Percussion Instrument Modelling In 3D: Sound Synthesis Through Time Domain Numerical Simulation

Alberto Torin

Doctor of Philosophy
University of Edinburgh
2015

Abstract

This work is concerned with the numerical simulation of percussion instruments based on physical principles. Three novel modular environments for sound synthesis are presented: a system composed of various plates vibrating under nonlinear conditions, a model for a nonlinear double membrane drum and a snare drum. All are embedded in a 3D acoustic environment. The approach adopted is based on the finite difference method, and extends recent results in the field. Starting from simple models, the modular instruments can be created by combining different components in order to obtain virtual environments with increasing complexity. The resulting numerical codes can be used by composers and musicians to create music by specifying the parameters and a score for the systems. Stability is a major concern in numerical simulation. In this work, energy techniques are employed in order to guarantee the stability of the numerical schemes for the virtual instruments, by imposing suitable coupling conditions between the various components of the system.

Before presenting the virtual instruments, the various components are individually analysed. Plates are the main elements of the multiple plate system, and they represent the first approximation to the simulation of gongs and cymbals. Similarly to plates, membranes are important in the simulation of drums. Linear and nonlinear plate/membrane vibration is thus the starting point of this work. An important aspect of percussion instruments is the modelling of collisions. A novel approach based on penalty methods is adopted here to describe lumped collisions with a mallet and distributed collisions with a string in the case of a membrane. Another point discussed in the present work is the coupling between 2D structures like plates and membranes with the 3D acoustic field, in order to obtain an integrated system. It is demonstrated how the air coupling can be implemented when nonlinearities and collisions are present. Finally, some attention is devoted to the experimental validation of the numerical simulation in the case of tom tom drums. Preliminary results comparing different types of nonlinear models for membrane vibration are presented.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Alberto Torin*)

List of publications

Elements of this thesis have been published in the following papers.

(A) Alberto Torin and Stefan Bilbao. Numerical experiments with non-linear double membrane drums. In *Proceedings of the Stockholm Music Acoustics Conference 2013, SMAC 2013*, Stockholm, Sweden, 2013.

(B) Stefan Bilbao, Brian Hamilton, Alberto Torin, Craig Webb, Paul Graham, Alan Gray, Kostas Kavoussanakis, and James Perry. Large scale physical modeling sound synthesis. In *Proc. 4th Stockholm Musical Acoustics Conference (SMAC 2013)*, Stockholm, Sweden, 2013.

(C) Alberto Torin and Stefan Bilbao. A 3D Multi-Plate environment for sound synthesis. In *Proc. of the 16th Int. Conference on Digital Audio Effects (DAFx-13)*, Maynooth, Ireland, 2013.

(D) Alberto Torin and Michael Newton. Collisions in drum membranes: a preliminary study on a simplified system. In *Proc. of the Int. Symposium on Musical Acoustics (ISMA 2014)*, Le Mans, France, 2014.

(E) Alberto Torin and Michael Newton. Nonlinear effects in drum membranes. In *Proc. of the Int. Symposium on Musical Acoustics (ISMA 2014)*, Le Mans, France, 2014.

(F) Stefan Bilbao, Alberto Torin, Paul Graham, James Perry, and Gordon Delap. Modular physical modeling synthesis environments on GPU. In *Proc. 2014 International Computer Music Conference*, Athens, Greece, 2014.

(G) Alberto Torin, Brian Hamilton, and Stefan Bilbao. An energy conserving finite difference scheme for the simulation of collisions in snare drums. In *Proc. of the 17th Int. Conference on Digital Audio Effects (DAFx 14)*, Erlangen, Germany, 2014.

(H) Stefan Bilbao and Alberto Torin. Numerical simulation of string/barrier collisions: The fretboard. In *Proc. of the 17th Int. Conference on Digital Audio Effects (DAFx 14)*, Erlangen, Germany, 2014.

(I) Brian Hamilton and Alberto Torin. Finite difference schemes on hexagonal grids for thin linear plates with finite volume boundaries. In *Proc. of the 17th Int. Conference on Digital Audio Effects (DAFx 14)*, Erlangen, Germany, 2014.

(J) Stefan Bilbao, Alberto Torin, and Vasileios Chatziioannou. Numerical modeling of collisions in musical instruments. *Acta Acustica united with Acustica*, 101(1):155–173, 2015.

(K) Stefan Bilbao and Alberto Torin. Numerical modeling and sound synthesis for articulated string/fretboard interactions. *Journal of the Audio Engineering Society*, 63(5):336–347, 2015.

(L) James Perry, Stefan Bilbao, and Alberto Torin. Hierarchical parallelism in a physical modelling synthesis code. In *ParCo Conference*, Edinburgh, UK, 2015.

Acknowledgements

I am indebted to many people that helped me during these years as a PhD student. First, I would like to thank Stefan Bilbao, whose underlying support has been essential to move my first steps in this fascinating subject, and to dovetail the various pieces of this work. A warm *merci beaucoup* goes to Cyril Touzé, for the useful discussions during my stay in Paris and via email. Thanks also to Olivier Thomas, whose help has been essential in understanding some of the subtleties of nonlinear plate models. I am very grateful to my examiners, Antoine Chaigne and John Chick, for their important suggestions on how to improve this work.

I would like to thank my colleagues in the Acoustics and Audio Group for the fruitful work done together. My *PhD twin* Brian Hamilton deserves a special mention for his innumerable tricks and hacks that contributed to make my computing life easier. Many thanks to Craig Webb, Michele Ducceschi, Reg Harrison, Charlotte Desvages and Amaya López-Carromero. Thanks also to Mike Newton for the great tips on experimental setups and cullen skink.

Many thanks to the people involved in the NESS Project for their help during these years: Kostas Kavoussanakis, Alan Gray, James Perry, Paul Graham, Adrian Mouat.

This work would not have been possible without the financial support of the European Research Council under Grant StG-2011-279068-NESS.

Finally, I would like to thank my family. Thanks also to Rita and Emilio for their hospitality during the summer 2015. And thanks Cristina, for everything! This work is dedicated to you.

To Cristina,
to Giovanni and his snare drum

List of symbols

α	Collision nonlinear exponent, *p. 82*
η_c	Air/membrane coupling constant, Eq. (5.59), *p. 119*
κ	Plate coefficient, *p. 15*
ν	Poisson's ratio, *p. 14*
Φ	Airy's stress function, Eq. (3.25), *p. 63*
Π	Collision potential, Eq. (4.3), *p. 84*
Ψ	Acoustic velocity potential, *p. 107*
ρ	Density, *p. 15*
ρ_a	Density of air, *p. 107*
σ_0	Frequency independent loss coefficient, *p. 20*
σ_1	Frequency dependent loss coefficient, *p. 20*
ξ	Berger plate coefficient, Eq. (3.2), *p. 58*
A	Cross sectional area of a string, *p. 91*
c_a	Speed of sound in air, *p. 107*
D	Bending stiffness, Eq. (2.40), *p. 15*
E	Young's modulus, *p. 14*
F_s	Sampling rate of a simulation, *p. 26*
$g(x, y)$	Spatial distribution, *p. 88*
H	Thickness, *p. 14*
h	Spatial grid step of a simulation, *p. 26*
I_s	Moment of inertia, *p. 91*
K	Collision coefficient, *p. 82*
k	Time step of a simulation, *p. 26*
M	Mass of the lumped mallet, *p. 83*
T/T_m	Membrane tension, *p. 20*
T_s	String tension, *p. 91*
u	Vertical displacement of the string, *p. 91*
w	Vertical displacement of the plate/membrane, *p. 14*
$\partial\mathcal{D}$	Boundary of a 2D domain, *p. 14*
$\mathcal{D}$	2D domain, *p. 14*
$\mathcal{D}_s$	String domain (1D), *p. 91*
$\mathcal{V}$	3D domain, *p. 105*
$\mathfrak{D}_s$	Discrete string domain (1D), *p. 92*
$\mathfrak{D}$	Discrete 2D domain, *p. 26*
$\mathfrak{V}$	Discrete 3D domain, *p. 105*
$\mathcal{L}$	von Kármán operator (continuous case), Eq. (3.29), *p. 64*
$\mathfrak{T}$	Berger operator (continuous case), Eq. (3.2), *p. 58*
$\mathcal{O}_m$	Linear differential operator, stiff membrane (continuous case), Eq. (2.64), *p. 21*
$\mathcal{O}_s$	Linear differential operator, stiff string (continuous case), Eq. (4.39), *p. 91*
ϕ_ν	Plate tensor mapping, Eq. (2.26), *p. 12*
$\mathfrak{O}_m$	Finite difference operator, stiff membrane (discrete case), Eq. (2.165), *p. 49*
$\mathfrak{O}_s$	Finite difference operator, stiff string (discrete case), Eq. (4.43), *p. 92*
$\mathfrak{l}$	von Kármán operator (discrete case), Eq. (3.50), *p. 67*
$\mathfrak{t}$	Berger operator (discrete case), Eq. (3.9), *p. 60*
$\mathfrak{H}_a$	Energy for the acoustic field (continuous case), Eq. (5.16), *p. 108*
$\mathfrak{H}_{\text{Berger}}$	Energy for the Berger plate (continuous case), Eq. (3.7), *p. 59*

$\mathfrak{H}_\mathrm{M}$	Energy for the mallet (continuous case), Eq. (4.5), *p. 84*
$\mathfrak{H}_\mathrm{pl}$	Energy for the plate (continuous case), Eq. (2.51), *p. 18*
$\mathfrak{H}_\mathrm{rpl}$	Energy for the rectangular plate (continuous case), Eq. (2.73), *p. 22*
$\mathfrak{H}_\mathrm{sm}$	Energy for the stiff membrane (continuous case), Eq. (2.63), *p. 21*
$\mathfrak{H}_\mathrm{s}$	Energy for the string (continuous case), Eq. (4.41a), *p. 92*
$\mathfrak{H}_\mathrm{vkp}$	Energy for the von Kármán plate (continuous case), Eq. (3.41), *p. 65*
$\mathfrak{B}_\mathrm{pl}$	Boundary energy for the plate (continuous case), Eq. (2.49), *p. 18*
$\mathfrak{B}_\mathrm{rpl}$	Boundary energy for the rectangular plate (continuous case), Eq. (2.74), *p. 23*
$\mathfrak{B}_\mathrm{s}$	Boundary energy for the string (continuous case), Eq. (4.41c), *p. 92*
$\mathfrak{B}_a$	Boundary energy for the acoustic field (continuous case), Eq. (5.17), *p. 108*
$\mathfrak{B}_\mathrm{vkp}$	Boundary energy for the von Kármán plate (continuous case), Eq. (3.41), *p. 65*
$\mathfrak{Q}/\mathfrak{Q}_m$	Dissipated power, plate/membrane equation (continuous case), Eq. (2.59), *p. 20*
$\mathfrak{Q}_s$	Dissipated power, string equation (continuous case), Eq. (4.41b), *p. 92*
$\mathfrak{h}_\mathrm{Berger}$	Energy for the Berger plate (discrete case), Eq. (3.13), *p. 60*
$\mathfrak{h}_\mathrm{a}$	Energy for the acoustic field (discrete case), Eq. (5.30), *p. 110*
$\mathfrak{h}_\mathrm{M}$	Energy for the mallet (discrete case), Eq. (4.10), *p. 85*
$\mathfrak{h}_\mathrm{rpl}$	Energy for the rectangular plate (discrete case), Eq. (2.124), *p. 33*
$\mathfrak{h}_\mathrm{s}$	Energy for the string (discrete case), Eq. (4.45a), *p. 92*
$\mathfrak{h}_\mathrm{vkp}$	Energy for the von Kármán plate (discrete case), Eq. (3.57), *p. 68*
$\mathfrak{e}^n$	Normalised energy variation, Eq. (2.151), *p. 39*
$\mathfrak{b}_\mathrm{rpl}$	Boundary energy for the rectangular plate (discrete case), Eq. (2.128), *p. 34*
$\mathfrak{b}_a$	Boundary energy for the acoustic field (discrete case), Eq. (5.31), *p. 110*
$\mathfrak{b}_\mathrm{vkp}$	Boundary energy for the von Kármán plate (discrete case), Eq. (3.57), *p. 68*
$\mathfrak{q}$	Dissipated power, plate/membrane equation (discrete case), Eq. (2.159), *p. 48*
$\mathfrak{q}_s$	Dissipated power, string equation (discrete case), Eq. (4.45b), *p. 92*

Contents

Chapter 1

Introduction

1.1 Brief historical background on sound synthesis

Digital sound synthesis is the process of generating sound using electronic devices (either hardware or software). Starting from the 1950s, several techniques were used to this aim, like additive synthesis, subtractive synthesis, wavetable synthesis and AM/FM synthesis, just to mention a few. More on these approaches can be found in [132, 71, 150]. All these techniques share a very simple implementation and an extreme efficiency, although they often depend on many parameters that are perceptually difficult to interpret. Such methods can be defined abstract, as they lack an underlying physical interpretation.

An alternative to these abstract methods is given by physical modelling synthesis. This approach requires the knowledge of the physical description of a musical instrument, together with a numerical method to simulate it. The goal is to produce realistic simulations of the target systems that can be controlled in a simple way by a musician or composer. Several methods have been proposed in the past 60 years, and the interested reader is referred to [191, 19] for a complete overview.

Historically, the first physical model in the context of musical acoustics is that of a vocal tract by Kelly and Lochbaum [112] in 1962, while the first simulation of a vibrating string is due to Ruiz [156] and to Hiller and Ruiz [104, 105] in the early 70s. Lumped networks of interacting masses with springs and dampers were proposed by Cadoz in 1979 [41] and became the basis for the CORDIS and CORDIS-ANIMA synthesis environments [42]. A later approach based on modal decomposition was first developed by Adrien at IRCAM [1]. Broadly speaking, each vibrating mode of the instrument is associated with a frequency and with a damping coefficient, and their evolution can be calculated independently (in the linear case) based on the initial conditions. This method was at the basis of the MOSAIC-Modalys system [133, 76]. Digital waveguides were introduced in the 1980s by Smith [162, 164], and became probably the most popular synthesis technique for the following twenty years. Improvements and generalisations, like the use of wave digital filters, originally developed by Fettweis [82] in the context of digital filter design, favoured the application of this technique to a wide variety of systems [163, 64, 194], but see also [191] for a more comprehensive list. Commercial products based on digital waveguides were soon developed, the most notable of which is perhaps the Yamaha VL1 synthesiser. Finally, grid-based methods like finite difference, finite elements and finite volumes have been gaining momentum in the last years. Although the first work employing this approach is the already cited Ruiz model, new significant advances in this field had to wait a couple of decades, until the works by Chaigne and collaborators (see below for further details). The new goal, beside sound synthesis, was to produce numerical models to help the physical investigation of real instruments. This approach is now becoming widespread due to the availability of more powerful computational resources.

1.2 Finite difference method

Acoustical simulation using the finite difference method is today an important topic of research. The idea behind this approach is very simple: the model describing the instrument at hand is first discretised over a spatial and temporal grid; then, the spatial differential operators are substituted with combinations of neighbouring grid points; finally, the time integration of the partial differential equation is performed as a recursion on the previously computed values of the grid functions. A detailed analysis of this method is given in the classic textbooks by Strikwerda [168] and by Gustafsson, Kreiss and Oliger [92].

The concept of finite differences is very old, and is related with the very definition of derivative. From the formula

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \lim_{h\to 0} \frac{f(x+h)-f(x)}{h},$$

which defines the derivative $\mathrm{d}f/\,\mathrm{d}x$ of a function $f(x)$, it is straightforward to devise a finite difference calculation in terms of a fixed (and small) value for h. Some important attempts at solving partial differential equations with finite differences are due to Richardson in the context of weather prediction, and date back to 1922 [148]. It was only in 1928, however, that this method became popular, thanks to the seminal work of Courant, Friedrichs and Lewy [65]. From a historical point of view, it is interesting that the Second World War provided a great impetus to the development of time domain finite differences, thanks to the appearance of the first digital computers [38]. Important contributions were also given by Lax and Rycthmyer [122] and von Neumann [199]. This was well before the invention of the so-called finite difference time domain (FDTD) method by Yee [208], popularised later by Taflove [172]. Other historical developments are reported in [178], to which the interested reader is referred.

In the field of musical acoustics and sound synthesis, finite difference methods were used sporadically at first, with the Ruiz [156] and Hiller and Ruiz [104, 105] model of a string being the first attempt of this kind. Other important contributions are the works of Portnoff [143], Bacon and Bowsher [10] and Boutillon [36]. In more recent years, the increase of computational power has allowed more complex models to be developed, namely plucked stringed instruments [48] and piano string/hammer interaction [49, 50], later extended with the inclusion of a soundboard [89]. Finite difference methods were then successfully applied to the simulation of percussion instruments such as xylophones [51, 72], while a hybrid finite element/finite difference approach was adopted in the simulation of timpani drums [146, 52]. As this thesis is concerned also with 3D simulation, it is important to acknowledge the contribution of Botteldooren in room acoustic modelling [33, 34]. Since the 2000s, advances in the field have been rapid and diverse, so no attempt is made in this section to track the vast literature that has been produced. The works relevant to this thesis will be acknowledged in the introductions to the various chapters. The reader is referred to the textbook by Bilbao [19] for an extensive treatment of the matter.

1.3 Percussion instruments

Percussion instruments have been an interesting topic of research in musical acoustics for a long time. At the end of the 19th century, Lord Rayleigh was probably the first to report quantitative observations regarding the vibration of timpani drum membranes in his *Theory of sound* [169]. The pioneers in systematic percussion instruments studies are Fletcher and Rossing, who collected the findings of their many works in their classic textbooks [84, 154, 152].

As far as simulation is concerned, the design of a numerical model can be a particularly challenging task for at least three reasons. First of all, the instrument to be modelled must be considered as a 3D object—the membrane of a timpani drum, e.g., interacts with the volume of air enclosed by the cavity. Related to this is the fact that many disparate components are present at the same time. In a snare drum, for example, two membranes are coupled together by a shell enclosing an air cavity, with a set of thin metal wires in contact with the lower membrane. Finally, percussion instrument simulation is complicated by the presence of multiple strong nonlinearities, both in the collision interactions with mallets and snares, as well as in the drumhead vibration itself. Several previous attempts have been made in the past to simulate drums and percussion in general, and they are acknowledged in the introductory remarks to Chapter 6. This manuscript gives a novel contribution to this particular field of research.

1.4 Thesis objectives and outline

This thesis is concerned with the simulation of percussion instruments in 3D. The aim is to produce virtual instruments based on the physical behaviour of gongs and drums that are embedded in a 3D space and are coupled with the surrounding air. This work also presents original contributions to the modelling of distributed collisions [27]. It is demonstrated how different components can be combined together in a modular way in order to create three different virtual instruments: a system composed of several plates vibrating under nonlinear conditions in 3D, the nonlinear double membrane drum and the snare drum.

Although several approaches are possible to the modelling of musical instruments, all the numerical codes presented here rely for their design on energy techniques [128]. Every algorithm has, in fact, an associated numerical quantity, which can be related to the energy of the system in the continuous case. By ensuring the non-negativity of the discrete energy, the stability of the numerical scheme can be guaranteed. This approach finds extensive use in the work of Bilbao and collaborators [19, 96, 69], van Walstijn and Chatziioannou [58], Chabassier [46], Falaize, Lopes, Hélie and collaborators [81, 80], albeit with some important variations.

Being subject to the "tyranny of energy"[1], according to Chabassier's witty definition [46, p. 219], might seem an excessive burden, given that several alternative schemes are generally available. Energy methods, however, offer several advantages, and they are particularly useful when strong nonlinearities are present and other numerical analysis techniques (like frequency domain analysis) cannot be applied. Furthermore, without this tool the stability of the algorithms in the presence of many interacting components could hardly be ascertained. In this case, the numerical energy of the scheme becomes a powerful debugging tool which can be used

---

[1] "La dictature de l'énergie", in the original manuscript in French.

to identify the possible sources of error. Conservation ideas here are not limited to the semi-discrete spatial problem, as in the case of the already cited [46], but require the calculation of a discrete energy in both time and space.

The basic ideas regarding the functioning of the finite difference method have been discussed in the previous section. Most of the finite difference schemes that will appear in the rest of this manuscript can ultimately be written as a recursion which takes the following form [23]

$$\mathbf{A}^n\mathbf{w}^{n+1} = \mathbf{B}^n\mathbf{w}^n - \mathbf{C}^n\mathbf{w}^{n-1} + \mathbf{k}^n$$

for each single component of the scheme. Here, $\mathbf{w}$ represents the values of the grid function, in vector form. This could be, e.g, the displacement of a drum membrane, or the pressure of the acoustic field. $\mathbf{A}^n$, $\mathbf{B}^n$ and $\mathbf{C}^n$ are square matrices, possibly dependent on time, that incorporate the action of the finite difference operators, and $\mathbf{k}^n$ is a vector that depends on known quantities. When $\mathbf{A}^n$ is the identity matrix, then the resulting scheme is explicit, otherwise the solution of a linear system is needed, and the scheme is said to be implicit. An important exception to this representation is given by schemes that involve collisions—in this case, the solution of a nonlinear equation is required at every time step. Note that the update is second order, reflecting the order of the differential equation, and is thus minimal in terms of memory requirements. Such is not the case when standard integration techniques like Runge-Kutta are used. This is done to improve the efficiency of the numerical implementations.

One disadvantage of finite difference simulations is their computational cost, both in terms of memory requirements and operation count. For this reason, large scale problems have been out of the capabilities of computers for a long time, and even smaller systems generally require some time to be computed. Recent works [159, 202, 99] aim at exploring strategies to speed up the computation of finite difference schemes in sound synthesis applications. In recent years, in fact, the increased availability of graphical processing units (GPUs) has opened new avenues of research in this sense. Even the simulation of large 3D spaces at audio rate is becoming tractable, and this has been the subject of a recent work by Webb [201]. With clever implementation strategies, smaller problems, like networks of plates and strings, have also been implemented as a real-time plug-in for commercial laptops [203]. The virtual instruments presented in this work have been successfully ported to a hybrid GPU/CPU implementation, speeding up considerably the run time with respect to the original prototyping code designed by the Author [28, 142]. This topic, however, will not be discussed in detail, though further insights are provided in Chapter 6.

