardware hyper-thread on a Intel Xenon Scalable Processor (Skylake), with base frequency of 2 GHz and maximal overclock frequency of 3.5 GHz. As no parallelism was employed, the usage of multi-vCPUs systems provides no performance difference. The 12 virtual machines were chosen from 3 locations, in a way that each of the 4 algorithms, were simulated at the same 3 locations simultaneously. For each model, the execution time was measured by varying the length of the synthesized sound T_f from 1 to 256 seconds, choosing values as 2^i s, where $i = 0, 1, 2, ..., 7, 8$. The choose of the maximal length was made considering the average length of a commercial song.

Results and discussion. The simulation results are shown in the Figure 2. It is noticeable from the Figure 2a, that the FDSM is slower than the FDM, for planar and nonplanar motion, and for all synthesized signal lengths (T_f). In the Figure 2b, Θ is defined as the normalized average execution time per second of synthesized signal. In this representation, $\mathcal{O}(n \log n)$ and $\mathcal{O}(n)$ are divided by n, resulting in $\mathcal{O}(\log n)$ and $\mathcal{O}(1)$, helping thereby to indicate the asymptotic behavior of both methods. It is important to notice that after 128 seconds of synthesized signal, the FDM starts to behave linearly, with slope smaller than the FDSM curves.



(a) FDSM and FDM average execution times, for planar and nonplanar motions, and different lengths of synthesized signal

(b) Normalized FDSM and FDM average execution times for planar and nonplanar motions, per second of synthesized signal, for different lengths of synthesized signal

Figure 2. FDSM and FDM time-cost analysis for planar and nonplanar motions.

Due to the linearity of the FDSM curves, it is possible to extract their average execution times per second of synthesized signal, for any length of synthesized signal. For planar motion, the average value is 6.01 seconds, with coefficient of variation (CV) of 0.076%, and for nonplanar motion, the average value is 10.38 seconds, with CV of 0.134%. For the linear part of the FDM curves, the values are 0.506 seconds with CV of 0.034% for planar, and 3.994 seconds with CV of 0.011% for nonplanar motion. Therefore, for the synthesis of signals longer than 128 seconds, the FDM method is 11.88 times faster than the FDSM for planar, and 2.6 times faster, for nonplanar motion.

4 Conclusions

The simulation pointed a clear difference in computational cost between the FDM and FDSM. Despite the logarithmic behavior of the FDM algorithm, it has been shown that it is faster than the FDSM for all the measured values.

It is important to notice that the FDM has limitations regarding nonlinear phenomena inclusion, unlike the FDSM, what can compromise the realism of the resulting synthesized sound. However, nonlinearities in the FDSM require a larger sampling frequency as pointed by Paiva et al. [9], which is proportional to the execution time for this method.

Therefore, for further works, nonlinear phenomena like string/string collisions, string/fret collisions and geometric string nonlinearities can be included to the FDSM. Moreover, psychometric tests to evaluate the importance of those phenomena for synthesized sounds realism must be performed in order to validate them in terms of perceptual aspects .

Acknowledgements

The authors are grateful to the Brazilian research funding agencies CAPES, CNPq (Grant No. 313620/2018) and FAPESP (Grants No. 2019/02882-7 and 2018/08721-2).

References

[1] Woodhouse, J., 2004. On the synthesis of guitar plucks. *Acta Acustica United With Acustica*, vol. 90, pp. 928–944.

[2] Inácio, O. & Antunes, J., 2008. Computational modelling of string–body interaction for the violin family and simulation of wolf notes. *Journal of Sound and Vibration*, vol. 310, pp. 260–286.

[3] Demoucron, M., 2008. *On the control of virtual violins-Physical modelling and control of bowed string instruments*. PhD thesis, Université Pierre et Marie Curie-Paris VI; Royal Institute of Technology, Stockholm, Sweden.

[4] Paté, A., Carrou, J. L., , & Fabre, B., 2014. Predicting the decay time of solid body electric guitar tones. *The Journal of the Acoustical Society of America*, vol. 135, pp. 3045–3055.

[5] Bilbao, S. & Torin, A., 2015. Numerical modeling and sound synthesis for articulated string/fretboard interactions. *Journal of the Audio Engineering Society*, vol. 63, pp. 336–347.

[6] Desvages, C. & Bilbao, S., 2016. Two-polarisation physical model of bowed strings with nonlinear contact and friction forces, and application to gesture-based sound synthesis. *Applied Sciences*, vol. 6.

[7] Debut, V., Antunes, J., Marques, M., & Carvalho, M., 2016. Physics-based modeling techniques of a twelve-string portuguese guitar: A non-linear time-domain computational approach for the multiple-strings/bridge/soundboard coupled dynamics. *Applied Acoustics*, vol. 108, pp. 3–18.

[8] Issanchou, C., Carrou, J. L., Touzé, C., Fabre, B., , & Doaré, O., 2018. String/frets contacts in the electric bass sound: Simulations and experiments. *Applied Acoustics*, vol. 129, pp. 217–228.

[9] Paiva, G. O., Ablitzer, F., Gautier, F., & dos Santos, J. M. C., 2019. Collisions in double string plucked instruments: Physical modelling and sound synthesis of the viola caipira. *Journal of Sound and Vibration*, vol. 443, pp. 178–197.

[10] Green, D. W., Winandy, J. E., & Kretschmann, D. E., 1999. Mechanical properties of wood. *Wood handbook: wood as an engineering material. Madison, WI: USDA Forest Service, Forest Products Laboratory, 1999. General technical report FPL; GTR-113: Pages 4.1-4.45*, vol. 113.