The outline of this manuscript is given below. Each chapter is accompanied by an individual introduction that describes the topic and acknowledges the previous works. Supplementary material, such as audio and video files, can be found at the Author's web page:

`http://www2.ph.ed.ac.uk/~s1164558/`

Chapter 2 - Linear plate vibration

The basic equations describing the transverse linear vibration of thin plates according to the Kirchhoff model are discussed for rectangular and circular geometries, and particular attention is devoted to the link between energy and boundary conditions in the continuous case. Then, a brief introduction on the finite difference method is presented, together with the notation adopted throughout this work. Finite difference schemes for rectangular and circular plates are presented. A complete discussion of energy-based stability techniques including the derivation of boundary conditions is provided, together with simulation results demonstrating the stability of the numerical implementations.

Chapter 3 - Nonlinear plate vibration

Two different models for the nonlinear vibration of thin plates are discussed. First, the Berger model is presented, together with the continuous time equations and one possible finite difference implementation. Numerical results are presented. A second, more extended section is devoted to the von Kármán nonlinearity. The underlying model is presented in continuous time first, together with a discussion on its energy properties. An energy conserving finite difference scheme is then presented. Different implementation strategies are evaluated, and their results compared. The relevance of this model for musical acoustics is also discussed. Finally, a brief section explores the extension of the von Kármán model to membranes, which is necessary to obtain more realistic sounds from virtual drums. This constitutes one of the contributions of this work.

Chapter 4 - Collisions

In this chapter, a recently presented model for the simulation of collisions is explored. In the first section, the simple case of a lumped object colliding with a rigid barrier is presented as a test case, in order to introduce the notation and the basic concepts. The energy conservation properties of this new implementation are tested. A second section is devoted to the lumped/distributed collision, which models the interaction between a mallet and a plate or membrane. Finally, the distributed/distributed collisions between a string and a membrane will be discussed. The continuous time model is derived, together with an energy balance for the system. A finite difference scheme and its implementation is given. Particular attention is devoted to the stable termination of the string on the membrane. Simulation results are presented. The extension of the collision model to the distributed/distributed case and its numerical implementation is one of the contributions of this work.

Chapter 5 - Coupling with the acoustic field in 3D

The 3D wave equation for the acoustic field is the main subject of this chapter. The basic equations in terms of an acoustic velocity potential are first discussed in the lossless case. Then, the numerical discretisation of this model is performed. The core of the chapter deals with embeddings, that is the coupling between 2D structures and the 3D field. After describing the simple case of a linear membrane, air coupling is extended to several different systems, including Berger nonlinearity, von Kármán nonlinearity, the mallet/membrane collision and the string/membrane collisions. Air coupling in the presence of von Kármán nonlinearity and collisions is one of the contributions of this work.

Chapter 6 - Virtual instruments

This chapter presents the ultimate contribution of this thesis. Three different virtual instruments are presented: a system of multiple plates in 3D, the nonlinear double membrane drum (a model for bass drums and tom toms), and finally the snare drum. All these instruments are embedded in the 3D acoustic field. They are obtained by combining together in a modular way the components presented in the previous chapters. As such, the resulting numerical schemes possess a numerical energy that can be used to guarantee their stability. For each of these virtual instruments, a description of the physical model is given, together with the implementation details and simulation results. The implementation of these instruments as energy conserving algorithms is original.

Appendix A - Experimental investigation of nonlinearities in drum membranes

In this short appendix, a preliminary investigation on nonlinearities in drum membranes is presented. This work has been conducted in collaboration with M. Newton at the University of Edinburgh, and has been the subject of a recent conference article. Some data not previously published are considered here, as well.

Appendix B - Finite precision effects in FDTD schemes

This appendix presents some preliminary considerations regarding the role of round off errors in finite difference time domain schemes. Different implementation strategies are compared for the simulation of simple systems, such as the harmonic oscillator. Heuristic recipes for the reduction of rounding errors are presented.

Chapter 2

Linear plate vibration

The behaviour of gongs and cymbals falls within the realm of thin stiff structures. It is thus natural to begin the study of these objects with an examination of linear plate vibration. Plates are flat structures, with intrinsic stiffness, and with a thickness H much smaller than the lateral dimensions, such that a two dimensional model is sufficient to describe their behaviour. The first works in plate vibration theory can be traced back as far as the 18th century with Euler [79]. Needless to say, the literature on the subject is vast. A historic survey of the matter will not be attempted here, but the interested reader is referred to [196, 4].

It is useful, now, to introduce some terminology and to discuss some preliminary points. A first distinction can be made between "thin" and "thick" models. Depending on the ratio between the thickness and the typical dimension of the plate, in fact, different assumptions can be adopted. Thin models were presented in their final form by Kirchhoff [114] and Love [129], using underlying hypotheses similar to those of the Euler-Bernoulli bending theory of beams [91]. Dropping some of these assumptions led Reissner [145] and Mindlin [131] to a more general (thick) model. In this work, only the thin models will be discussed.

In stiff objects, longitudinal and transverse wave propagation coexist, the former generated by in-plane vibration of the plate, the latter by the motion perpendicular to the plate itself. In musical acoustics, one is generally interested in the transverse vibrations of plates, as it is this motion that most strongly interacts with the air. In thin models, the transverse motion decouples from the longitudinal vibrations, and these are usually neglected. This is not the case, however, for more general models, like that of Mindlin and Reissner mentioned above. Longitudinal vibrations are also to be taken into account when nonlinearities are considered, see Chapter 3.

Limiting the attention to the transverse vibration of plates for the moment, different regimes can be found depending on the relative size of the displacement w compared to the thickness H, where w is a measure of the maximum amplitude of vibration. When w is small ($w \ll H$), simplifying assumptions can be adopted in order to reduce the complexity of the equations, which become linear in w. This case will be dealt with in this chapter. When the displacement is comparable to the thickness, the resulting equations in w are nonlinear [136]. This will be the subject of the next chapter.

The behaviour of real gongs and cymbals departs significantly from that of the simple models described above in that they often incorporate irregularities or variations in their geometry (see [174] and references therein for a short but detailed survey.) Crash cymbals, for example, are better approximated by a spherical cap, and they often have a small dome at the centre. Gongs and cymbals sometimes exhibit a decreasing thickness along their radius, from the centre towards the rim. Local defects due to the manufacturing process are also quite common, and can have a strong impact on the modal analysis of the structure [43]. It is clear, then, that flat homogeneous structures cannot reproduce these features. A thorough analysis of these instruments would require the study of shells [125, 180], or curved plates, which can become very involved from a simulation point of view. This problem will not be touched upon in this work.

The numerical simulation of plate-like structures is of pivotal importance in many branches of engineering, especially in the design of vehicles and buildings. In this context, the most adopted approach is the finite element method [55, 210]. In musical acoustics, simulations of plates have been performed to model soundboards in guitars [77, 68], pianos [89, 46], violins [107], as well as gong-like structures for sound synthesis [118, 73] and reverberation [5, 16].

In this chapter and in the next, only thin models of flat homogeneous plates will be discussed, and it will be clear that these are able nonetheless to reproduce the main features of gongs and cymbals. The underlying theoretical models will be presented first, while numerical simulation and results will be discussed in the following section.

2.1 Preliminaries I: Notation

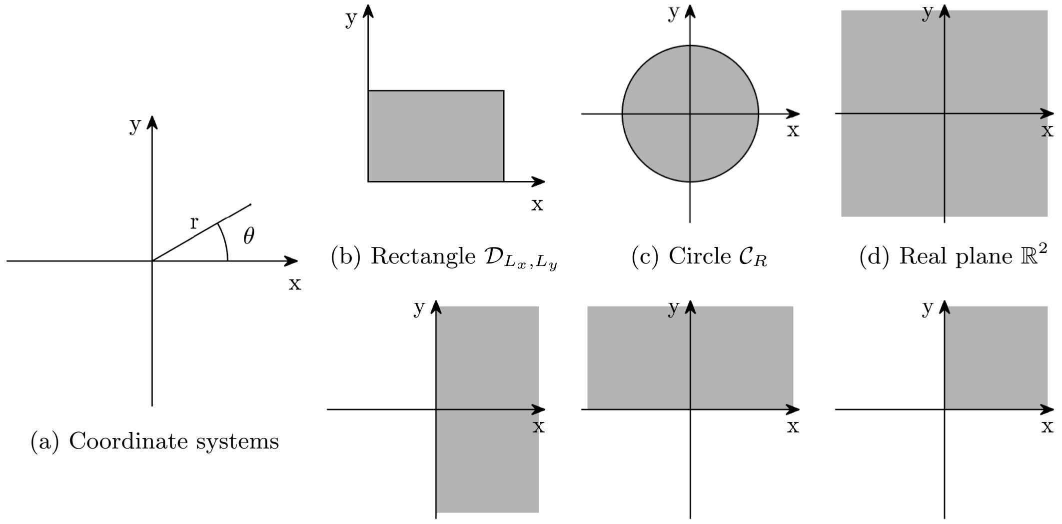
This preliminary section provides a short mathematical foundation for the systems described in the first chapters of this work.

2.1.1 Domains and coordinate systems

Consider a 2D region $\mathcal{D} \subseteq \mathbb{R}^2$, subset of the real plane $\mathbb{R}^2$, endowed with a set of orthonormal coordinates. In continuous physical models, it is customary to adopt either Cartesian (x, y) or polar (r, θ) coordinates, depending on the geometry of the domain. From a finite difference simulation point of view, however, Cartesian coordinates are generally preferred as they lead to algorithmic simplicity, regardless of the particular geometry of the plate. In acoustical applications, typical domains of interest are rectangles $\mathcal{D}_{L_x,L_y}$, with sides L_x and L_y, or circles of radius R. Other useful domains for general analysis purposes are the real plane $\mathbb{R}^2$, the half planes $\mathbb{H}^x$ and $\mathbb{H}^y$, the quarter plane $\mathbb{Q}$ (not to be confused with the mathematical set of rational numbers). The mathematical definition of these domains is the following:

- Rectangle: $\mathcal{D}_{L_x,L_y} = \left\{(x, y) \in \mathbb{R}^2,\ 0 \le x \le L_x,\ 0 \le y \le L_y\right\}$
- Circle: $\mathcal{C}_R = \{(r, \theta) \in \mathbb{R}^2,\ 0 \le r \le R,\ 0 \le \theta < 2\pi\}$
- Real plane: $\mathbb{R}^2$
- Half plane, x-direction: $\mathbb{H}^x = \{(x, y) \in \mathbb{R}^2,\ x \ge 0\}$
- Half plane, y-direction: $\mathbb{H}^y = \{(x, y) \in \mathbb{R}^2,\ y \ge 0\}$
- Quarter plane: $\mathbb{Q} = \{(x, y) \in \mathbb{R}^2,\ x \ge 0,\ y \ge 0\}$.

These are represented in Figure 2.1.



(a) Coordinate systems

(b) Rectangle $\mathcal{D}_{L_x,L_y}$ (c) Circle $\mathcal{C}_R$ (d) Real plane $\mathbb{R}^2$

(e) Half x-plane $\mathbb{H}^x$ (f) Half y-plane $\mathbb{H}^y$ (g) Quarter plane $\mathbb{Q}$

Figure 2.1: Coordinate systems and domains of interest in the 2D case.

2.1.2 Scalar, vector and tensor fields

Scalar fields, or scalar functions, over $\mathcal{D} \subseteq \mathbb{R}^2$ are maps $f : \mathcal{D} \rightarrow \mathbb{R}$ that associate a real number to each point of the domain. In this work, they will be denoted by lower case letters.

A 2D vector field over $\mathcal{D}$ is a map $\mathbf{v} : \mathcal{D} \rightarrow \mathbb{R}^2$, where each point of the domain is related to a 2D vector, that is an object in the Euclidean space $\mathbb{R}^2$ that has both a magnitude and a direction. Given two vectors $\mathbf{v}$ and $\mathbf{u}$, and denoting by θ the angle between them, it is possible to define the inner (scalar) product as

$$\mathbf{v} \cdot \mathbf{u} = |\mathbf{v}|\,|\mathbf{u}| \cos\theta, \tag{2.1}$$

where $|\cdot|$ denotes the magnitude of the vector. It follows from this definition that the Euclidean length of the vector is given by

$$|\mathbf{v}| = \sqrt{\mathbf{v} \cdot \mathbf{v}}. \tag{2.2}$$

Let $\{\mathbf{e}_1, \mathbf{e}_2\}$ be the standard orthonormal basis of $\mathbb{R}^2$. With a slight abuse of notation, it is possible to identify the vector with its components,

$$\mathbf{v} = (v_1, v_2)^T, \qquad v_j = \mathbf{v} \cdot \mathbf{e}_j \text{ for } j = 1, 2, \tag{2.3}$$

where the superscript ^{T} denotes the transposition operation. The scalar product (2.1) can thus be expressed as

$$\mathbf{v} \cdot \mathbf{u} = v_1\, u_1 + v_2\, u_2. \tag{2.4}$$

Another important operation involving vectors is the outer (tensor) product. Given $\mathbf{v}$ and $\mathbf{u}$, the result $\mathbf{T}$ of this operation is defined as

$$\mathbf{T} = \mathbf{v} \otimes \mathbf{u}, \quad \text{with } T_{ij} = v_i\, u_j, \quad i, j = 1, 2. \tag{2.5}$$

T is a rank 2 tensor, that is an object in $\mathbb{R}^4$, and it can be expressed as a 2×2 matrix

$$\mathbf{T} = \begin{pmatrix} v_1 u_1 & v_1 u_2 \\ v_2 u_1 & v_2 u_2 \end{pmatrix}. \tag{2.6}$$

Tensors will be denoted here by bold upper case Latin letters. A basis for the space $\mathbb{R}^4$ is given by the four so-called dyads $\mathbf{e}_i \otimes \mathbf{e}_j$. As such, any tensor **T** can be written as

$$\mathbf{T} = \sum_{i,j=1,2} T_{ij}\, \mathbf{e}_i \otimes \mathbf{e}_j. \tag{2.7}$$

The product between a vector **t** and the tensor (2.6) can be defined using the dyadic notation as [123]

$$\mathbf{t} \cdot (\mathbf{v} \otimes \mathbf{u}) = (\mathbf{t} \cdot \mathbf{v})\mathbf{u}, \qquad (\mathbf{v} \otimes \mathbf{u}) \cdot \mathbf{t} = (\mathbf{u} \cdot \mathbf{t})\mathbf{v}. \tag{2.8}$$

It is evident that left- and right-multiplying a vector by a tensor generally yields different results. A special tensor is the identity $\mathbb{1}$, defined by requiring that, for any vector **v**,

$$\mathbb{1} \cdot \mathbf{v} = \mathbf{v} \cdot \mathbb{1} = \mathbf{v}. \tag{2.9}$$

Another useful operation involving tensors is the doubly contracted product, generalising the scalar product for vectors, which is defined for any two tensors **T** and **U** as

$$\mathbf{T} : \mathbf{U} = T_{11}U_{11} + T_{12}U_{12} + T_{21}U_{21} + T_{22}U_{22}. \tag{2.10}$$

Associated with this product is the Frobenius norm defined as

$$|\mathbf{T}| = \sqrt{\mathbf{T} : \mathbf{T}}. \tag{2.11}$$

This notation will be used in order to distinguish this tensor norm from the L_2 norm defined in (2.18). An equivalent definition is given by

$$|\mathbf{T}| = \sqrt{\mathrm{tr}(\mathbf{T}^T\mathbf{T})}, \tag{2.12}$$

where $\mathrm{tr}(\,\cdot\,)$ represents the trace operator. The trace of a tensor is defined as the sum of the diagonal terms,

$$\mathrm{tr}(\mathbf{T}) = \mathbb{1} : \mathbf{T}. \tag{2.13}$$

A tensor field over $\mathcal{D}$ assigns a tensor to each point of the domain.

2.1.3 Differential operators and PDEs

The physical description of the systems presented in this work is given in terms of partial differential equations (PDEs). These involve partial differentiation with respect to time, in the form $\partial/\partial t$ and its powers, and space. In the Cartesian case, the spatial partial differential operators are written as $\partial/\partial x$ and $\partial/\partial y$, or combinations of the previous. In this work, these operators will be denoted simply with ∂_t, ∂_x and ∂_y, respectively.

Two combinations of the spatial derivatives will play a crucial role in the remainder of this work. The Laplacian and the biharmonic operators, indicated with Δ and Δ^2, respectively, are defined in 2D as follows:

$$\Delta = \partial_x^2 + \partial_y^2, \qquad \Delta^2 = \Delta\Delta = \partial_x^4 + 2\partial_x^2\partial_y^2 + \partial_y^4. \tag{2.14}$$

Additional operators will sometimes appear, namely the gradient $\boldsymbol{\nabla}$ and the divergence $\nabla\cdot$. When applied to a scalar function $w(x,y)$ and to a vector field $\mathbf{v}(x,y)$ they give

$$\boldsymbol{\nabla} w = (\partial_x w, \partial_y w)^T, \qquad \nabla \cdot \mathbf{v} = \partial_x v_1 + \partial_y v_2. \tag{2.15}$$

The gradient of a vector field and the divergence of a tensor field will also be useful, and they are defined as

$$\boldsymbol{\nabla}\mathbf{v} = \begin{bmatrix} \partial_x v_1 & \partial_x v_2 \\ \partial_y v_1 & \partial_y v_2 \end{bmatrix}, \qquad \nabla \cdot \mathbf{T} = \begin{bmatrix} \partial_x T_{11} + \partial_y T_{21} \\ \partial_x T_{12} + \partial_y T_{22} \end{bmatrix}. \tag{2.16}$$

When written in terms of these operators, the physical equations remain the same under a change of coordinate system.

Identities

Some useful identities are given here, without proof, for scalar, vector and tensor fields introduced Section 2.1.2:

$$\nabla \cdot (f\mathbf{v}) = f\nabla \cdot \mathbf{v} + \mathbf{v} \cdot \boldsymbol{\nabla} f \tag{2.17a}$$

$$\nabla \cdot (\mathbf{T}\mathbf{v}) = \mathbf{T} : \boldsymbol{\nabla}\mathbf{v} + (\nabla \cdot \mathbf{T}^T) \cdot \mathbf{v}. \tag{2.17b}$$

2.1.4 Integral relations

The definition of energy plays an important role in plate theory, as in all physical systems. Energy analysis of 2D systems relies on an inner product, a bilinear mapping that, given two continuous functions f and g over a domain $\mathcal{D} \subseteq \mathbb{R}^2$, is defined as

$$\langle f, g \rangle_{\mathcal{D}} = \int_{\mathcal{D}} f\, g \,\mathrm{d}\sigma, \tag{2.18}$$

where $\mathrm{d}\sigma = \mathrm{d}x\,\mathrm{d}y$ denotes the differential area element. The more compact notation on the left will be used to simplify the energy expressions in the various cases, while calculations will be performed using the more standard notation on the right. Associated with this inner product is the L_2 norm $\|f\|_{\mathcal{D}}$ of a function f, defined as

$$\|f\|_{\mathcal{D}} = \sqrt{\langle f, f \rangle_{\mathcal{D}}}. \tag{2.19}$$

A key tool in energy manipulations is integration by parts, and for functions f and g vanishing at infinity one can write

$$\int_{\mathbb{R}^2} (\partial_x f) g \, \mathrm{d}\sigma = -\int_{\mathbb{R}^2} f \partial_x g \, \mathrm{d}\sigma, \qquad \int_{\mathbb{R}^2} (\partial_y f) g \, \mathrm{d}\sigma = -\int_{\mathbb{R}^2} f \partial_y g \, \mathrm{d}\sigma. \tag{2.20}$$

Similar relations hold for the second derivatives, leading to the following equation involving the Laplacian:

$$\int_{\mathbb{R}^2} (\Delta f) \, g \, \mathrm{d}\sigma = \int_{\mathbb{R}^2} f \Delta g \, \mathrm{d}\sigma. \tag{2.21}$$

When the domain is finite, Green's theorem can be used to transform a surface integral into a line integral of a flux, in the following way:

$$\int_{\mathcal{D}} \nabla \cdot \mathbf{v} \, \mathrm{d}\sigma = \oint_{\partial \mathcal{D}} \mathbf{n} \cdot \mathbf{v} \, \mathrm{d}s, \tag{2.22}$$

for any vector function $\mathbf{v}$ defined over a domain $\mathcal{D}$, where $\mathbf{n}$ is the outward unit vector normal to the boundary. When combined with (2.17a) and (2.17b), one obtains

$$\int_{\mathcal{D}} f \nabla \cdot \mathbf{v} \, \mathrm{d}\sigma = -\int_{\mathcal{D}} \mathbf{v} \cdot \boldsymbol{\nabla} f \, \mathrm{d}\sigma + \oint_{\partial \mathcal{D}} \mathbf{n} \cdot (f \mathbf{v}) \, \mathrm{d}s \tag{2.23a}$$

$$\int_{\mathcal{D}} \mathbf{v} \cdot (\nabla \cdot \mathbf{S}) \, \mathrm{d}\sigma = -\int_{\mathcal{D}} \mathbf{S} : \boldsymbol{\nabla} \mathbf{v} \, \mathrm{d}\sigma + \oint_{\partial \mathcal{D}} \mathbf{n} \cdot (\mathbf{S} \mathbf{v}) \, \mathrm{d}s \tag{2.23b}$$

for $\mathbf{S}$ a symmetric tensor.

Another useful identity in energy manipulations which involves time operators is the following:

$$\frac{\mathrm{d}f}{\mathrm{d}t} f = \frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{1}{2} f^2 \right), \tag{2.24}$$

which holds for any function $f(t)$. For any function $g(x, y, t)$, the previous equation can be extended to the integral identity

$$\int_{\mathcal{D}} g \, \partial_t g \, \mathrm{d}\sigma = \frac{\mathrm{d}}{\mathrm{d}t} \int_{\mathcal{D}} \frac{1}{2} g^2 \, \mathrm{d}\sigma. \tag{2.25}$$

2.1.5 A useful tensor mapping

Consider the tensor mapping $\phi_\nu : \mathbb{R}^4 \to \mathbb{R}^4$ defined as [173]:

$$\mathbf{T} \to \phi_\nu(\mathbf{T}) = (1 - \nu)\mathbf{T} + \nu \, \mathrm{tr}(\mathbf{T}) \, \mathbb{1}, \tag{2.26}$$

with $0 < \nu < 1$, $\mathbb{1}$ the identity tensor and tr the trace operator defined in (2.13). This will be useful in the derivation of the equations of motion for a plate in the next section. ϕ_ν is invertible, as can be seen with a short calculation, with

$$\phi_\nu^{-1}(\mathbf{T}) = \frac{1}{1 - \nu^2} \left[(1 + \nu)\mathbf{T} - \nu \, \mathrm{tr}(\mathbf{T}) \, \mathbb{1} \right]. \tag{2.27}$$

When doubly contracted with $\mathbf{T}$, this gives

$$\mathbf{T} : \phi_\nu^{-1}(\mathbf{T}) = \frac{1}{1-\nu^2}\left[(1+\nu)\left|\mathbf{T}\right|^2 - \nu(\mathrm{tr}(\mathbf{T}))^2\right], \tag{2.28}$$

where the norm of a tensor has been defined in (2.11). Using the identity

$$\mathbf{T} : \phi_\nu^{-1}(\mathbf{U}) = \mathbf{U} : \phi_\nu^{-1}(\mathbf{T}), \tag{2.29}$$

one can write

$$\partial_t \mathbf{T} : \phi_\nu^{-1}(\mathbf{T}) = \frac{1}{2}\partial_t\left(\mathbf{T} : \phi_\nu^{-1}(\mathbf{T})\right). \tag{2.30}$$

When integrated over the domain $\mathcal{D}$, this becomes

$$\int_{\mathcal{D}} \partial_t \mathbf{T} : \phi_\nu^{-1}(\mathbf{T})\,\mathrm{d}\sigma = \frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{D}} \frac{1}{2}\mathbf{T} : \phi_\nu^{-1}(\mathbf{T})\,\mathrm{d}\sigma. \tag{2.31}$$

Similar relations can be written when ϕ_ν^{-1} is replaced with ϕ_ν.

Finally, taking the trace of ϕ_ν yields

$$\mathrm{tr}(\phi_\nu(\mathbf{T})) = (1-\nu)\,\mathrm{tr}(\mathbf{T}) + 2\nu\,\mathrm{tr}(\mathbf{T}) \tag{2.32a}$$

$$= (1+\nu)\,\mathrm{tr}(\mathbf{T}). \tag{2.32b}$$

2.1.6 Frequency domain analysis

It is sometimes useful to analyse a particular PDE in the frequency and wavevector domain. When the domain is infinite, this can be done by means of a Fourier transform. An equivalent approach in Cartesian coordinates consists in inserting in the PDE the following *ansatz* [92]

$$w(x,y,t) = \mathrm{e}^{\mathrm{i}\omega t + \mathrm{i}\beta_x x + \mathrm{i}\beta_y y}. \tag{2.33}$$

This corresponds to a wave with temporal frequency $\omega/2\pi$ and spatial wavevector $\boldsymbol{\beta} = (\beta_x, \beta_y)^T$. The action of the partial differential operators on this test function can thus be rewritten in terms of simple multiplications:

$$\begin{gathered}\partial_t w = \mathrm{i}\omega w, \quad \partial_x w = \mathrm{i}\beta_x w, \quad \partial_y w = \mathrm{i}\beta_y w, \\ \partial_t^2 w = -\omega^2 w, \quad \partial_x^2 w = -\beta_x^2 w, \quad \partial_y^2 w = -\beta_y^2 w, \quad \partial_x\partial_y w = -\beta_x\beta_y w.\end{gathered} \tag{2.34}$$

For the Laplacian and biharmonic operators (2.14), one obtains

$$\Delta w = -\left|\boldsymbol{\beta}\right|^2 w, \qquad \Delta^2 w = \left|\boldsymbol{\beta}\right|^4 w. \tag{2.35}$$

2.2 Thin plate model

2.2.1 Equations of motion

Consider a two dimensional region $\mathcal{D}$, with $\mathcal{D} \subset \mathbb{R}^2$, delimited by a smooth boundary $\partial\mathcal{D}$, and let it define the mid-surface of the plate in its reference configuration. The 3D volume Γ occupied by the plate can be expressed as $\Gamma = \mathcal{D} \times [-H/2\,, +H/2]$, where H is the thickness of the plate itself. Then, let $(\mathbf{e}_1, \mathbf{e}_2)$ be an orthonormal basis of $\mathcal{D}$, and let $\mathbf{x} \in \mathcal{D}$ denote the position of a point in this coordinate system. A third unit vector $\mathbf{e}_z$ can be introduced, which is normal to the surface at all points, such that $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_z)$ be an orthonormal basis of $\mathbb{R}^3$. Let $\mathbf{u}(\mathbf{x}, t) = (u_1(\mathbf{x}, t), u_2(\mathbf{x}, t))^T$ and $w(\mathbf{x}, t)$ be the in-plane and transverse displacements of the plate at position $\mathbf{x}$ and time $t \in \mathbb{R}^+$.

The classic theory of linear thin plate vibration is based on Kirchhoff and Love hypotheses [114, 129]:

1. Any normal to the plate mid-surface before deformation remains normal to the deformed mid-surface.
2. Transverse stresses and transverse shear stresses are neglected.
3. The material is linear (the relation between strains and stresses in the material is of Hooke's type), homogeneous and isotropic.
4. Rotatory inertia is neglected.

Under these assumptions, the second rank tensor $\mathbf{E}$ representing the plane part of the Green-Lagrange strain tensor can be written as

$$\mathbf{E} = \boldsymbol{\epsilon} + z\boldsymbol{\kappa}, \tag{2.36}$$

where the membrane strain tensor $\boldsymbol{\epsilon}$ and the curvature tensor $\boldsymbol{\kappa}$ are defined as

$$\boldsymbol{\epsilon} = \frac{1}{2}\left(\boldsymbol{\nabla}\mathbf{u} + (\boldsymbol{\nabla}\mathbf{u})^T\right), \qquad \boldsymbol{\kappa} = -\boldsymbol{\nabla}\boldsymbol{\nabla} w. \tag{2.37}$$

Using hypothesis 3, for a material of Young's modulus E (in Pa) and dimensionless Poisson's ratio ν $(0 < \nu < 1/2)$, the relationship between the strain tensor $\mathbf{E}$ and the stress tensor $\mathbf{S}$ can be written as

$$\mathbf{S} = \frac{E}{1-\nu^2}\,\phi_\nu(\mathbf{E}), \tag{2.38}$$

where ϕ_ν is the linear tensor mapping defined in (2.26). The bending tensor $\mathbf{M}$ can be obtained by integrating across the thickness of the plate,

$$\mathbf{M} = \int_{-H/2}^{H/2} z\mathbf{S}\,\mathrm{d}z = D\,\phi_\nu(\boldsymbol{\kappa}), \tag{2.39}$$

with bending stiffness D defined as

$$D = \frac{EH^3}{12(1-\nu^2)}. \tag{2.40}$$

Note that the contribution from $\boldsymbol{\epsilon}$ has disappeared.

Let now $\mathbf{q} = (q_1, q_2)^T$ be the vector of the shear forces along the principal axis. Two additional equations can be derived by requiring the equilibrium of the forces along the vertical axis and of the moments along the two principal directions [196, 91]. Finally, the Kirchhoff equations for thin plate vibration can be written in condensed form as

$$\rho H\, \partial_t^2 w = \nabla \cdot \mathbf{q} \quad (2.41a) \qquad \mathbf{q} = \nabla \cdot \mathbf{M} \quad (2.41b) \qquad \mathbf{M} = D\phi_\nu(\boldsymbol{\kappa}), \quad (2.41c)$$

where ρ is the surface density of the plate (in kg/m^2). In the literature [196], (2.41a) is referred to as the kinematic equation, (2.41b) is the equilibrium balance, and (2.41c) is the constitutive equation, relating the moments with the plate curvatures. It can be inferred from (2.41c) and from the definition of the tensor mapping ϕ_ν (2.26) that the tensor $\mathbf{M}$ is symmetric.

In a simulation setting, it is useful to rewrite this system in a first order form, as in the case of the so-called Yee's FDTD scheme [208, 172]. By introducing the transverse velocity $v = \partial_t w$ and the vector of angular velocities $\boldsymbol{\omega}$, the system (2.41) then becomes

$$\rho H\, \partial_t v = \nabla \cdot \mathbf{q} \tag{2.42a}$$

$$\mathbf{q} = \nabla \cdot \mathbf{M} \tag{2.42b}$$

$$\partial_t \mathbf{M} = -D\phi_\nu(\boldsymbol{\nabla}\boldsymbol{\omega}) \tag{2.42c}$$

$$\boldsymbol{\omega} = \boldsymbol{\nabla} v \tag{2.42d}$$

At the boundary of the region $\mathcal{D}$, appropriate conditions must be supplied. One possible way of extracting suitable boundary conditions is by means of energy methods, see Section 2.2.2.

Fourth order equation

The system of equations presented before can be combined into a single, fourth order equation in w, which reads

$$\partial_t^2 w = -\kappa^2 \Delta\Delta w = -\kappa^2 \Delta^2 w, \tag{2.43}$$

with $\kappa = \sqrt{D/\rho H}$; Δ^2 is the biharmonic operator defined in (2.14).

Dispersion relation

The dispersion relation for the plate equation can be obtained by inserting (2.33) into (2.43), together with the phase velocity v_ϕ and the group velocity v_g

$$\omega = \pm\kappa|\boldsymbol{\beta}|^2 \quad \Longrightarrow \quad v_\phi = \frac{\omega}{|\boldsymbol{\beta}|} = \kappa|\boldsymbol{\beta}|\,, \quad v_g = \frac{\mathrm{d}\omega}{\mathrm{d}|\boldsymbol{\beta}|} = 2\kappa|\boldsymbol{\beta}|\,, \tag{2.44}$$

where ω is the angular velocity and $\boldsymbol{\beta}$ is the wavevector (see Section 2.1.6). The system is thus dispersive, with high frequency waves travelling faster than low frequency ones. As a consequence of this hypothesis, this model predicts infinite group velocity in the limit of high frequencies, a result which is obviously non-physical. In order to correct this behaviour, different models must be adopted, like that of Mindlin and Reissner mentioned earlier [131, 145]. The dispersion relation in this case is [179]

$$\frac{\rho^2 H^4}{12}\omega^4 + \left(\rho H D|\boldsymbol{\beta}|^2 + \frac{\rho H^4}{12}Gc_f|\boldsymbol{\beta}|^2 + \rho H^2 G c_f\right)\omega^2 + Gc_f H D|\boldsymbol{\beta}|^4 = 0, \tag{2.45}$$

where all the constants have been defined before, except for the shear modulus $G = E/2(1+\nu)$ and the shear correction factor $c_f \approx 5/6$. It is now possible to solve this fourth order equation in terms of ω^2, from which two different branches appear, well separated in the frequency range, see Figure 2.2. The lower one is of interest in the context of musical acoustics, and is plotted in Figure 2.3 as a function of $|\boldsymbol{\beta}|$ for steel plates of different thickness. The Mindlin-Reissner dispersion relation is compared with the Kirchhoff relation (2.44), and a deviation in cents between the two models is given. It is clear that for small values of H, the two models have a similar behaviour, with a maximum deviation in cents over the human hearing range that does not exceed 30 cents. Different is the case of a 1 cm thick metal plate, for which the difference between the two models is more pronounced. As the computation effort required by the Mindlin-Reissner model far overcomes the benefits when very thin plates are involved, only the Kirchhoff model will be discussed here. Notice, however, that the results of Figure 2.3 are strictly limited to metal plates. Other materials, in fact, have different physical parameters that can significantly limit the validity of this choice.

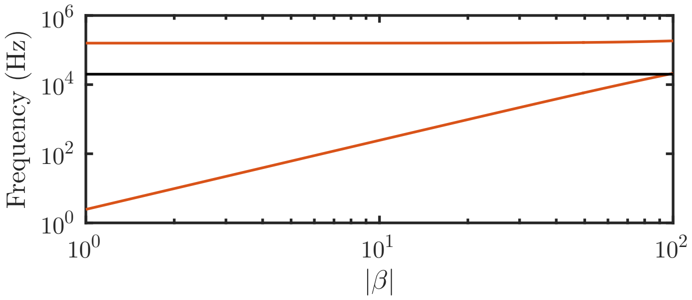


Figure 2.2: Dispersion relation for the Mindlin-Reissner model for a 1 cm thick steel plate. Two different branches are present, one of which lies above the range of human hearing (marked with a black horizontal line at 20 kHz).

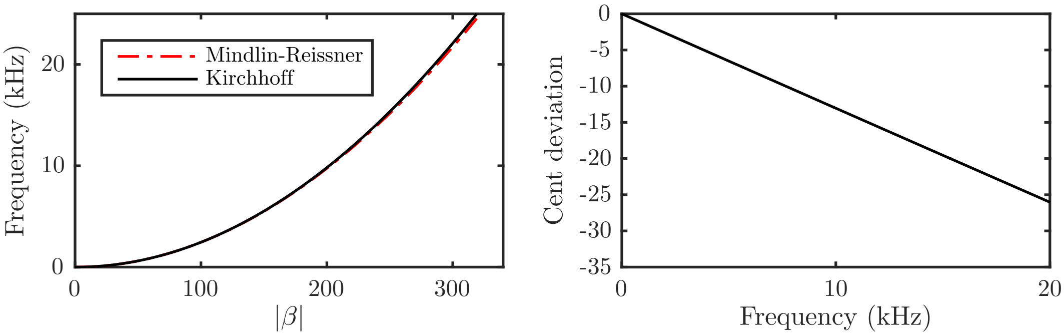
Initial conditions

The solution of the plate equations described so far requires the specification of initial conditions. The displacement and velocity of every point of the plate at $t = 0$ must be supplied. A spatial distribution that will be often used in this work is the so-called raised cosine [19]. It is specified by a width r and an amplitude A, and it is centred at some point (x_0, y_0) of the plate. First, consider the function

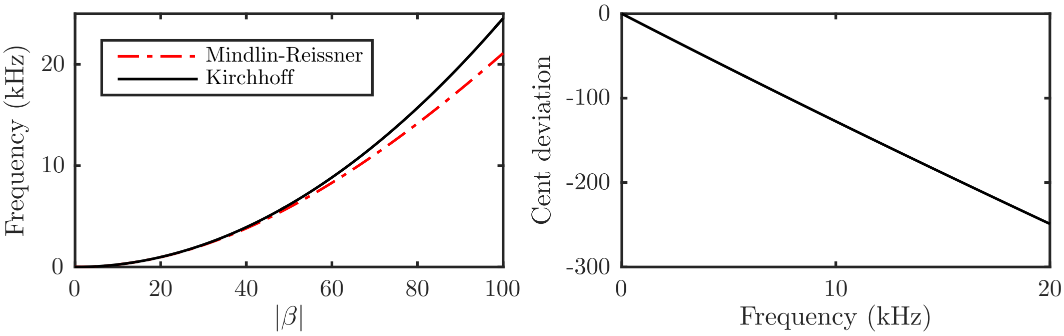
$$o(x,y) = \sqrt{(x-x_0)^2 + (y-y_0)^2}, \tag{2.46}$$

which represents the distance from (x_0, y_0). The raised cosine distribution $g(x,y)$ can thus be defined as

$$g(x,y) = \begin{cases} \frac{A}{2}\left[1 + \cos(\pi\, o(x,y)/r)\right] & \text{for } o(x,y) < r \\ 0 & \text{otherwise} \end{cases} \tag{2.47}$$



(a) $H = 1$ mm



(b) $H = 1$ cm

Figure 2.3: Dispersion relations for the Kirchhoff and Mindlin-Reissner plate models for a steel plate with thickness as indicated and $\rho = 7800$ kg/m^3, $E = 200$ GPa and $\nu = 0.33$. The variation in cents between the two models across the range of human hearing is also plotted. For a 1 mm thick plate, the two curves are very similar, but more significant differences can be noted for thicker plates.

Such a function is represented in Figure 2.4, with parameters as indicated in the caption, and can be used to initialise both the displacement and the velocity of the plate. More realistic models require the inclusion of a striking mechanism like a mallet, as discussed in Chapter 4.

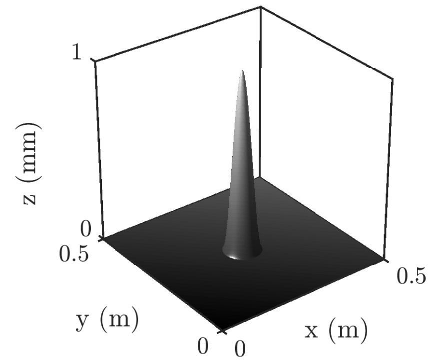


Figure 2.4: Raised cosine distribution positioned at the centre of a square plate with $L_x = L_y = 0.5$ m, amplitude $A = 1$ mm and width $r = 0.05$ m.

2.2.2 Energy analysis and boundary conditions

It is possible now to derive an energy balance for the system (2.42). Starting from (2.42a), multiply it by v and integrate over the domain $\mathcal{D}$:

$$
\begin{aligned}
\rho H \int_{\mathcal{D}} v\, \partial_t v \,\mathrm{d}\sigma = {} & \int_{\mathcal{D}} v\, \nabla \cdot \mathbf{q} \,\mathrm{d}\sigma \\
\overset{(2.23\text{a})}{=} {} & -\int_{\mathcal{D}} \boldsymbol{\nabla} v \cdot \mathbf{q} \,\mathrm{d}\sigma + \oint_{\partial\mathcal{D}} \mathbf{n} \cdot (v\mathbf{q}) \,\mathrm{d}s \\
\overset{\substack{(2.42\text{b}),\\ (2.42\text{d})}}{=} {} & -\int_{\mathcal{D}} \boldsymbol{\omega} \cdot (\nabla \cdot \mathbf{M}) \,\mathrm{d}\sigma + \oint_{\partial\mathcal{D}} \mathbf{n} \cdot (v\mathbf{q}) \,\mathrm{d}s \\
\overset{(2.23\text{b})}{=} {} & +\int_{\mathcal{D}} \mathbf{M} : \boldsymbol{\nabla} \boldsymbol{\omega} \,\mathrm{d}\sigma - \oint_{\partial\mathcal{D}} \mathbf{n} \cdot (\mathbf{M}\boldsymbol{\omega}) \,\mathrm{d}s + \oint_{\partial\mathcal{D}} \mathbf{n} \cdot (v\mathbf{q}) \,\mathrm{d}s \\
\overset{\substack{(2.42\text{c}),\\ (2.27)}}{=} {} & -\frac{1}{D} \int_{\mathcal{D}} \mathbf{M} : \phi^{-1}(\partial_t \mathbf{M}) \,\mathrm{d}\sigma - \oint_{\partial\mathcal{D}} \mathbf{n} \cdot (\mathbf{M}\boldsymbol{\omega}) \,\mathrm{d}s + \oint_{\partial\mathcal{D}} \mathbf{n} \cdot (v\mathbf{q}) \,\mathrm{d}s \\
\overset{\substack{(2.30),\\ (2.28)}}{=} {} & -\frac{1}{D(1-\nu^2)} \frac{\mathrm{d}}{\mathrm{d}t} \int_{\mathcal{D}} \Big((1+\nu)\, |\mathbf{M}|^2 - \nu \left(\mathrm{tr}(\mathbf{M})\right)^2 \Big) \,\mathrm{d}\sigma + \mathfrak{B}_{\mathrm{pl}}.
\end{aligned}
\tag{2.48}
$$

In the last line, $\mathfrak{B}_{\mathrm{pl}}$ represents the boundary integrals along $\partial\mathcal{D}$,

$$
\mathfrak{B}_{\mathrm{pl}} = -\oint_{\partial\mathcal{D}} \mathbf{n} \cdot (\mathbf{M}\boldsymbol{\omega}) \,\mathrm{d}s + \oint_{\partial\mathcal{D}} \mathbf{n} \cdot (v\mathbf{q}) \,\mathrm{d}s, \tag{2.49}
$$

where $\mathbf{n}$ is a unit outer normal vector on $\partial\mathcal{D}$, and groups together the boundary terms for the plate. Making use of (2.24) to modify the left hand side of the equation, it is possible to rewrite (2.48) as an energy balance of the form

$$
\frac{\mathrm{d}\mathfrak{H}_{\mathrm{pl}}}{\mathrm{d}t} = \mathfrak{B}_{\mathrm{pl}}, \tag{2.50}
$$

with total energy $\mathfrak{H}_{\mathrm{pl}}$ defined as

$$
\mathfrak{H}_{\mathrm{pl}} = \frac{\rho}{2} \left\| v \right\|_{\mathcal{D}}^2 + \frac{1}{2D(1-\nu^2)} \left[(1+\nu) \left\| \mathbf{M} \right\|_{\mathcal{D}}^2 - \nu \left\| \mathrm{tr}\, \mathbf{M} \right\|_{\mathcal{D}}^2 \right]. \tag{2.51}
$$

The expression for $\mathfrak{H}_{\mathrm{pl}}$ corresponds to the energy found in the literature [180, 196].

When a finite domain is taken into account, energy analysis yields in a natural way the correct boundary conditions, as will be seen shortly. Every time integration by parts is used, in fact, a new term is produced which is resolved at the boundary. By requiring these terms to vanish identically, energy conservation can be guaranteed.

Boundary conditions

At any point of the boundary $\partial\mathcal{D}$, let s denote the arc-length and $(\mathbf{n}, \boldsymbol{\tau})$ be the normal and tangent unit vectors, respectively. Using (2.42d) and the fact that every vector $\mathbf{v}$ can be decomposed into an orthonormal basis as

$$
\mathbf{v} = \mathbf{n}(\mathbf{n} \cdot \mathbf{v}) + \boldsymbol{\tau}(\boldsymbol{\tau} \cdot \mathbf{v}), \tag{2.52}
$$

the boundary terms $\mathfrak{B}_{\mathrm{pl}}$ can be expanded into

$$\mathfrak{B}_{\mathrm{pl}} = -\oint_{\partial\mathcal{D}} \mathbf{n}\cdot\mathbf{M}\mathbf{n}(\mathbf{n}\cdot\boldsymbol{\nabla}v)\,\mathrm{d}s - \oint_{\partial\mathcal{D}} \mathbf{n}\cdot\mathbf{M}\boldsymbol{\tau}(\boldsymbol{\tau}\cdot\boldsymbol{\nabla}v)\,\mathrm{d}s + \oint_{\partial\mathcal{D}} \mathbf{n}\cdot(v\mathbf{q})\,\mathrm{d}s \tag{2.53a}$$

$$= -\oint_{\partial\mathcal{D}} M_{nn}\partial_n v\,\mathrm{d}s + \oint_{\partial\mathcal{D}} \partial_s M_{n\tau} v\,\mathrm{d}s + \oint_{\partial\mathcal{D}} v\,q_n\,\mathrm{d}s, \tag{2.53b}$$

with $M_{nn} = \mathbf{n}\cdot\mathbf{M}\mathbf{n}$, $M_{n\tau} = \mathbf{n}\cdot\mathbf{M}\boldsymbol{\tau}$ and $q_n = \mathbf{n}\cdot\mathbf{q}$. In the second line, integration by parts has been used to swap the tangential gradient, while no extra boundary term is produced for a smooth contour. Extra terms are produced, however, when the boundary is not smooth, as in the case of, e.g., a rectangular plate. In this case, in fact, corner conditions appear (see Section 2.2.4). Three sets of conditions are generally found in the literature, which allow $\mathfrak{B}_{\mathrm{pl}}$ to identically vanish.

a) Clamped conditions. The plate is held fixed at the boundary with vanishing slope.

$$v|_{\partial\mathcal{D}} = 0, \qquad \text{and} \qquad \partial_n v|_{\partial\mathcal{D}} = 0. \tag{2.54}$$

b) Simply supported conditions. The plate is held fixed at the boundary, but the edge is free to rotate. In this case, the bending moment normal to the boundary is required to be zero.

$$v|_{\partial\mathcal{D}} = 0, \qquad \text{and} \qquad M_{nn}|_{\partial\mathcal{D}} = 0. \tag{2.55}$$

c) Free conditions. The plate has no constraints applied at the ends.

$$M_{nn}|_{\partial\mathcal{D}} = 0, \qquad \text{and} \qquad q_n + \partial_s M_{n\tau}|_{\partial\mathcal{D}} = 0. \tag{2.56}$$

2.2.3 Losses and tension

Equation (2.43) can be modified in order to take into account losses and tension. Losses play an important role in physical systems, and have a crucial effect in determining the perceptual features of musical instruments. In plate vibration, an important contribution to this topic has been given by Lambourg and Chaigne [118, 53, 119] and, more recently, by Humbert *et al.* [106]. Three sources of losses are generally considered, namely thermal effects, viscoelasticity and air radiation, and their relative importance depends on the type of material considered.

Radiation effects can be modelled by considering the coupling of the plate with the surrounding air and by including this in the simulation. These effects will be discussed in Section 5.4. On the other hand, modelling thermal and viscoelastic effects in the time domain requires the introduction of fractional derivatives [53]. Even though numerical simulations involving these operators have been performed (see, e.g., [22] and references therein, but also [141] for recent advances), their use can become rapidly unwieldy in terms of memory requirements. In sound synthesis applications, one can adopt a different approach which is loosely based on physical and perceptual considerations, but is in turn very simple to implement numerically. In this case, Eq. (2.43) can be modified in the following way:

$$\partial_t^2 w = -\kappa^2\Delta^2 w - 2\sigma_0\partial_t w + 2\sigma_1\Delta\partial_t w, \tag{2.57}$$

where the last two terms account for frequency independent and frequency dependent damping, respectively. The parameters σ_0 and σ_1 that govern losses can be set in terms of decay times at some chosen frequencies (see [19]).

Energy analysis techniques can be applied to (2.57) in order to derive an energy balance. Multiplying (2.57) by $\partial_t w$ and integrating over the domain $\mathcal{D}$ gives

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{pl}}}{\mathrm{d}t} = \mathfrak{B}_{\mathrm{pl}} - 2\sigma_0 \rho H \int_{\mathcal{D}} \partial_t w \partial_t w \,\mathrm{d}\sigma + 2\sigma_1 \rho H \int_{\mathcal{D}} \partial_t w \partial_t \Delta w \,\mathrm{d}\sigma \tag{2.58a}$$

$$\overset{(2.23\mathrm{a})}{=} \mathfrak{B}_{\mathrm{pl}} - 2\sigma_0 \rho H \int_{\mathcal{D}} (\partial_t w)^2 \,\mathrm{d}\sigma - 2\sigma_1 \rho H \left(\int_{\mathcal{D}} |\partial_t \boldsymbol{\nabla} w)|^2 \,\mathrm{d}\sigma + \oint_{\partial\mathcal{D}} \partial_t w (\mathbf{n} \cdot \partial_t \boldsymbol{\nabla} w) \,\mathrm{d}s \right) \tag{2.58b}$$

$$= \mathfrak{B}_{\mathrm{pl}} - \mathfrak{Q} + \mathfrak{B}_{\mathrm{loss}} \tag{2.58c}$$

where $\mathfrak{H}_{\mathrm{pl}}$ is the total energy of the plate defined in (2.51), $\mathfrak{B}_{\mathrm{pl}}$ represents the boundary terms (2.49) and

$$\mathfrak{Q} = 2\sigma_0 \rho H \left\| \partial_t w \right\|_{\mathcal{D}}^2 + 2\sigma_1 \rho H \left\| \partial_t \boldsymbol{\nabla} w \right\|_{\mathcal{D}}^2 . \tag{2.59}$$

The loss term $\mathfrak{Q}$ is clearly non-negative, meaning that the total energy of the system $\mathfrak{H}_{\mathrm{pl}}$ is decreasing. The reason why the boundary terms $\mathfrak{B}_{\mathrm{pl}}$ have been reinstated is that an additional boundary term, $\mathfrak{B}_{\mathrm{loss}}$, is created by integration by parts from the loss term. It is clear that this new term disappears under clamped and simply supported conditions. This is not the case, however, for free conditions, which must be modified due to the presence of $\mathfrak{B}_{\mathrm{loss}}$. Remembering that $v = \partial_t w$, the conditions in (2.56) become

$$M_{nn} + 2\sigma_1 \rho H v|_{\partial\mathcal{D}} = 0 \quad \text{and} \quad q_n + \partial_s M_{n\tau}|_{\partial\mathcal{D}} = 0. \tag{2.60}$$

Assuming now that the boundary terms are zero, the energy balance (2.58) can be integrated over time between $t = 0$ and $t = T$ to give

$$\mathfrak{H}_{\mathrm{pl}}(T) + \int_0^T \mathfrak{Q} \,\mathrm{d}t = \mathfrak{H}_{\mathrm{pl}}(0). \tag{2.61}$$

In other words, the energy dissipated up to any time T summed to the residual energy returns the initial energy of the system. This rather obvious observation can be exploited in the discrete scheme to ensure energy conservation even in the lossy case.

Another extension of the thin plate equation is the inclusion of a tension applied at the rim. Plates with a very small thickness H compared to the lateral dimension L have generally a low flexural rigidity D, and thus require a tensile force along the middle surface in order to carry the loads they are subject to. These plates are generally referred to as membranes [196]. The equation of motion in this case can be written as

$$\rho H \partial_t^2 w = -D\Delta^2 w + T\Delta w, \tag{2.62}$$

where T is the tension (in N/m) and Δ is the Laplacian operator introduced in (2.14). This equation will be referred to as the stiff membrane equation in the remainder of this work [84]. When $D = 0$, (2.62) reduces to the 2D wave equation for an ideal membrane [135].

An energy balance for (2.62) can be easily arrived at, which has the form

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{sm}}}{\mathrm{d}t} = 0, \quad \mathfrak{H}_{\mathrm{sm}} = \mathfrak{H}_{\mathrm{pl}} + \frac{T}{2}\left\|\boldsymbol{\nabla} w\right\|^2_{\mathcal{D}}\,, \tag{2.63}$$

where $\mathfrak{H}_{\mathrm{sm}}$ represents the energy for the stiff membrane and $\mathfrak{H}_{\mathrm{pl}}$ is the energy for the plate (2.51). An additional boundary term, not shown in the equation, is also produced by the new term. However, as drum membranes are usually fixed at the rim, both fixed conditions introduced for the plate equation also satisfy the new boundary term, which automatically disappears. This would not be the case for a free membrane[1], which will not be discussed here.

Losses, tension and stiffness can be used together in the same equation, in order to describe a lossy stiff membrane. As all these terms are linear, they can be grouped under an operator $\mathcal{O}_m$ defined as

$$\mathcal{O}_m(T, D, \sigma_0, \sigma_1, \rho, H) = T\Delta - D\Delta^2 - 2\sigma_0\rho H\partial_t + 2\sigma_1\rho H\Delta\partial_t. \tag{2.64}$$

This operator represents all the possible combinations of the four terms, and will be extensively used in the following chapters in order to simplify the notation.

2.2.4 Rectangular plates

For a plate defined over a rectangular region $\mathcal{D}_{L_x,L_y}$ (see Section 2.1.1), it is convenient to write the equations of motions in Cartesian coordinates $(x, y) \in \mathcal{D}_{L_x,L_y}$. The moments tensor $\mathbf{M}$ can then be written as

$$\mathbf{M} = \begin{pmatrix} m_x & m_{xy} \\ m_{xy} & m_y \end{pmatrix}, \tag{2.65}$$

where m_x and m_y are the bending moments per unit length along the x and y axis, respectively, and m_{xy} is the twisting moment per unit length.

With this representation, and after having eliminated $\mathbf{q}$ and $\boldsymbol{\omega}$, the system (2.42) can be rewritten as

$$\rho H\,\partial_t v = \partial_x^2 m_x + \partial_y^2 m_y + 2\partial_x\partial_y m_{xy} \tag{2.66a}$$

$$\partial_t m_x = -D\left(\partial_x^2 v + \nu\partial_y^2 v\right) \tag{2.66b}$$

$$\partial_t m_y = -D\left(\partial_y^2 v + \nu\partial_x^2 v\right) \tag{2.66c}$$

$$\partial_t m_{xy} = -D\,(1-\nu)\partial_x\partial_y v. \tag{2.66d}$$

This form of the system lends itself to an easier implementation of the boundary conditions. Notice that (2.66b) and (2.66c) can be written in a more compact form as

$$\begin{bmatrix} \partial_t m_x \\ \partial_t m_y \end{bmatrix} = -D \begin{bmatrix} 1 & \nu \\ \nu & 1 \end{bmatrix} \begin{bmatrix} \partial_x^2 v \\ \partial_y^2 v \end{bmatrix}, \tag{2.67}$$

---

[1] Although difficult to realise in practice, this configuration is technically possible. Imagine, e.g., a trampoline surface, tensioned by ropes, whose boundaries are able to move freely in the transverse direction.

which can be easily inverted to obtain

$$\begin{bmatrix} \partial_x^2 v \\ \partial_y^2 v \end{bmatrix} = -\frac{1}{D(1-\nu^2)} \begin{bmatrix} 1 & -\nu \\ -\nu & 1 \end{bmatrix} \begin{bmatrix} \partial_t m_x \\ \partial_t m_y \end{bmatrix}. \tag{2.68}$$

Energy

Although the energy expression (2.51) derived in the general case is perfectly valid here, it is useful to redo the analysis in the Cartesian case, as the mathematical manipulations closely resemble those of the discrete case (see Section 2.4.1).

Consider an infinite domain first, such that boundaries need not be taken into account. In order to obtain an energy balance for the system, start by multiplying (2.66a) by v and integrate over $\mathbb{R}^2$:

$$\begin{aligned} \rho H \int_{\mathbb{R}^2} v\,\partial_t v \,\mathrm{d}\sigma &= \int_{\mathbb{R}^2} v \left(\partial_x^2 m_x + \partial_y^2 m_y + 2\partial_x\partial_y m_{xy}\right) \mathrm{d}\sigma && (2.69\text{a}) \\ &= \underbrace{\int_{\mathbb{R}^2} \left(\partial_x^2 v\, m_x + \partial_y^2 v\, m_y\right) \mathrm{d}\sigma}_{=\,\mathrm{d}\mathfrak{U}_1/\,\mathrm{d}t} + \underbrace{\int_{\mathbb{R}^2} 2\partial_x\partial_y v\, m_{xy}\,\mathrm{d}\sigma}_{=\,\mathrm{d}\mathfrak{U}_2/\,\mathrm{d}t}. && (2.69\text{b}) \end{aligned}$$

The two terms containing $\mathfrak{U}_1$ and $\mathfrak{U}_2$, which are the contributions to the total potential energy $\mathfrak{U}$, can be manipulated to give

$$\begin{aligned} \frac{\mathrm{d}\mathfrak{U}_1}{\mathrm{d}t} &\overset{(2.68)}{=} -\frac{1}{D(1-\nu^2)} \int_{\mathbb{R}^2} (\partial_t m_x - \nu\partial_t m_y) m_x + (\partial_t m_y - \nu\partial_t m_x) m_y \,\mathrm{d}\sigma && (2.70\text{a}) \\ &= -\frac{1}{D(1-\nu^2)} \int_{\mathbb{R}^2} \left(m_x\partial_t m_x + m_y\partial_t m_y - \nu m_x\partial_t m_y - \nu m_y \partial_t m_x\right) \mathrm{d}\sigma && (2.70\text{b}) \\ &= -\frac{1}{2D(1-\nu^2)} \frac{\mathrm{d}}{\mathrm{d}t} \int_{\mathbb{R}^2} \left(m_x^2 + m_y^2 - 2\nu m_x m_y\right) \mathrm{d}\sigma && (2.70\text{c}) \end{aligned}$$

and

$$\begin{aligned} \frac{\mathrm{d}\mathfrak{U}_2}{\mathrm{d}t} &\overset{(2.66\text{d})}{=} -\frac{2}{D(1-\nu)} \int_{\mathbb{R}^2} m_{xy}\partial_t m_{xy}\,\mathrm{d}\sigma && (2.71\text{a}) \\ &= -\frac{1}{D(1-\nu)} \frac{\mathrm{d}}{\mathrm{d}t} \int_{\mathbb{R}^2} m_{xy}^2\,\mathrm{d}\sigma. && (2.71\text{b}) \end{aligned}$$

By rearranging all the terms on the left hand side of the equation, and by using the definition for the L_2 norm (2.19), it is possible to write the energy balance of the system as

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{rpl}}}{\mathrm{d}t} = 0, \tag{2.72}$$

with total energy $\mathfrak{H}_{\mathrm{rpl}}$ for the rectangular plate defined as

$$\begin{aligned} \mathfrak{H}_{\mathrm{rpl}} = \frac{\rho H}{2} \|v\|_{\mathbb{R}^2}^2 + \\ \frac{1}{2D(1-\nu^2)}\Big(\|m_x\|_{\mathbb{R}^2}^2 + \|m_y\|_{\mathbb{R}^2}^2 - 2\nu\langle m_x, m_y\rangle_{\mathbb{R}^2}\Big) + \frac{1}{D(1-\nu)} \|m_{xy}\|_{\mathbb{R}^2}^2. \end{aligned} \tag{2.73}$$

When written in Cartesian coordinates, the energy $\mathfrak{H}_{\mathrm{pl}}$ derived in (2.51) is equivalent to $\mathfrak{H}_{\mathrm{rpl}}$.

Finite domains and boundary terms

In order to analyse boundary conditions, consider now the quarter plane $\mathbb{Q}$ defined in Section 2.1.1. The calculation is similar to the previous one, but this time the energy balance is

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{rpl}}}{\mathrm{d}t} = \mathfrak{B}_{\mathrm{rpl}}, \quad \mathfrak{B}_{\mathrm{rpl}} = \mathfrak{B}_1 + \mathfrak{B}_2. \tag{2.74}$$

As a matter of fact, performing integration to obtain $\mathrm{d}\mathfrak{U}_1/\mathrm{d}t$ and $\mathrm{d}\mathfrak{U}_2/\mathrm{d}t$ produces the boundary terms

$$\mathfrak{B}_1 = \int_{\mathbb{R}^{y+}} \left(v\partial_x m_x - \partial_x v m_x\right) \mathrm{d}y + \int_{\mathbb{R}^{x+}} \left(v\partial_y m_y - \partial_y v m_y\right) \mathrm{d}x \tag{2.75}$$

and

$$\mathfrak{B}_2 = \int_{\mathbb{R}^{y+}} -2\partial_y m_{xy} v \,\mathrm{d}y + \int_{\mathbb{R}^{x+}} 2m_{xy}\partial_x v \,\mathrm{d}x, \tag{2.76}$$

respectively. By applying again integration by parts on the last term of $\mathfrak{B}_2$, an extra condition at the corner is produced

$$\mathfrak{B}_2 = \int_{\mathbb{R}^{y+}} -2\partial_y m_{xy} v \,\mathrm{d}y + \int_{\mathbb{R}^{x+}} -2\partial_x m_{xy} v \,\mathrm{d}x - \left(m_{xy}v\right)\big|_{x=0,y=0}. \tag{2.77}$$

In order for the energy to be conserved, $\mathfrak{B}_{\mathrm{rpl}}$ must identically vanish. This can be obtained by imposing suitable boundary conditions.

Boundary conditions

Boundary conditions in Cartesian coordinates can be obtained by comparing (2.75) and (2.77) with the general conditions (2.54), (2.55) and (2.56). Over the side at $y = 0$, e.g., these become

$$\text{(clamped)} \qquad w = 0, \quad \partial_x w = 0 \tag{2.78}$$

$$\text{(simply supported)} \qquad w = 0, \quad m_x = 0 \tag{2.79}$$

$$\text{(free)} \qquad m_x = 0, \quad \partial_x m_x + 2\partial_y m_{xy} = 0. \tag{2.80}$$

Note that $v = 0$ implies $w = 0$. At the corner ($x = 0$, $y = 0$), two conditions can alternatively be used, which correspond to a fixed or free corner:

$$\text{(fixed corner)} \qquad w = 0 \tag{2.81}$$

$$\text{(free corner)} \qquad m_{xy} = 0. \tag{2.82}$$

In the case of a finite rectangular domain D_{L_x,L_y}, the boundary conditions along the other edges can be obtained by symmetry.

Modes

Modal shapes for rectangular plates are generally difficult to obtain analytically [124]. Considering a harmonic solution of the form $W(x,y)\mathrm{e}^{\mathrm{i}\omega t}$, one is interested in solving the eigenvalue equation

$$\Delta^2 W(x,y) = \gamma^4 W(x,y), \tag{2.83}$$

with $\gamma^2 = \omega/\kappa$. One can attempt separation of variables to $W(x,y) = X(x)Y(y)$ in order to find the modes, but this method only works when two opposite simply supported edges are present [124, 91]. The simplest possible case is that with four simply supported edges—the modal shapes $W_{a,b}(x,y)$ can be written as

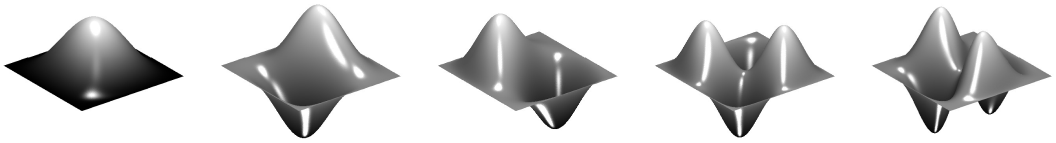
$$W_{a,b}(x,y) = A_{a,b} \sin\left(\frac{\pi a x}{L_x}\right) \sin\left(\frac{\pi b y}{L_y}\right), \tag{2.84}$$

where a and b are positive integers and $A_{a,b}$ is an appropriate normalising constant. The associated modal frequencies are given by

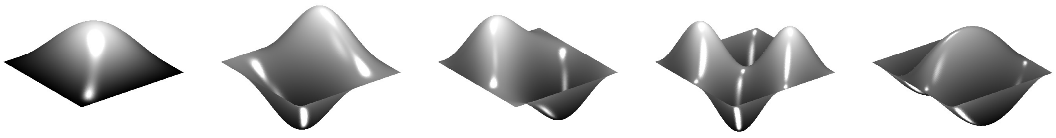
$$f_{a,b} = \frac{\omega_{a,b}}{2\pi} = \frac{\pi\kappa}{2}\left[\frac{a^2}{L_x^2} + \frac{b^2}{L_y^2}\right], \qquad a,b \geq 1. \tag{2.85}$$

In all other cases, given the lack of an analytic solution, numerical methods become an attractive option [73]. Figure 2.5 shows the first five modes for a square plate under three different boundary conditions. In the free case, three zero-frequency modes exist, corresponding to rigid body motion (one translation plus two rotations around the axes), but they are not shown in the picture.

(a) Clamped



(b) Simply supported



(c) Free

increasing mode number ⟶

Figure 2.5: First five modes in the modal series of a square plate in order of increasing frequency for three different boundary conditions: a) clamped, b) simply supported, c) free. In the free case, the first three modes corresponding to rigid body motion are not represented.

2.2.5 Circular plates

Circular plate vibration is of great importance for the simulation of gongs and linear drum vibration. The natural choice of coordinates for a problem defined over a circular domain is of course polar coordinates. Equation (2.43) used for the rectangular plates remains valid, provided one expresses the biharmonic operator in the new coordinates.

Consider harmonic functions of the form $w(r,\theta,t) = \mathrm{e}^{\mathrm{i}\omega t}W(r,\theta)$, for a given eigenfrequency ω. Assuming any of the boundary conditions given in Section 2.2.2, the solution $W(r,\theta)$ to the eigenvalue equation (2.83) can be expressed using separation of variables as

$$W(r,\theta) = R(r)\Theta(\theta). \tag{2.86}$$

It is well known that solutions to this equation can be written as a linear combination of Bessel's functions J_m and I_m in the radial coordinate, multiplied by a (co-)sinusoidal function in the angular variable:

$$W(r,\theta) = \begin{pmatrix}\cos(m\theta)\\ \sin(m\theta)\end{pmatrix}\left[AJ_m(\gamma r) + BI_m(\gamma r)\right], \tag{2.87}$$

for $m \geq 0$ integer, scaling constants A and B and $\gamma^2 = \omega/\kappa$ [124, 134, 91]. The presence of sine and cosine determines a degeneracy for all the modes with $m > 0$. In this case, in fact, two modal shapes are present which are identical, apart from a spatial phase shift, and are associated to the same modal frequency. This is similar to what happens in the case of a square plate.

The function $J_m(x)$ is Bessel's function of the first kind, defined as the solution to the differential equation

$$x^2\,\frac{\mathrm{d}^2 J_m(x)}{\mathrm{d}x^2} + x\,\frac{\mathrm{d}J_m(x)}{\mathrm{d}x} + (x^2 - m^2)J_m(x) = 0, \tag{2.88}$$

while $I_m(x)$ is the modified Bessel's function defined as

$$I_m(x) = \mathrm{i}^m J_m(\mathrm{i}x). \tag{2.89}$$

In order to find modal frequencies and modal shapes, one has to apply boundary conditions to (2.87), and solve transcendental equations involving Bessel's functions and their derivatives. In the clamped case, for example, conditions (2.54) can be rephrased as

$$W|_{r=R} = 0, \qquad \partial_r W|_{r=R} = 0, \tag{2.90}$$

where R is the radius of the plate. Applying these conditions leads to the following equations:

$$AJ_m(\gamma R) + BI_m(\gamma R) = 0 \tag{2.91a}$$

$$A\gamma J_m'(\gamma R) + B\gamma I_m'(\gamma R) = 0, \tag{2.91b}$$

where the prime denotes differentiation with respect to the argument. Solving for A in the first equation and substituting it back into the second, leads to the following transcendental equation in γR:

$$J_m(\gamma R)I'_m(\gamma R) = J'_m(\gamma R)I_m(\gamma R). \tag{2.92}$$

This equation has an infinite number of solutions in γ, which in turn give the allowed frequencies $\omega_{m,n}$ for given values of R and κ. Hence, every modal shape depends on two integers, m and n, which represent the number of nodal diameters and nodal circles, respectively. In the literature, it is customary to indicate the values for the square of the nondimensional frequencies $\zeta^2 = \gamma^2 R^2$, from which the frequencies $\omega_{m,n}$ can be inferred. Table 2.1 shows the first values of ζ^2, calculated numerically by solving (2.92).

n\m	**0**	**1**	**2**	**3**	**4**	**5**	**6**
0	10.22	21.26	34.88	51.03	69.67	90.74	114.21
1	39.77	60.83	84.58	111.02	140.11	171.80	206.07
2	89.10	120.08	153.82	190.30	229.52	271.43	316.00
3	158.18	199.05	242.72	289.18	338.41	390.39	445.09
4	247.01	297.76	351.34	407.73	466.93	528.90	593.64

Table 2.1: Values of the nondimensional frequencies squared $\zeta^2 = \gamma^2 R^2$ for a clamped circular plate. Note that in this table, the number of nodal circles n does not include the boundary circle.

2.3 Preliminaries II: Finite difference method

In this section, some basic facts about the finite difference method are presented. Here, the notation used in the remainder of this work is established.

2.3.1 Grid functions and domains

The numerical simulation of PDEs with the finite difference method [92] requires the use of grid functions that approximate their continuous counterparts at discrete locations in space and time. Let $w^n_{l_x,l_y}$ represent an approximation of the plate displacement $w(x,y,t)$ at position $(x = l_x h, y = l_y h)$ and time $t = nk$, where h and k are the spatial and temporal grid steps, respectively. In acoustical simulation it is useful to define the sampling rate F_s of a simulation *a priori*. The time step k is then obtained as the inverse of F_s,

$$k = 1/F_s. \tag{2.93}$$

For any of the domains $\mathcal{D}$ described in Section 2.1, the spatial indices l_x, l_y will be restricted to integer numbers such that, for each couple, the point $(l_x h, l_y h) \in \mathcal{D}$. This set of points will be denoted by $\mathfrak{D}$, with $\mathfrak{D} \subseteq \mathbb{Z}^2$. In some instances, infinite and semi-infinite domains will be considered for analysis purposes, as in the continuous case. The set of integer indices representing the infinite domain $\mathbb{R}^2$ will be denoted by $\mathfrak{D}_{\mathbb{R}^2}$, or equivalently by $\mathbb{Z}^2$. The discrete half plane will be denoted with $\mathfrak{D}_{\mathbb{H}}$, while the discrete quarter plane will be denoted with $\mathfrak{D}_{\mathbb{Q}}$. For a rectangular region with sides L_x and L_y, it is useful to consider the number of intervals along the x and y axis, N_x and N_y, that satisfy the relations

$$L_x = h\,N_x, \qquad L_y = h\,N_y, \tag{2.94}$$

with $\mathfrak{D}_{N_x,N_y} = \{(l_x, l_y) \in \mathbb{Z}^2,\, 0 \le l_x \le N_x,\, 0 \le l_y \le N_y\}$. The number of grid points along each axis will then be $N_x + 1$ and $N_y + 1$, respectively.

2.3.2 Finite difference operators

Differential operators can be substituted with discrete operators that act on the grid functions just introduced. Consider, first, the simple forward and backward shift operators in time, whose action on a grid function is defined as

$$\mathrm{s}_{t+} w^n_{l_x,l_y} = w^{n+1}_{l_x,l_y}, \qquad \mathrm{s}_{t-} w^n_{l_x,l_y} = w^{n-1}_{l_x,l_y}. \tag{2.95}$$

Continuous derivatives can be approximated in various ways by using combinations of these operators. The first time derivative can be discretised with forward, backward and centred operators, respectively:

$$\delta_{t+} = \frac{\mathrm{s}_{t+} - 1}{k}, \qquad \delta_{t-} = \frac{1 - \mathrm{s}_{t-}}{k}, \qquad \delta_{t\cdot} = \frac{\mathrm{s}_{t+} - \mathrm{s}_{t-}}{2k}, \tag{2.96}$$

while the second difference operator may be written as

$$\delta_{tt} = \delta_{t+}\delta_{t-} = \frac{\mathrm{s}_{t+} - 2 + \mathrm{s}_{t-}}{k^2}. \tag{2.97}$$

Additional operators that approximate the identity are called averaging operators and can be defined as follows:

$$\mu_{t+} = \frac{\mathrm{s}_{t+} + 1}{2}, \qquad \mu_{t-} = \frac{\mathrm{s}_{t-} + 1}{2}, \qquad \mu_{t\cdot} = \frac{\mathrm{s}_{t+} + \mathrm{s}_{t-}}{2}. \tag{2.98}$$

Analogous definitions hold for the spatial operators. Shift operators are defined as

$$\mathrm{s}_{x+} w^n_{l_x,l_y} = w^n_{l_x+1,l_y}, \quad \mathrm{s}_{x-} w^n_{l_x,l_y} = w^n_{l_x-1,l_y}, \tag{2.99}$$

$$\mathrm{s}_{y+} w^n_{l_x,l_y} = w^n_{l_x,l_y+1}, \quad \mathrm{s}_{y-} w^n_{l_x,l_y} = w^n_{l_x,l_y-1}. \tag{2.100}$$

Forward, backward and centred differences in x are obtained as

$$\delta_{x+} = \frac{\mathrm{s}_{x+} - 1}{h}, \qquad \delta_{x-} = \frac{1 - \mathrm{s}_{x-}}{h}, \qquad \delta_{x\cdot} = \frac{\mathrm{s}_{x+} - \mathrm{s}_{x-}}{2h}, \tag{2.101}$$

with the operators along the y axis defined in analogous way. Finally, second differences are defined as

$$\delta_{xx} = \delta_{x+}\delta_{x-} = \frac{\mathrm{s}_{x-} - 2 + \mathrm{s}_{x+}}{h^2}, \qquad \delta_{yy} = \delta_{y+}\delta_{y-} = \frac{\mathrm{s}_{y-} - 2 + \mathrm{s}_{y+}}{h^2}. \tag{2.102}$$

The discrete approximation δ_Δ of the continuous Laplacian Δ is obtained by combining these operators,

$$\delta_\Delta = \delta_{xx} + \delta_{yy} = \frac{s_{x-} + s_{y-} - 4 + s_{x+} + s_{y+}}{h^2}. \tag{2.103}$$

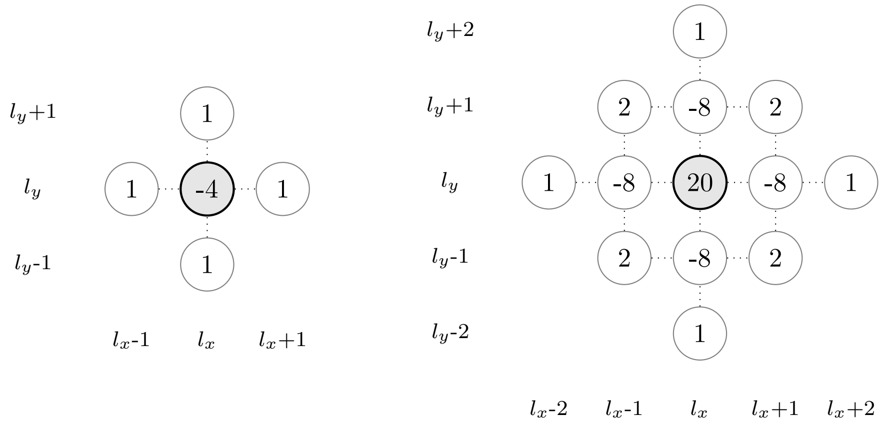
Fourth difference operators can be defined as

$$\delta_{xxxx} = \delta_{xx}\delta_{xx} = \frac{s_{x-}^2 - 4s_{x-} + 6 - 4s_{x+} + s_{x+}^2}{h^4}, \tag{2.104}$$

with analogous definition for δ_{yyyy}. The discrete approximation of the biharmonic operator Δ^2 can be built as follows:

$$\delta_{\Delta,\Delta} = \delta_\Delta\delta_\Delta = \delta_{xxxx} + 2\delta_{xx}\delta_{yy} + \delta_{yyyy}. \tag{2.105}$$

The spatial stencils for the Laplacian and biharmonic operators are shown in Figure 2.6, together with the coefficients of the various nodes, scaled by h^2 and h^4, respectively. Notice that these are but the simplest ways to discretise the Laplacian and biharmonic operators, and other possibilities are available [15, 24].



(a) Laplacian operator (b) Biharmonic operator

Figure 2.6: Stencils for the Laplacian operator (2.103) and for the biharmonic operator (2.105), scaled by a factor h^2 and h^4, respectively. The node at which the difference operator is calculated is marked in gray.

Accuracy of finite difference operators

The analysis of the error that is produced when a differential operator is discretised with finite differences is an important issue, which is dealt with in detail in several works [92, 168, 19]. The classic approach is to apply the discrete operator to an infinitely differentiable function, and to estimate the truncation error through a Taylor series expansion. In this way, it is possible to show that the centred operators introduced before are second-order accurate. Assuming $f(t)$ to be a function of time, one can write:

$$\delta_{t\cdot} f(t) = \frac{1}{2k}\big(f(t+k) - f(t-k)\big) = \frac{\mathrm{d}f}{\mathrm{d}t} + \frac{k^2}{6}\frac{\mathrm{d}^3 f}{\mathrm{d}t^3} + \ldots \tag{2.106}$$

As such, the error committed in the evaluation of the first derivative with a centred discrete operator is proportional to k^2. A similar reasoning can be applied to the second difference δ_{tt}, and can be extended to spatial operators, as well.

It is clear from this analysis that for any scheme, increasing the sampling rate of the simulation (thus reducing the time and spatial steps) leads to progressively more accurate approximations of the problem at hand. This, however, is achieved at the expense of a larger number of grid points and, consequently, of higher memory cost and operation count. The rate at which these scale depends on the particular system under consideration. In the worst case, that of 3D room acoustics simulations, memory locations and number of operations grow with the third and fourth power of the sampling rate, respectively [19]. One must then weigh very carefully the possibility of oversampling the numerical simulation, and maybe try the use of different and more accurate schemes first. However, when balanced with the disadvantages coming from the increased computational cost, this strategy can always be applied to ameliorate the behaviour of the scheme.

Some useful identities

Some identities involving discrete operators are sometimes useful in energy manipulations. Here they are given without proof (but see [19] for further information):

$$f\delta_{t\cdot}f = \frac{1}{2}\delta_{t+}(f\mathrm{s}_{t-}f). \tag{2.107a}$$

$$\delta_{tt} = \frac{2}{k}\left(\delta_{t\cdot} - \delta_{t-}\right) \tag{2.107b}$$

$$f\mathrm{s}_{t\pm}f = \left(\mu_{t\pm}f\right)^2 - \frac{k^2}{4}\left(\delta_{t\pm}f\right)^2 \tag{2.107c}$$

2.3.3 Discrete inner products

A discrete analogue of the continuous inner product (2.18) can be introduced in order to perform energy analysis on the system. Given two grid functions f_{l_x,l_y} and g_{l_x,l_y}, defined over any discrete domain $\mathfrak{D}$, the discrete inner product and the associated norm are defined as

$$\langle f, g\rangle_{\mathfrak{D}} = \sum_{(l_x,l_y)\in\mathfrak{D}} h^2 f_{l_x,l_y} g_{l_x,l_y}, \qquad \|f\|_{\mathfrak{D}} = \sqrt{\langle f, f\rangle_{\mathfrak{D}}}. \tag{2.108}$$

Notice that the same notation is used for the continuous and discrete inner product and norm. The context and the indication of the domain where these are calculated should clearly suggest which instance is considered. In the discrete case, summation by parts has the role of integration by parts. When the domain of integration is finite, boundary terms appear every time summation by parts is used [19]. Consider, e.g., the half plane $\mathfrak{D}_{\mathbb{H}^x}$, with $x \geq 0$. The following relations hold:

$$\langle f, \delta_{x-}g\rangle_{\mathfrak{D}_{\mathbb{H}^x}} = -\langle \delta_{x+}f, g\rangle_{\mathfrak{D}_{\mathbb{H}^x}} - \{f, \mathrm{s}_{x-}g\}_{0,\mathbb{Z}}, \tag{2.109a}$$

$$\langle f, \delta_{xx}g\rangle_{\mathfrak{D}_{\mathbb{H}^x}} = \langle \delta_{xx}f, g\rangle_{\mathfrak{D}_{\mathbb{H}^x}} - \{f, \delta_{x-}g\}_{0,\mathbb{Z}} + \{\delta_{x-}f, g\}_{0,\mathbb{Z}}. \tag{2.109b}$$

The notation $\{\cdot,\cdot\}$ introduced in [19] indicates a 1D sum over the boundary of the region $\mathfrak{D}_{\mathbb{H}^x}$; in the first case, one has

$$\{f, \mathrm{s}_{x-}g\}_{0,\mathbb{Z}} = \sum_{l_y=-\infty}^{\infty} h f_{0,l_y} g_{-1,l_y}. \tag{2.110}$$

As is apparent, the function g is evaluated at a location outside the half plane $\mathfrak{D}_{\mathbb{H}^x}$. These locations are referred to as "ghost" or virtual points, and their values can be determined from the numerical boundary conditions. Many such examples will be discussed in the remainder of this work.

Some useful inequalities

Some useful relations are now presented, which hold for any two discrete functions f and g defined over the infinite discrete domain $\mathbb{Z}^2$ (see [19]):

$$\|f+g\|_{\mathbb{Z}^2} \le \|f\|_{\mathbb{Z}^2} + \|g\|_{\mathbb{Z}^2}\,, \tag{2.111a}$$

$$\|\delta_{x\pm} f\|^2_{\mathbb{Z}^2} \le \frac{4}{h^2}\,\|f\|^2_{\mathbb{Z}^2}\,, \tag{2.111b}$$

$$\|\delta_{xx} f\|^2_{\mathbb{Z}^2} \le \frac{16}{h^4}\,\|f\|^2_{\mathbb{Z}^2}\,, \tag{2.111c}$$

$$\|\delta_{\Delta} f\|^2_{\mathbb{Z}^2} \le 4\frac{16}{h^4}\,\|f\|^2_{\mathbb{Z}^2}\,. \tag{2.111d}$$

Weighted inner product

A different choice is the weighted inner product defined as

$$\langle f, g\rangle'_{\mathfrak{D}} = \sum_{(l_x,l_y)\in\mathfrak{D}} \alpha_{l_x,l_y} h^2 f_{l_x,l_y} g_{l_x,l_y}, \tag{2.112}$$

where α_{l_x,l_y} is a coefficient that depends on N_n, the number of nearest neighbours of the grid point that lie outside of the grid, in the following way:

$$\alpha_{l_x,l_y} = 2^{-N_n}, \qquad \text{with} \qquad N_n = 0, 1, 2. \tag{2.113}$$

Here, $N_n = 0$ for an interior point, $N_n = 1$ for an edge point, and $N_n = 2$ for a corner point. The coefficient α_{l_x,l_y} can be related to the area of the cell surrounding the grid point near the boundaries [96] in a finite volume framework, leading to more precise boundary conditions (see Section 2.4.1).

2.3.4 Matrix interpretation of finite difference schemes

For the sake of generality and ease of programming, it can be useful to recast finite difference schemes in a state-space form. Regardless of their dimensionality, grid functions can be easily written in vector form. Consider, for example, Figure 2.7, where a rectangular plate is approximated with a 2D grid: the values of $w^n_{l_x,l_y}$ can be arranged into a vector $\mathbf{w}^n$ by concatenating columns along the y-axis with increasing value of x. For 3D grids, as those involving, e.g., wave propagation in air, the same approach can be repeated for successive slices with increasing vertical coordinate.

It is now possible to express discrete spatial operators in terms of matrices [92]. Consider first a 1D strip of contiguous points along the x axis. Leaving aside boundary conditions for the moment, the second difference operator (scaled by h^2) can be written as

$$h^2\,\mathbf{D}_{xx} = \underbrace{\begin{bmatrix} \ddots & \ddots & & & \\ 1 & \text{-2} & 1 & & \\ & 1 & \text{-2} & 1 & \\ & & 1 & \text{-2} & 1 \\ & & & \ddots & \ddots \end{bmatrix}}_{N_x+1 \text{ columns}} \left.\vphantom{\begin{bmatrix}a\\a\\a\\a\\a\end{bmatrix}}\right\} N_x+1 \text{ rows} \tag{2.114}$$

A similar definition holds for $\mathbf{D}_{yy}$ operating along the y axis, which this time will be of side length $N_y + 1$.

The difference matrices $\mathbf{D}^{(2)}_{xx}$ and $\mathbf{D}^{(2)}_{yy}$ operating over the entire 2D grid can be constructed in the following way:

$$\mathbf{D}^{(2)}_{xx} = \mathbf{D}_{xx} \otimes \mathbb{1}_{N_y+1}, \qquad \mathbf{D}^{(2)}_{yy} = \mathbb{1}_{N_x+1} \otimes \mathbf{D}_{yy}, \tag{2.115}$$

where $\mathbb{1}_N$ represents an identity matrix of size N and the symbol $\otimes$ denotes the Kronecker product of two matrices. The superscript (2) will be avoided whenever there is no risk of confusion about the dimension of the matrix. The Laplacian and biharmonic operators $\mathbf{D}_\Delta$ and $\mathbf{D}_{\Delta,\Delta}$, matrix representations of the operators (2.105), can be written as

$$\mathbf{D}_\Delta = \mathbf{D}^{(2)}_{xx} + \mathbf{D}^{(2)}_{yy}, \qquad \mathbf{D}_{\Delta,\Delta} = \mathbf{D}_\Delta \mathbf{D}_\Delta. \tag{2.116}$$

The sparsity patterns for $\mathbf{D}^{(2)}_{xx}$, $\mathbf{D}^{(2)}_{yy}$, $\mathbf{D}_\Delta$ and $\mathbf{D}_{\Delta,\Delta}$ are plotted in Figure 2.8.

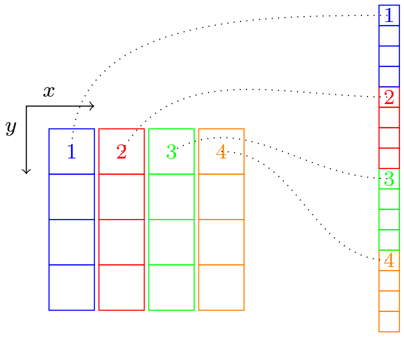
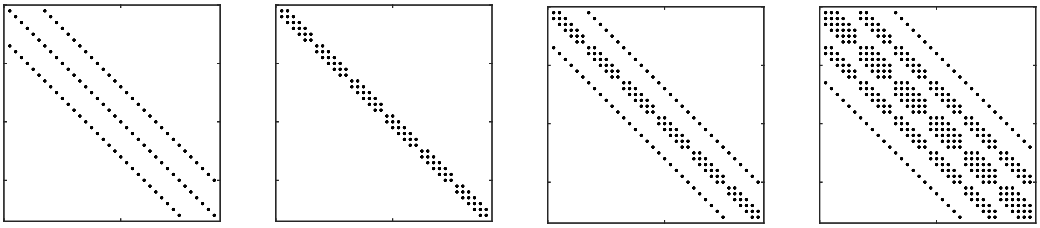


Figure 2.7: Illustration of the vectorisation of a 2D grid function. A vector is created by concatenating successive stripes along the y axis.

2.4 Finite difference schemes for rectangular plates

In this section, the simulation of rectangular plates is presented. In order to simplify the discussion, a finite difference scheme is derived first for the infinite domain $\mathbb{Z}^2$, such that the energy analysis be not complicated by the presence of boundary conditions. These are introduced and dealt with in a second stage.



(a) $\mathbf{D}^{(2)}_{xx}$ (b) $\mathbf{D}^{(2)}_{yy}$ (c) $\mathbf{D}_{\Delta}$ (d) $\mathbf{D}_{\Delta,\Delta}$

Figure 2.8: Sparsity plots for the finite difference operators in matrix form for a 2D grid with 6×6 points.

2.4.1 Interleaved scheme and energy analysis

As anticipated in Section 2.2.4, a numerical discretisation of (2.66) can be the starting point for the simulation of rectangular plates. To this end, first introduce the discrete variables

$$v^{n+1/2}_{l_x,l_y}, \quad m^{n}_{x\,l_x,l_y}, \quad m^{n}_{y\,l_x,l_y}, \quad m^{n}_{xy\,l_x+1/2,l_y+1/2}, \tag{2.117}$$

defined over interleaved grids in space and time (see Figures 2.9 and 2.10). The discrete variables v, m_x and m_y are placed at the nodes of the main grid, while the values of m_{xy} lie at the nodes of the dual grid, which coincide with the centres of the main grid. The velocity $v^{n+1/2}_{l_x,l_y} = \delta_{t+} w^{n}_{l_x,l_y}$, which is defined as the discrete time derivative of the plate displacement w^n, is placed at interleaved time steps with respect to the other variables, in order for the scheme to be centred at time nk. A discrete approximation of system (2.66) can be written as

$$\rho H \delta_{t-} v = \delta_{xx} m_x + \delta_{yy} m_y + 2\delta_{x-,y-} m_{xy} \tag{2.118a}$$

$$\delta_{t+} m_x = -\,D\left(\delta_{xx} v + \nu \delta_{yy} v\right) \tag{2.118b}$$

$$\delta_{t+} m_y = -\,D\left(\delta_{yy} v + \nu \delta_{xx} v\right) \tag{2.118c}$$

$$\delta_{t+} m_{xy} = -\,D\left(1-\nu\right) \delta_{x+,y+} v, \tag{2.118d}$$

where the spatial and time dependence of the variables has been omitted for readability. Here and in the remainder of the section, every instance of v stands for $v^{n+1/2}_{l_x,l_y}$, and a similar notation is used for the other variables defined in (2.117). When all the variables at time step nk are supposed known, the update can be obtained by solving (2.118a) first, and then using the result to calculate the moments using the other three equations.

A discrete analogue of the energy (2.73) can be obtained by taking an inner product of $\mu_{t-} v$ with (2.118a) over the infinite domain $\mathbb{Z}^2$

$$\rho H \langle \mu_{t-} v, \delta_{t-} v \rangle_{\mathbb{Z}^2} = \langle \mu_{t-} v, \delta_{xx} m_x + \delta_{yy} m_y + 2\delta_{x-,y-} m_{xy} \rangle_{\mathbb{Z}^2} \tag{2.119a}$$

$$= \underbrace{\langle \mu_{t-} \delta_{xx} v, m_x \rangle_{\mathbb{Z}^2} + \langle \mu_{t-} \delta_{yy} v, m_y \rangle_{\mathbb{Z}^2}}_{= \delta_{t+} \mathfrak{u}_1} + \underbrace{2 \langle \mu_{t-} \delta_{x+,y+} v, m_{xy} \rangle_{\mathbb{Z}^2}}_{= \delta_{t+} \mathfrak{u}_2}, \tag{2.119b}$$

where, as in the continuous case, the potential energy $\mathfrak{u} = \mathfrak{u}_1 + \mathfrak{u}_2$ is given by the separate contributions of $\mathfrak{u}_1$ and $\mathfrak{u}_2$. Similarly to the analysis done in Section 2.2.4 for the continuous case, equations (2.118b) and (2.118c) can be easily inverted to obtain

$$\begin{bmatrix} \delta_{xx}v \\ \delta_{yy}v \end{bmatrix} = -\frac{1}{D(1-\nu^2)} \begin{bmatrix} 1 & -\nu \\ -\nu & 1 \end{bmatrix} \begin{bmatrix} \delta_{t+}m_x \\ \delta_{t+}m_y \end{bmatrix}. \tag{2.120}$$

Using this relation, the term containing the potential energy $\mathfrak{u}_1$ can be further manipulated as

$$\delta_{t+}\mathfrak{u}_1 = -\frac{1}{D(1-\nu^2)}\Big[\langle \delta_{t\cdot}(m_x - \nu m_y), m_x\rangle_{\mathbb{Z}^2} + \langle \delta_{t\cdot}(m_y - \nu m_x), m_y\rangle_{\mathbb{Z}^2}\Big] \tag{2.121a}$$

$$= -\frac{1}{D(1-\nu^2)}\Big[\langle \delta_{t\cdot}m_x, m_x\rangle_{\mathbb{Z}^2} + \langle \delta_{t\cdot}m_y, m_y\rangle_{\mathbb{Z}^2} - \nu\left(\langle \delta_{t\cdot}m_x, m_y\rangle_{\mathbb{Z}^2} + \langle \delta_{t\cdot}m_y, m_x\rangle_{\mathbb{Z}^2}\right)\Big] \tag{2.121b}$$

$$\overset{(2.107a)}{=} -\frac{1}{2D(1-\nu^2)}\delta_{t+}\Big[\langle m_x, \mathrm{s}_{t-}m_x\rangle_{\mathbb{Z}^2} + \langle m_y, \mathrm{s}_{t-}m_y\rangle_{\mathbb{Z}^2} - \nu\langle m_x, \mathrm{s}_{t-}m_y\rangle_{\mathbb{Z}^2} - \nu\langle m_y, \mathrm{s}_{t-}m_x\rangle_{\mathbb{Z}^2}\Big]. \tag{2.121c}$$

Using (2.118d), the term containing $\mathfrak{u}_2$ can be written as

$$\delta_{t+}\mathfrak{u}_2 = -\frac{2}{D(1-\nu)}\langle \delta_{t\cdot}m_{xy}, m_{xy}\rangle_{\mathbb{Z}^2} \overset{(2.107a)}{=} -\frac{1}{D(1-\nu)}\delta_{t+}\langle m_{xy}, \mathrm{s}_{t-}m_{xy}\rangle_{\mathbb{Z}^2}. \tag{2.122}$$

Thus, the discrete energy of the plate $\mathfrak{h}_{\mathrm{rpl}} \equiv \mathfrak{h}_{\mathrm{rpl}}^{n+1/2}$, a scalar time series, is conserved:

$$\delta_{t+}\mathfrak{h}_{\mathrm{rpl}} = 0, \tag{2.123}$$

similarly to the continuous case (2.72). The explicit expression for $\mathfrak{h}_{\mathrm{rpl}}$ can be written as

$$\mathfrak{h}_{\mathrm{rpl}} = \frac{\rho H}{2}\|v\|^2_{\mathbb{Z}^2} + \frac{1}{D(1-\nu)}\langle m_{xy}, \mathrm{s}_{t-}m_{xy}\rangle_{\mathbb{Z}^2} +$$
$$\frac{1}{2D(1-\nu^2)}\left(\langle m_x, \mathrm{s}_{t-}m_x\rangle_{\mathbb{Z}^2} + \langle m_y, \mathrm{s}_{t-}m_y\rangle_{\mathbb{Z}^2} - \nu\langle m_x, \mathrm{s}_{t-}m_y\rangle_{\mathbb{Z}^2} - \nu\langle m_y, \mathrm{s}_{t-}m_x\rangle_{\mathbb{Z}^2}\right). \tag{2.124}$$

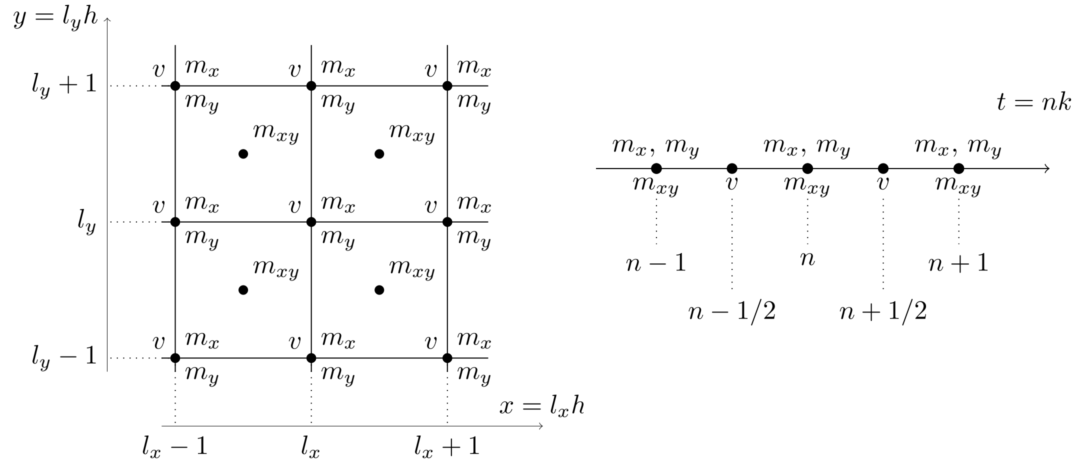


Figure 2.9: Staggered grid in space (left) and time (right) for the plate equation.

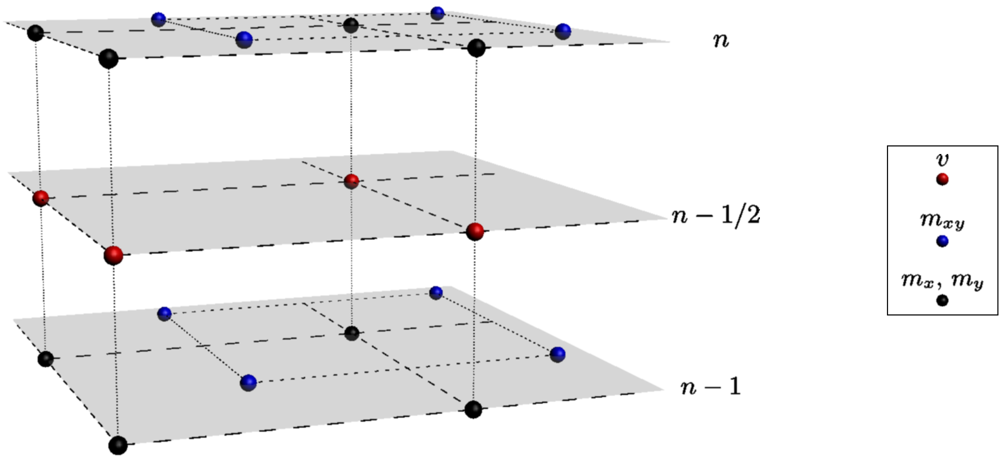


Figure 2.10: Staggered grid in space and time. The black nodes represent the main grid for the variables m_x and m_y at integer multiples of the time step, the blue nodes represent m_{xy} on the dual grid while the red nodes represent v over the main grid but at half integer time steps.

2.4.2 Boundary conditions

The finite difference scheme (2.118) must be accompanied by appropriate boundary conditions. These may be derived by applying energy techniques, as in the continuous case. To this end, consider the discrete quarter plane $\mathfrak{D}_{\mathbb{Q}}$. Starting from (2.119a), the summation by parts performed using (2.109b) in order to arrive at (2.119b) now produces the boundary terms $\mathfrak{b}_1$ and $\mathfrak{b}_2$, defined as

$$\mathfrak{b}_1 = \{\mu_{t-}v, \delta_{x-}m_x\}_{0,\mathbb{Z}} - \{\delta_{x-}\mu_{t-}v, m_x\}_{0,\mathbb{Z}} + \{\mu_{t-}v, \delta_{y-}m_y\}_{\mathbb{Z},0} - \{\delta_{y-}\mu_{t-}v, m_y\}_{\mathbb{Z},0} \tag{2.125}$$

and

$$\mathfrak{b}_2 = -2\{\mu_{t-}v, \delta_{y-}m_{xy}\}_{0,\mathbb{Z}} + 2\{\delta_{x-}\mu_{t-}v, m_{xy}\}_{\mathbb{Z},0}. \tag{2.126}$$

This last expression can be modified by applying summation by parts on the second term,

$$\mathfrak{b}_2 = -2\{\mu_{t-}v, \delta_{y-}m_{xy}\}_{0,\mathbb{Z}} - 2\{\mu_{t-}v, \delta_{x-}m_{xy}\}_{\mathbb{Z},0} - v|_{0,0}\, m_{xy}|_{-1/2,-1/2}\,. \tag{2.127}$$

The energy balance in this case becomes

$$\delta_{t+}\mathfrak{h}_{\mathrm{rpl}} = \mathfrak{b}_{\mathrm{rpl}}, \qquad \mathfrak{b}_{\mathrm{rpl}} = \mathfrak{b}_1 + \mathfrak{b}_2, \tag{2.128}$$

which is analogous to (2.74) in the continuous case. Once again, in order for the scheme to be lossless, all the boundary terms must identically vanish. This can be achieved by imposing the appropriate boundary conditions, as discussed in the next paragraph.

Non-centred boundary conditions

Boundary conditions analogous to (2.78)-(2.80) in the continuous case must be imposed in the numerical scheme. These can be deduced by requiring the boundary terms $\mathfrak{b}_1 + \mathfrak{b}_2$ to vanish.

Over the side at $l_y = 0$, e.g., three physically meaningful conditions may be derived:

$$\text{(clamped)} \qquad v|_{l_y=0} = 0, \quad \delta_{x-} v|_{l_y=0} = 0 \tag{2.129}$$

$$\text{(simply supported)} \qquad v|_{l_y=0} = 0, \quad m_x|_{l_y=0} = 0 \tag{2.130}$$

$$\text{(free)} \qquad m_x|_{l_y=0} = 0, \quad \delta_{x-} m_x|_{l_y=0} + 2\,\delta_{y-} m_{xy}|_{l_y=-1/2} = 0. \tag{2.131}$$

A set of similar conditions may be deduced for the side at $l_x = 0$. In the free case, the corner conditions (2.81) and (2.82) become

$$\text{(fixed corners)} \qquad v|_{l_x=0, l_y=0} = 0 \tag{2.132}$$

$$\text{(free corners)} \qquad m_{xy}|_{l_x=-1/2, l_y=-1/2} = 0. \tag{2.133}$$

It is interesting to note that the clamped and free conditions are non-centred around the side of the plate. More precise, centred conditions can be arrived at using the weighted inner product (2.108).

Centred boundary conditions

In order to improve the accuracy of the numerical scheme, centred conditions can be implemented at the boundaries. Such conditions can be arrived at by applying energy techniques similar to those used before, but for a different choice of inner product. One possibility is the use of the weighted inner product defined in (2.112). When energy analysis is performed on the system using this modified inner product, different conditions which are centred over the boundaries are produced, while the rest of the calculation remains the same. Over the side at $l_y = 0$, these may be written as

$$\text{(centred clamped)} \qquad v|_{l_y=0} = 0, \quad \delta_{x\cdot} v|_{l_y=0} = 0 \tag{2.134}$$

$$\text{(simply supported)} \qquad v|_{l_y=0} = 0, \quad m_x|_{l_y=0} = 0 \tag{2.135}$$

$$\text{(centred free)} \qquad m_x|_{l_y=0} = 0, \quad \delta_{x\cdot} m_x|_{l_y=0} + 2\,\mu_{x+}\delta_{y-} m_{xy}|_{l_y=-1/2} = 0. \tag{2.136}$$

In the free case, the corner conditions (2.81) and (2.82) become

$$\text{(fixed corners)} \qquad v|_{l_x=0, l_y=0} = 0 \tag{2.137}$$

$$\text{(centred free corners)} \qquad \mu_{x+,y+} m_{xy}|_{l_x=-1/2, l_y=-1/2} = 0. \tag{2.138}$$

Notice that, since v and m_x are defined on the boundary, simply supported conditions (2.135) are always centred, regardless of the choice of inner product. Analogous conditions can be applied on the other sides of the domain.

2.4.3 Matrix implementation and fourth order scheme

When all the variables are consolidated into vector form and all the operators are expressed as matrices, the system (2.118) can be written as

$$\begin{aligned}
\rho H \delta_{t-} \mathbf{v}^{n+1/2} &= \mathbf{D}^{\dagger}_{xx}\mathbf{m}^n_x + \mathbf{D}^{\dagger}_{yy}\mathbf{m}^n_y + 2\mathbf{D}^{\dagger}_{x-,y-}\mathbf{m}^n_{xy} && (2.139a)\\
\delta_{t+}\mathbf{m}^n_x &= -D\left(\mathbf{D}^{\ddagger}_{xx}\mathbf{v}^{n+1/2} + \nu\mathbf{D}^{\ddagger}_{yy}\mathbf{v}^{n+1/2}\right) && (2.139b)\\
\delta_{t+}\mathbf{m}^n_y &= -D\left(\mathbf{D}^{\ddagger}_{yy}\mathbf{v}^{n+1/2} + \nu\mathbf{D}^{\ddagger}_{xx}\mathbf{v}^{n+1/2}\right) && (2.139c)\\
\delta_{t+}\mathbf{m}^n_{xy} &= -D\,(1-\nu)\,\mathbf{D}^{\ddagger}_{x+,y+}\mathbf{v}^{n+1/2}. && (2.139d)
\end{aligned}$$

The time indices have been reinstated here, as this formulation now bears strong resemblance with how the scheme is actually implemented. The matrices $\mathbf{D}^{\dagger}$ and $\mathbf{D}^{\ddagger}$ with the relative subscripts are the matrix representation of the finite difference operators appearing in (2.118). Their construction has been briefly discussed in Section 2.3.4. Boundary conditions can now be incorporated in each individual matrix, and the biharmonic operator can thus be obtained by combining them together:

$$\mathbf{D}_{\Delta,\Delta} = \mathbf{D}^{\dagger}_{xx}\mathbf{D}^{\ddagger}_{xx} + \mathbf{D}^{\dagger}_{yy}\mathbf{D}^{\ddagger}_{yy} + 2(1-\nu)\mathbf{D}^{\dagger}_{x+,y+}\mathbf{D}^{\ddagger}_{x-,y-} + \nu\left(\mathbf{D}^{\dagger}_{xx}\mathbf{D}^{\ddagger}_{yy} + \mathbf{D}^{\dagger}_{yy}\mathbf{D}^{\ddagger}_{xx}\right). \quad (2.140)$$

The presence of ν seems to contradict the simple expression (2.105) for the biharmonic operator. It is to be noted, however, that the terms multiplying Poisson's ratio in (2.140) exactly cancel in the interior of the plate, leaving unbalanced terms only near the boundaries. These are necessary in the free case for the correct boundary conditions to appear.

Simple scheme

As demonstrated in the continuous case, the first order system in v and the moments is mathematically equivalent to the fourth order equation (2.43). It is not surprising, then, that such an equivalence exist in the discrete case, as well. By combining the equations in (2.139) and remembering the relation $\mathbf{v} = \delta_{t+}\mathbf{w}$ between the velocity and the displacement in vector form, it is possible to arrive at the recursion

$$\rho H \delta_{tt}\mathbf{w} = -D\,\mathbf{D}_{\Delta,\Delta}\mathbf{w}. \quad (2.141)$$

The explicit update for $\mathbf{w}^{n+1}$ can be thus obtained as

$$\mathbf{w}^{n+1} = \mathbf{B}\mathbf{w}^n - \mathbf{w}^{n-1}, \qquad \mathbf{B} = \left(2\,\mathbb{1} - k^2\kappa^2\mathbf{D}_{\Delta,\Delta}\right), \quad (2.142)$$

with $\kappa = \sqrt{D/\rho H}$ and $\mathbb{1}$ the identity matrix. This scheme has the undoubted advantage of requiring only a single variable to be updated, using just a matrix-vector multiplication and one addition, and is therefore very easy to implement. Looking at (2.142) in more detail, the update at some point (l_x, l_y) in the interior of the plate can be written as

$$\begin{aligned} w^{n+1}_{l_x,l_y} =&(2-20\mu^2)w^n_{l_x,l_y} + 8\mu^2\left(w^n_{l_x+1,l_y} + w^n_{l_x-1,l_y} + w^n_{l_x,l_y+1} + w^n_{l_x,l_y-1}\right) \\ &- 2\mu^2\left(w^n_{l_x+1,l_y+1} + w^n_{l_x-1,l_y-1} + w^n_{l_x-1,l_y+1} + w^n_{l_x+1,l_y-1}\right) \\ &- \mu^2\left(w^n_{l_x+2,l_y} + w^n_{l_x-2,l_y} + w^n_{l_x,l_y+2} + w^n_{l_x,l_y-2}\right) - w^{n-1}_{l_x,l_y}, \end{aligned} \tag{2.143}$$

where $\mu = k\kappa/h^2$ is a dimensionless parameter that will be set according to stability conditions (see below). Figure 2.11 shows the diagram of the grid points required for the update of the scheme. The spatial stencil for the biharmonic operator was given in Figure 2.6b, and can be compared with the coefficients in (2.143).

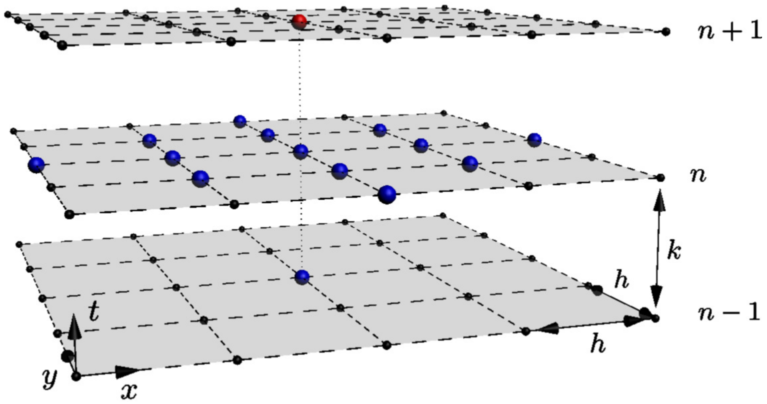


Figure 2.11: Update diagram for the plate scheme, showing the spatial and temporal dependency of the unknown grid point (red) with respect to the known values (blue).

An energy expression for this one-variable scheme can be obtained from (2.124) by substituting the expressions for m_x, m_y and m_{xy} given in (2.118b), (2.118c) and (2.118d), respectively. With a few elementary manipulations, the following expression for the energy can be obtained:

$$\begin{aligned} \mathfrak{h}^{n+1/2}_{\text{rpl}} =& \frac{\rho H}{2}\left\|\delta_{t+}w^n\right\|^2_{\mathfrak{D}} + D(1-\nu)\left(\langle\delta_{x+,y+}w^{n+1}, \delta_{x+,y+}w^n\rangle_{\mathfrak{D}}\right) + \\ &\frac{D}{2}\left[\nu\langle\delta_\Delta w^{n+1}, \delta_\Delta w^n\rangle_{\mathfrak{D}} + (1-\nu)\left(\langle\delta_{xx}w^{n+1}, \delta_{xx}w^n\rangle_{\mathfrak{D}} + \langle\delta_{yy}w^{n+1}, \delta_{yy}w^n\rangle_{\mathfrak{D}}\right)\right]. \end{aligned} \tag{2.144}$$

This coincides with that given in [19] using *ad hoc* considerations.

Stability of the scheme

Given a particular value of the time step k, in order for the scheme (2.141) to be stable, the spatial grid step must be chosen according to the condition

$$\frac{k\kappa}{h^2} = \mu \le \mu_{\max} = \frac{1}{4} \quad \Longrightarrow \quad h \ge h_{\min} = 2\sqrt{k\kappa}. \tag{2.145}$$

Energy techniques can be applied in order to arrive at this inequality, and the full calculation is postponed to the next paragraph. The idea is to bound the energy expression (2.144) from below with a positive quantity, in order to guarantee the non-negativity of $\mathfrak{h}_{\text{rpl}}$. This approach leads to the same condition as von Neumann analysis, which is discussed in [19].

Proof of stability

Consider the discrete energy $\mathfrak{h}_{\text{rpl}}$ for the rectangular plate defined in (2.144). This can be bounded from below in the following way using (2.107c):

$$\begin{aligned}\mathfrak{h}_{\text{rpl}} \geq \frac{\rho H}{2}\left\|\delta_{t+}w\right\|^2_{\mathfrak{D}} + D(1-\nu)\left(\left\|\mu_{t+}\delta_{x+,y+}w\right\|^2_{\mathfrak{D}} - \frac{k^2}{4}\left\|\delta_{t+}\delta_{x+,y+}w\right\|^2_{\mathfrak{D}}\right) + \\ \frac{D}{2}\left[\nu\left(\left\|\mu_{t+}\delta_{\Delta}w\right\|^2_{\mathfrak{D}} - \frac{k^2}{4}\left\|\delta_{t+}\delta_{\Delta}w\right\|^2_{\mathfrak{D}}\right) + (1-\nu)\left(\left\|\mu_{t+}\delta_{xx}w\right\|^2_{\mathfrak{D}} - \frac{k^2}{4}\left\|\delta_{t+}\delta_{xx}w\right\|^2_{\mathfrak{D}}\right)\right. \\ \left. + (1-\nu)\left(\left\|\mu_{t+}\delta_{yy}w\right\|^2_{\mathfrak{D}} - \frac{k^2}{4}\left\|\delta_{t+}\delta_{yy}w\right\|^2_{\mathfrak{D}}\right)\right].\end{aligned} \tag{2.146}$$

Considering now the terms with the minus sign, these can be manipulated as follows

$$-\frac{k^2}{4}D\frac{(1-\nu)}{2}\left[2\left\|\delta_{t+}\delta_{x+,y+}w\right\|^2_{\mathfrak{D}} + \frac{\nu}{(1-\nu)}\left\|\delta_{t+}\delta_{\Delta}w\right\|^2_{\mathfrak{D}} + \left\|\delta_{t+}\delta_{xx}w\right\|^2_{\mathfrak{D}} + \left\|\delta_{t+}\delta_{yy}w\right\|^2_{\mathfrak{D}}\right] \tag{2.147a}$$

$$\geq -\frac{k^2}{4}D\frac{(1-\nu)}{2}\frac{16}{h^4}\left[2\left\|\delta_{t+}w\right\|^2_{\mathfrak{D}} + 4\frac{\nu}{(1-\nu)}\left\|\delta_{t+}w\right\|^2_{\mathfrak{D}} + \left\|\delta_{t+}w\right\|^2_{\mathfrak{D}} + \left\|\delta_{t+}w\right\|^2_{\mathfrak{D}}\right] \tag{2.147b}$$

$$\geq -8D\frac{k^2}{h^4}\left\|\delta_{t+}w\right\|^2_{\mathfrak{D}}, \tag{2.147c}$$

where in the second line, the relations (2.111b), (2.111c) and (2.111d) have been used. Using this last expression, Eq. (2.146) can be rewritten as

$$\begin{aligned}\mathfrak{h}_{\text{rpl}} \geq \left(\frac{\rho H}{2} - 8D\frac{k^2}{h^4}\right)\left\|\delta_{t+}w\right\|^2_{\mathfrak{D}} + D(1-\nu)\left\|\mu_{t+}\delta_{x+,y+}w\right\|^2_{\mathfrak{D}} + \\ \frac{D}{2}\left[\nu\left\|\mu_{t+}\delta_{\Delta}w\right\|^2_{\mathfrak{D}} + (1-\nu)\left\|\mu_{t+}\delta_{xx}w\right\|^2_{\mathfrak{D}} + (1-\nu)\left\|\mu_{t+}\delta_{yy}w\right\|^2_{\mathfrak{D}}\right].\end{aligned} \tag{2.148}$$

A stability condition for the scheme can now be derived by requiring the non-negativity of the first term, that is

$$\frac{\rho H}{2} - 8D\frac{k^2}{h^4} \geq 0 \quad \Rightarrow \quad h^4 \geq 16\kappa^2 k^2. \tag{2.149}$$

This is equivalent to the condition (2.145) given above.

Accuracy of the finite difference scheme

The scheme (2.141) is second-order both in time and space, in other words the truncation error of the scheme is $\mathcal{O}(k^2) + \mathcal{O}(h^2)$. This, however, only holds in the hypothesis of reducing both the time and spatial steps at the same rate. In reality, the stability condition (2.145) requires that h be proportional to $\sqrt{k}$. Thus, the overall accuracy of the scheme reduces to a first-order in time [118]:

$$\mathcal{O}(k^2) + \mathcal{O}(h^2(k)) \overset{(2.145)}{=} \mathcal{O}(k^2) + \mathcal{O}(k) = \mathcal{O}(k). \tag{2.150}$$

In order to recover a second-order accurate scheme, Lambourg proposes the use of a fourth-order accurate spatial discretisation for the biharmonic operator [118]. This improvement, however, leads to a scheme that is twice more computationally expensive at the same sampling rate, and for which a numerical energy is difficult to derive.

2.4.4 Energy conservation

In the continuous case, the energy of the plate under lossless conditions is exactly conserved. A similar property can be verified for the discrete scheme as well, as is shown in this section. In the present case, however, the energy conservation holds to machine precision, with small variations which can be attributed to round off errors. More information can be found in Appendix B.

As a simple example, consider the lossless scheme (2.141) for the plate, initialised with a raised cosine initial deformation. In order to evaluate the energy properties of the scheme, the normalised energy variations $\mathfrak{e}^n$ will be considered. These are defined with respect to the initial energy $\mathfrak{h}^0$ as

$$\mathfrak{e}^n = \frac{\mathfrak{h}^n - \mathfrak{h}^0}{\lfloor \mathfrak{h}^0 \rfloor_2}, \tag{2.151}$$

where the quantity $\lfloor \mathfrak{h}^0 \rfloor_2 = 2^e \geq 0$ for some integer exponent e denotes the nearest power of two to $\mathfrak{h}^0$, rounding towards zero.

When the scheme is well designed, normalised energy variations $\mathfrak{e}^n$ are on the order of machine accuracy, which for double precision arithmetic is $\epsilon = 2^{-52} \approx 2.22 \times 10^{-16}$. This can be seen in Figure 2.12, where $\mathfrak{e}^n$ is plotted for a typical simulation with duration one second. It is apparent that energy jumps are quantised in multiples of ϵ. See also Appendix B for additional comments on normalised energy variations and round off errors.

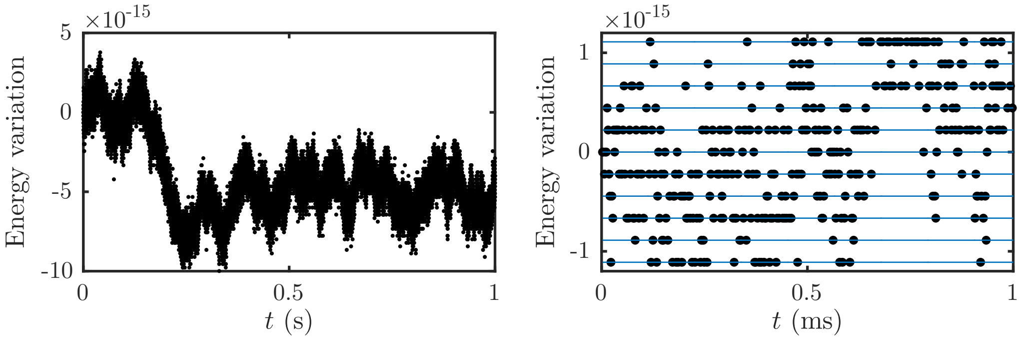


Figure 2.12: Left: Normalised energy variations for the plate with free boundary conditions under an initial raised cosine deformation over 1 s of output. Right: Zooming in on the initial portion of the graph reveals the machine precision quantisation of the energy jumps. Horizontal lines represent multiples of ϵ. The sampling rate is 44.1 kHz.

Linearity and numerical errors

The numerical scheme (2.141) inherits the linearity of the continuous equation (2.43). This property, however, does not exactly hold in general due to the presence of round off errors [103]. Consider, in fact, a square plate with $L = 0.5$ m initialised with a raised cosine deformation of amplitude 10 mm and diameter 10 cm applied at the centre. Figure 2.13 shows the normalised energy variations of a simulation where the initial amplitude is multiplied by 1, 2, and 3 times. Doubling the original amplitude does not affect the energy variations, which are exactly the same. This holds for any scaling which is a power of 2. Tripling the initial amplitude, however, reveals that a different pattern is produced, meaning that the outputs are not exactly scaled by 3. Linearity is broken at machine accuracy. This phenomenon can be explained by the presence of round off errors in the numerical representation of numbers. Multiplying by a power of 2, in fact, requires simply a change in the exponent in the floating point representation. Any other factor, instead, produces rounding that generates slightly different outputs.

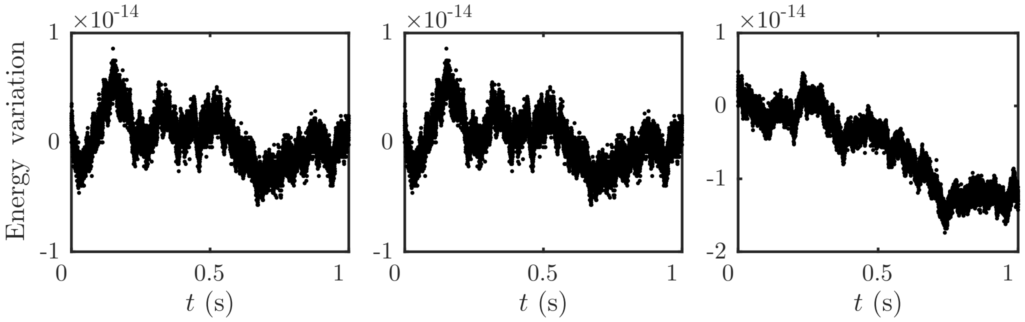


Figure 2.13: Linearity of the discrete plate equation. Normalised energy variations are plotted for scaling factors of 1, 2, and 3 times the initial amplitude (left, centre and right, respectively). Doubling the initial amplitude produces the same normalised variations as the original amplitude, as required by linearity. This property, however, is not exactly preserved when the excitation is three times the original value. Different errors, and thus different outputs, are produced. The sampling rate of the simulations is 44.1 kHz.

Normalised energy variations as random walks

Consider now the step-by-step normalised energy variations defined as follows

$$\mathfrak{e}_{\Delta}^{n+1/2} = \mathfrak{e}^{n+1} - \mathfrak{e}^{n}. \tag{2.152}$$

The quantity $\mathfrak{e}_{\Delta}^{n+1/2}$ is a measure of the energy jumps between two consecutive time steps. These energy variations for the simulation plotted in Figure 2.12 are presented in Figure 2.14. As can be seen, the values of $\mathfrak{e}_{\Delta}$ follow a Gaussian distribution centred around zero.

This observation seems to hint at an underlying random walk behaviour of the energy trajectories. Some recent works are available on this subject [197, 63], as this problem is extremely important in the long-term stability of numerical integration. This interpretation is based on the seminal work of Henrici [101] on round off errors as random variables, and can be traced back to the works of Brouwer on astronomical calculations [39].

Random walks in 1D [121, 155] are paths that can be defined in the following way. Starting from an arbitrary value (typically $x^0 = 0$), the value of x^n after n steps is given recursively by

$$x^n = x^{n-1} + \upsilon^n = x^0 + \sum_{j=1}^{n} \upsilon^j, \tag{2.153}$$

where $\upsilon^j \in \mathbb{Z}$ is a series of random values. A particularly simple case is that with $\upsilon^j = \pm 1$ for every j. The values of υ^j can also be defined by a Gaussian distribution, which is the scenario that is found in the present case. The quantities $\mathfrak{e}^n_\Delta$ introduced above then act as the variables υ^j in (2.153), and starting from the initial value $\mathfrak{e}^0 = 0$, they determine the trajectory plotted in Figure 2.12.

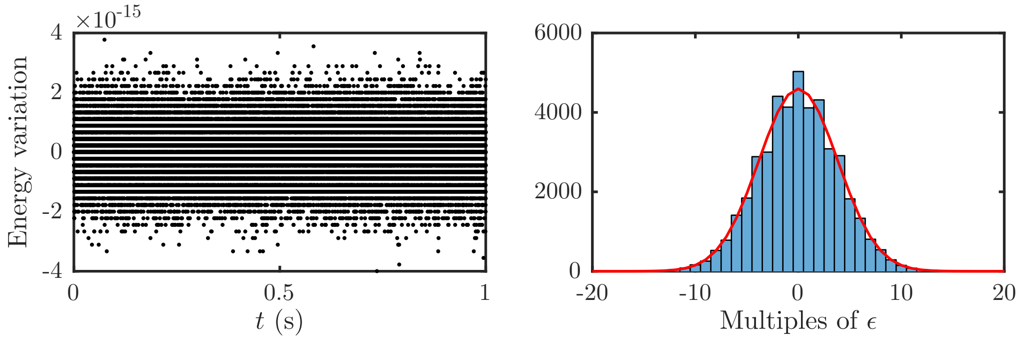


Figure 2.14: Left: Step-by-step normalised energy variations $\mathfrak{e}^n_\Delta$ for the simulation presented in Figure 2.12. Right: Histogram of the step-by-step energy variations in multiples of ϵ, $\mathfrak{e}^n_\Delta/\epsilon$. A Gaussian distribution of the random values is apparent, hinting at a random walk behaviour of the energy variations. The mean value of this set of data is -0.0004ϵ, while the standard deviation is 3.82ϵ. A Gaussian distribution (in red) with these parameters and scaled by the total number of time steps (44100) is superimposed to the graph.

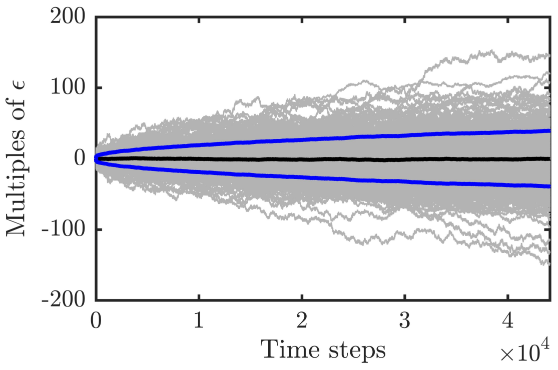


Figure 2.15: Achieving Brouwer's law with the finite difference scheme for the linear plate. Energy variations from 332 different simulations with random initial parameters are plotted in gray over 44100 time steps; the various trajectories average to zero, and such average is plotted with a bold black line; the standard deviation grows proportionally to $\sqrt{n}$, thus following Brouwer's law.

Achieving Brouwer's law

Looking at a single energy variation plot as that in Figure 2.12 could not be enough to determine if a numerical scheme behaves correctly, due to the statistical nature of round-off errors. More convincing conclusions can be drawn if several simulations are run with different initial parameters, as suggested by Hairer and other researchers [94, 197]. If the energy variations over various simulations average to zero, and their standard deviation grows with a rate proportional to $\sqrt{n}$, where n is the simulation time index, then Brouwer's law is satisfied, and the scheme behaves correctly.

Figure 2.15 shows the energy variations for 332 simulations with different initial parameters.

	Simply supported case, ζ/π^2									
Mode n.	**1**	**2**	**3**	**4**	**5**	**25**	**50**	**75**	**100**	$N_x \times N_y$
$\eta = 1.0$	2.00	4.99	4.99	7.99	9.97	39.57	71.64	103.66	138.31	51 × 51
	2.00	5.00	5.00	8.00	9.99	39.89	72.66	105.41	143.30	101 × 101
	2.00	5.00	5.00	8.00	10.00	39.97	72.91	105.85	144.57	201 × 201
	2.00	5.00	5.00	8.00	10.00	40.00	73.00	106.00	145.00	Theoretical
$\eta = 1.5$	1.44	2.78	4.44	4.99	5.77	25.62	48.66	69.84	92.01	51 × 76
	1.44	2.78	4.44	5.00	5.78	25.74	49.25	70.77	93.09	101 × 151
	1.44	2.78	4.44	5.00	5.78	25.77	49.40	71.03	93.35	201 × 301
	1.44	2.78	4.44	5.00	5.78	25.78	49.44	71.11	93.44	Theoretical
$\eta = 2.0$	1.25	2.00	3.25	4.24	4.99	19.91	36.58	52.21	71.13	51 × 101
	1.25	2.00	3.25	4.25	5.00	19.98	36.89	52.80	71.79	101 × 201
	1.25	2.00	3.25	4.25	5.00	19.99	36.97	52.95	71.95	201 × 401
	1.25	2.00	3.25	4.25	5.00	20.00	37.00	53.00	72.00	Theoretical

Table 2.2: Values of ζ/π^2 for a simply supported plate. The calculation is performed for various aspect ratios η. The modal frequencies ω_{mod} can be obtained by multiplying these numbers by $\kappa\pi^2$, see (2.154).

A square metal plate with thickness $H = 0.001$ m and $L_x = L_y = 0.5$ m has been initialised with a raised cosine deformation with random amplitudes, ranging over four orders of magnitude. The normalisation of energy variations obtained with (2.151) allows for a correct comparison between simulations with very different initial energies. It is possible to see how energy variations average to zero over a second of simulation (44100 time steps)—the value at the last time step is -0.06ϵ. The standard deviation grows proportionally to $\sigma\sqrt{n}$, with $\sigma = 0.19$.

2.4.5 Modal frequencies

As discussed in Section 2.2.4, analytic solutions for the biharmonic equation (2.83) are available only for the simply supported case. One possible way to calculate the modes and associated frequencies of the plate is by analysing the behaviour of the discrete biharmonic $\delta_{\Delta,\Delta}$ in its matrix representation $\mathbf{D}_{\Delta,\Delta}$. The modal frequencies ω_{mod} are related to the square root of the eigenvalues of the biharmonic operator by the following equation

$$\omega_{\mathrm{mod}} = \kappa\zeta, \qquad \zeta = \sqrt{\mathrm{eig}(\mathbf{D}_{\Delta,\Delta})}. \tag{2.154}$$

Once the eigenvalues are known, the modal shapes are the associated eigenmodes for the biharmonic matrix.

In the simply supported case, where analytical solutions exist, the modal frequencies given in closed form by Eq. (2.85) can be compared with the numerical simulation. The results are shown in Table 2.2, where the eigenvalues ζ are calculated for various aspect ratios $\eta = L_y/L_x$ and for different grid sizes. These are scaled by a factor of $1/\pi^2$, in order to be more easily compared with the analytical coefficient $[\frac{a^2}{L_x^2} + \frac{b^2}{L_y^2}]$ for $a, b = 1, 2, \ldots$, see Eq. (2.85). It is evident that the numerical discretisation produces slightly detuned frequencies, especially at large mode numbers. These, however, converge to the theoretical values in the limit of high grid density, as expected.

A similar comparison can be performed for the clamped and free plate, for which no analytical solution is available. The results are reported in Table 2.3 and 2.4, respectively. These values can be compared with those reported by Ducceschi in [73], which were obtained with a modal scheme and constitute an upper limit to the values of the exact modal frequencies.

Clamped case, non-centred conditions, ζ/π^2

Mode n.	**1**	**2**	**3**	**4**	**5**	**25**	**50**	**75**	**100**	$N_x \times N_y$
$\eta = 1.0$	3.51	7.15	7.15	10.54	12.80	44.28	77.29	109.85	144.90	51 × 51
	3.57	7.29	7.29	10.75	13.07	45.42	79.70	113.44	152.54	101 × 101
	3.61	7.36	7.36	10.86	13.20	45.94	80.71	114.93	154.93	201 × 201
	3.65	7.44	7.44	10.96	13.33	46.43	81.60	116.2	156.7	Ducceschi
$\eta = 1.5$	2.64	4.09	6.44	6.54	7.80	29.33	53.78	75.21	97.00	51 × 76
	2.69	4.16	6.57	6.64	7.94	30.05	55.39	77.44	99.40	101 × 151
	2.71	4.19	6.63	6.69	8.01	30.38	56.08	78.19	100.43	201 × 301
	2.74	4.22	6.70	6.74	8.09	30.69	56.69	78.89	101.40	Ducceschi
$\eta = 2.0$	2.40	3.12	4.41	6.23	6.26	22.09	40.79	57.22	74.45	51 × 101
	2.44	3.17	4.47	6.34	6.36	22.40	41.60	58.84	76.44	101 × 201
	2.47	3.20	4.50	6.38	6.42	22.55	42.05	59.54	77.14	201 × 401
	2.49	3.22	4.54	6.42	6.49	22.69	42.49	60.17	77.64	Ducceschi

Table 2.3: Values of ζ/π^2 for a clamped plate. The calculation is performed for various aspect ratios η. The modal frequencies ω_{mod} can be obtained by multiplying these numbers by $\kappa\pi^2$. Non-centred conditions are applied at the boundaries. Results are compared with those reported by Ducceschi [73] as upper limits.

Free case, non-centred conditions, ζ/π^2

Mode n.	**1**	**2**	**3**	**4**	**5**	**25**	**50**	**75**	**100**	$N_x \times N_y$
$\eta = 1.0$	1.30	1.91	2.36	3.35	3.35	20.67	48.45	73.88	101.81	51 × 51
	1.33	1.95	2.41	3.44	3.44	21.22	49.99	76.57	105.91	101 × 101
	1.35	1.97	2.43	3.48	3.48	21.52	50.71	77.82	107.71	201 × 201
	1.36	1.99	2.47	3.53	3.53	21.86	51.49	79.16	109.6	Ducceschi
$\eta = 1.5$	0.87	0.94	2.00	2.16	2.51	13.93	31.68	50.27	68.18	51 × 76
	0.89	0.95	2.04	2.20	2.56	14.19	32.37	51.65	70.41	101 × 151
	0.90	0.96	2.07	2.23	2.58	14.32	32.68	52.24	71.39	201 × 301
	0.91	0.97	2.09	2.25	2.60	14.50	33.11	52.92	72.46	Ducceschi
$\eta = 2.0$	0.53	0.65	1.43	1.48	2.14	11.14	23.73	37.65	51.16	51 × 101
	0.54	0.66	1.45	1.49	2.19	11.31	24.31	38.36	52.35	101 × 201
	0.54	0.67	1.47	1.50	2.21	11.39	24.60	38.65	52.81	201 × 401
	0.55	0.67	1.48	1.51	2.23	11.52	24.93	39.10	53.83	Ducceschi

Table 2.4: Values of ζ/π^2 for a free plate. The same considerations of the clamped case hold here, see the caption to Table 2.3. The first three rigid body modes of zero frequency are not reported.

Centred conditions

Consider the same meshing for the rectangular plate used before (see Figures 2.9 and 2.10). This time, however, centred conditions are applied at the boundaries. The eigenfrequencies ζ of the biharmonic operator can be recalculated for the clamped and free cases, and the results are collected in Tables 2.5 and 2.6. These appear to be in better agreement with those reported in the literature [124, 73]. Centred conditions, in fact, match better the actual area of the plate, while non-centred conditions overestimate it. It is always possible, however, to choose a different mesh, such that non-centred conditions align exactly with the boundary. This issue will not be discussed here.

Clamped case, centred conditions, ζ/π^2

Mode n.	**1**	**2**	**3**	**4**	**5**	**25**	**50**	**75**	**100**	$N_x \times N_y$
$\eta = 1.0$	3.64	7.41	7.41	10.92	13.25	45.64	79.43	112.60	148.35	51 × 51
	3.64	7.43	7.43	10.95	13.31	46.23	81.05	115.27	154.98	101 × 101
	3.65	7.43	7.43	10.96	13.33	46.38	81.46	115.97	156.30	201 × 201
	3.65	7.44	7.44	10.96	13.33	46.43	81.60	116.2	156.7	Ducceschi
$\eta = 1.5$	2.73	4.22	6.68	6.72	8.06	30.29	55.41	77.11	99.08	51 × 76
	2.74	4.22	6.69	6.74	8.08	30.59	56.37	78.52	100.76	101 × 151
	2.74	4.22	6.70	6.74	8.08	30.67	56.61	78.80	101.19	201 × 301
	2.74	4.22	6.70	6.74	8.09	30.69	56.69	78.89	101.40	Ducceschi
$\eta = 2.0$	2.49	3.22	4.53	6.40	6.46	22.53	41.85	58.86	75.79	51 × 101
	2.49	3.22	4.53	6.41	6.48	22.65	42.30	59.83	77.10	101 × 201
	2.49	3.22	4.54	6.42	6.48	22.68	42.45	60.08	77.50	201 × 401
	2.49	3.22	4.54	6.42	6.49	22.69	42.49	60.17	77.64	Ducceschi

Table 2.5: Values of ζ/π^2 for a clamped plate. The calculation is performed for various aspect ratios η. The modal frequencies ω_{mod} can be obtained by multiplying these numbers by $\kappa\pi^2$. Centred conditions (2.134) are applied at the boundaries.

Numerical dispersion

Frequency-domain analysis can be applied in order to find a dispersion relation analogous to (2.44) for the numerical scheme (2.141). By introducing the quantities $p_x = \sin^2(\beta_x h/2)$ and $p_y = \sin^2(\beta_y h/2)$, the numerical angular frequency ω_{num} produced by the scheme can be written as [19]

$$\omega_{\text{num}} = \pm \frac{2}{k} \sin^{-1} \left[2\mu \left(p_x + p_y \right) \right]. \tag{2.155}$$

Figure 2.16 shows the relative numerical dispersion as a function of $\boldsymbol{\beta}$, which is defined as

$$\Delta\omega = \frac{\omega_{\text{num}} - \omega_{\text{theor}}}{\omega_{\text{theor}}}, \tag{2.156}$$

where ω_{theor} is the theoretical frequency obtained from the dispersion relation (2.44) in the continuous case. It can be seen that waves along the diagonal and axial directions travel at different speeds. Notice that the numerical dispersion of the scheme is not to be confused with the original dispersive character of the continuous plate equation.

Free case, centred conditions, ζ/π^2

Mode n.	**1**	**2**	**3**	**4**	**5**	**25**	**50**	**75**	**100**	$N_x \times N_y$
$\eta = 1.0$	1.36	1.98	2.46	3.52	3.52	21.69	50.53	77.55	105.25	51 × 51
	1.36	1.98	2.46	3.52	3.52	21.79	51.18	78.63	108.50	101 × 101
	1.36	1.99	2.46	3.53	3.53	21.82	51.34	78.90	109.13	201 × 201
	1.36	1.99	2.47	3.53	3.53	21.86	51.49	79.16	109.6	Ducceschi
$\eta = 1.5$	0.90	0.96	2.09	2.24	2.59	14.37	32.56	51.80	70.33	51 × 76
	0.90	0.96	2.09	2.25	2.60	14.42	32.86	52.51	71.86	101 × 151
	0.90	0.96	2.09	2.25	2.60	14.44	32.94	52.69	72.17	201 × 301
	0.91	0.97	2.09	2.25	2.60	14.50	33.11	52.92	72.46	Ducceschi
$\eta = 2.0$	0.54	0.67	1.48	1.51	2.23	11.42	24.66	38.31	52.15	51 × 101
	0.54	0.67	1.48	1.51	2.23	11.45	24.82	38.74	52.92	101 × 201
	0.54	0.67	1.48	1.51	2.23	11.46	24.86	38.85	53.11	201 × 401
	0.55	0.67	1.48	1.51	2.23	11.52	24.93	39.10	53.83	Ducceschi

Table 2.6: Values of ζ/π^2 for a free plate. The same considerations of the clamped case hold here, see the caption to Table 2.5. The first three rigid body modes of zero frequency are not reported. Centred conditions (2.136) and (2.138) are applied at the boundaries.

The anisotropy of the numerical scheme has important perceptual effects. One possible way to ameliorate the behaviour of the Cartesian scheme is by using implicit schemes, which require the solution of a linear system in the update [19]. Another possibility is the use of hexagonal grids, which have proven to be more isotropic [98]. Both choices, however, come at larger computational costs.

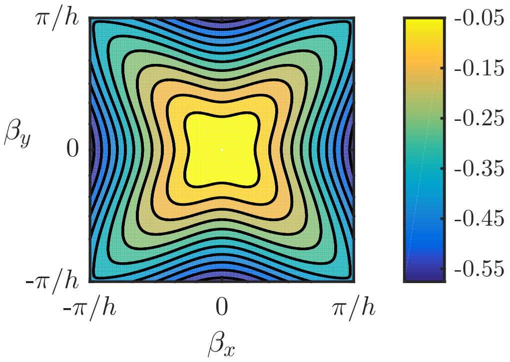


Figure 2.16: Relative numerical dispersion $\Delta\omega$ of the simple plate scheme (2.141). Each contour denotes a 5% error of the numerical frequency ω_{num} relative to the theoretical value given in Eq. (2.44).

Relation between numerical and modal frequencies

One interesting question that arises is what is the relation between the modal frequencies ω_{mod} defined in (2.154) and the numerical frequencies defined in (2.155). It is possible to show that the time integration performed with the scheme (2.141) causes the modal frequencies ω_{mod} to be mapped onto the numerical frequencies ω_{num} according to

$$\omega_{\text{num}} = \frac{2}{k} \sin^{-1}\left(\frac{k\omega_{\text{mod}}}{2}\right), \tag{2.157}$$

where k is the discrete time step. The numerical frequencies are those that effectively appear, e.g., in a Fourier spectrum of the simulated signal of the plate.

The values of ω_{num} become increasingly larger than the corresponding values of ω_{mod}. This can be seen in Figure 2.17, where the output spectrum of a numerical simulation is compared with the eigenfrequencies ω_{mod} of the system and with the numerical frequencies ω_{num}. In the upper part of the spectrum, eigenfrequencies and numerical frequencies split, with the latter being increasingly sharper.

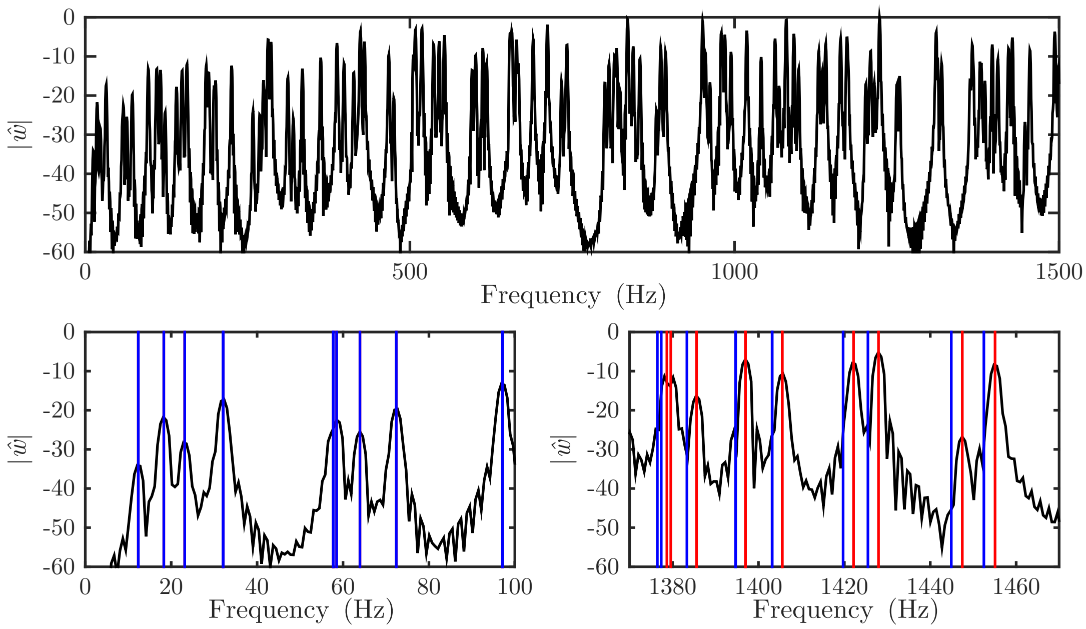
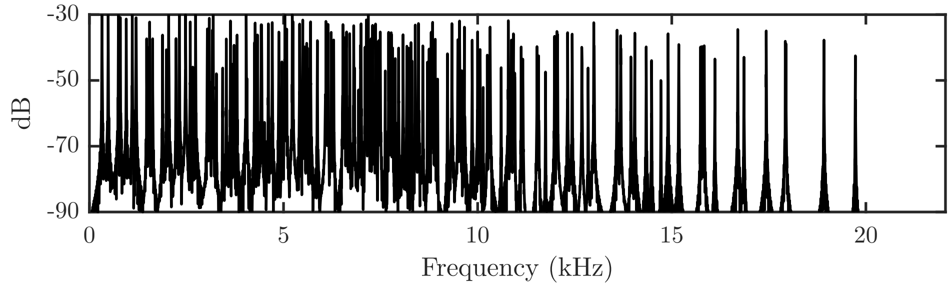


Figure 2.17: Top: Output spectrum for a square plate with free boundary conditions, initialised with a raised cosine deformation, over 1 s of simulation. The sampling rate is 44.1 kHz. Bottom left: Zoomed portion of the spectrum over the range 0-100 Hz. Modal frequencies ω_{mod} (blue) and numerical frequencies ω_{num} (red) almost coincide. Bottom right: Zoomed portion of the spectrum over the range 1370-1470 Hz. The splitting of the two frequencies becomes apparent, with the numerical scheme reproducing the numerical frequencies ω_{num}.

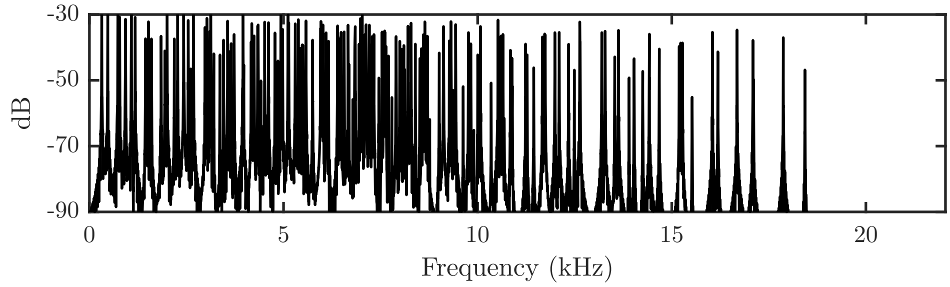
Bandlimiting effects away from the stability condition

The stability condition (2.145) dictates a minimum value h_{min} for the spatial grid step for a given time step k. In some circumstances, it might be tempting to choose a grid step $h \geq h_{\text{min}}$—this, in fact, determines a lower number of grid points, leading to a less computationally expensive scheme. This seemingly appealing strategy has however some drawbacks, the most important of which are bandlimiting effects.

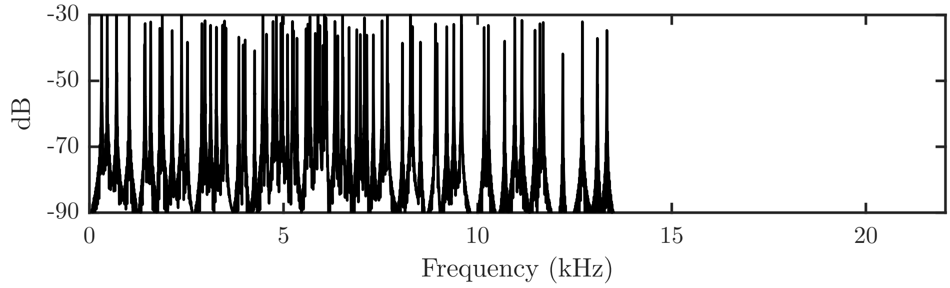
Consider to this end a rectangular steel plate with $L_x = 0.5$ m and $L_y = 0.3$ m, with thickness $H = 5$ mm and $\kappa = 0.77$, simulated at a sampling rate of 44.1 kHz. When the grid step $h = h_{\text{min}}$ is chosen according to (2.145), the output produced simulates frequencies up to the Nyquist frequency. When a value for h larger by 1% is chosen, the spectrum of the output obtained is considerably modified. Furthermore, an increase by 10% in the value of h produces an output that is bandlimited by more than 30%. The results of these simulations are plotted in Figure 2.18, where the various spectra are compared. For linear systems, it is therefore always wise to operate right at the stability of the scheme to achieve the best results in terms of output bandwidth.



(a) $h = h_{\min}$.



(b) $h = 1.01\,h_{\min}$.



(c) $h = 1.1\,h_{\min}$.

Figure 2.18: Output spectra from a plate simulation with different values of the grid step h. Top: h chosen at the stability condition (2.145). Centre: h by 1% larger than the stability limit. Bottom: h by 10% larger than the stability limit. The progressive bandlimiting effect or working away from the stability limit is apparent. A larger portion of the high frequency spectrum is lost.

2.4.6 Losses and tension

A finite difference approximation of the lossy plate equation (2.57) can be written as

$$\delta_{tt} w^n = -\kappa^2 \delta_{\Delta,\Delta} w^n - 2\sigma_0 \delta_{t\cdot} w^n + 2\sigma_1 \delta_{t\cdot} \delta_{\Delta} w^n, \tag{2.158}$$

where all the parameters have been introduced before. Notice that the time differences adopted are centred around $t = nk$, and it is possible to show that the stability condition (2.145) for the lossless plate remains unaltered. The energy balance for the system is

$$\delta_{t+}\mathfrak{h}_{\mathrm{rpl}}^{n+1/2} = -\mathfrak{q}^n, \quad \mathfrak{q}^n = 2\rho H\sigma_0 \left\|\delta_{t\cdot}w^n\right\|_{\mathfrak{D}}^2 + 2\rho H\sigma_1 \left(\left\|\delta_{t\cdot}\delta_{x-}w^n\right\|_{\mathfrak{D}}^2 + \left\|\delta_{t\cdot}\delta_{y-}w^n\right\|_{\mathfrak{D}}^2\right), \tag{2.159}$$

where $\mathfrak{h}_{\mathrm{rpl}}^{n+1/2}$ is the energy for the lossless plate (2.144). One major drawback of this scheme, though, is that it requires the solution of a linear system at every time step, given the presence of the Laplacian operator δ_{Δ} in the frequency dependent loss term. Given the small values of σ_0 and σ_1 that are generally adopted in musical acoustics, however, iterative methods can be successfully used in order to speed up the computation. These also give better results in terms of energy conservation, for the reasons outlined in Appendix B. The energy balance (2.159) implies that at every time step, the energy lost is equal to the loss term. In other words, a new balance similar to (2.61) can be derived at any time step N, which reads

$$\mathfrak{h}_{\mathrm{tot}}^{N+1/2} = \mathfrak{h}_{\mathrm{rpl}}^{N+1/2} + k\sum_{j=1}^{N}\mathfrak{q}^j = \mathfrak{h}_{\mathrm{rpl}}^{1/2}. \tag{2.160}$$

Here, the total energy in the simulation at time step $N + 1/2$, given by the sum of the energy of the plate and the accumulated dissipated power, is equal to the initial energy of the plate. Figure 2.19 shows the conservation of the total energy for a square plate under lossy conditions. The plate progressively loses energy, but the total stored energy in the system is conserved to machine accuracy. Details of the implementation, however, can have a decisive impact on the reduction of round off errors. See Appendix B for more details.

A simpler approach, that does not require the solution of a linear system, is to use a non-centred time difference in the frequency dependent loss term. The previous equation can be modified as

$$\delta_{tt}w = -\kappa^2\delta_{\Delta,\Delta}w - 2\sigma_0\delta_{t\cdot}w + 2\sigma_1\delta_{t-}\delta_{\Delta}w, \tag{2.161}$$

thus avoiding the matrix inversion. This requires the alteration of the stability condition, which now becomes

$$h \geq h_{\min} = \sqrt{4\sigma_1 k + \sqrt{(4\sigma_1 k)^2 + 16\kappa^2 k^2}}. \tag{2.162}$$

Finally, to conclude this section, the implementation of tension in the discrete case will be discussed. A finite difference scheme for the stiff membrane equation (2.62) can be written as

$$\rho H\delta_{tt}w = -D\delta_{\Delta,\Delta}w + T\delta_{\Delta}w, \tag{2.163}$$

where T is the tension applied at the rim of the membrane. The energy balance in this case reads

$$\delta_{t+}\mathfrak{h}_{\mathrm{sm}} = 0, \qquad \mathfrak{h}_{\mathrm{sm}} = \mathfrak{h}_{\mathrm{rpl}} + \frac{T}{2}\left(\langle\delta_{x-}w, \mathrm{s}_{t-}\delta_{x-}w\rangle_{\mathfrak{D}} + \langle\delta_{y-}w, \mathrm{s}_{t-}\delta_{y-}w\rangle_{\mathfrak{D}}\right) \tag{2.164}$$

where $\mathfrak{h}_{\mathrm{sm}}$ is the discrete energy for the stiff membrane. This balance is similar to the continuous

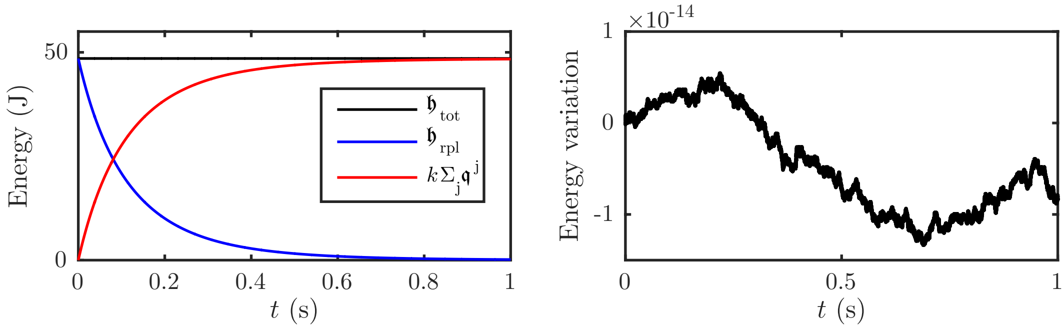


Figure 2.19: Left: Energy variation for the lossy plate. The energy for the plate (blue) decays with time, while the accumulated loss (red) increases. The total energy of the system (black) remains constant. Right: Normalised energy variations at machine accuracy. The simulation was run for a square plate with $L_x = L_y = 0.5$ m, $\kappa = 1.24$, $\sigma_0 = 1$ and $\sigma_1 = 0.0005$ and sampling rate of 44.1 kHz.

counterpart (2.63).

It is now useful to introduce the linear operator $\mathfrak{D}_m$, analogous to (2.64), which is defined as

$$\mathfrak{D}_m = -D\delta_{\Delta,\Delta} + T\delta_{\Delta} - 2\sigma_0\rho H\delta_{t\cdot} + 2\sigma_1\rho H\delta_{t-}\delta_{\Delta}. \tag{2.165}$$

The finite difference scheme for the linear stiff membrane can thus be written as

$$\rho H\delta_{tt}w = \mathfrak{D}_m w, \tag{2.166}$$

whose explicit update is given by

$$w^{n+1} = \frac{1}{(1+\sigma_0 k)}\Bigg[2w^n - w^{n-1} + \\ + \frac{k^2}{\rho H}\left(T\delta_{\Delta}w^n - D\delta_{\Delta,\Delta}w^n + \frac{\sigma_0}{k}\rho H w^{n-1} + 2\sigma_1\rho H\delta_{t-}\delta_{\Delta}w^n\right)\Bigg]. \tag{2.167}$$

The right hand side of the last equation can be condensed in a single term $\widetilde{w}$, which represents the quantities known from the previous time steps. This notation will be extensively used in the remainder of this work in order to present the matter in a more compact way. The stability condition for the scheme requires the minimum value of the grid step to be

$$h \geq h_{\min} = \sqrt{c^2k^2 + 4\sigma_1 k + \sqrt{(c^2k^2 + 4\sigma_1 k)^2 + 16\kappa^2k^2}}. \tag{2.168}$$

2.5 Simulation of circular plates

The simulation of circular plates is of paramount importance in many engineering applications. In this context, finite element methods [210] are generally adopted in order to solve the underlying differential equations of the problem. In some cases, this approach is used to calculate the modes of the system, thus obtaining a modal problem where the original PDEs are converted into a set of (possibly coupled) oscillators [68, 46]. In this work, however, the emphasis is given to finite difference methods, as they are easier to implement and require a relative lower amount of memory storage.

In this context, the natural approach for the simulation of circular plates would be a finite difference scheme in polar coordinates. As a matter of fact, polar schemes do exist [19], but they are hardly usable for sound synthesis purposes, given the severely bandlimited output that they generally produce. Part of the Author's work was concerned with finding alternative methods to simulate circular plates using finite difference approaches, possibly with the use of unstructured grids. One possible method has been devised and discussed in [98]—here, a finite volume implementation for a circular plate is presented, using square or hexagonal grids in the interior of the domain, and fitted cells at the boundary. This approach, however, only works for a plate with clamped boundary conditions, while no satisfactory method is available for simply supported and free conditions. For this reason, although novel, this methodology will not be presented in this work.

After discussing the limitations of finite difference schemes in polar coordinates, one alternative approach based on staircase approximations is presented in the remainder of this chapter. Although still unsatisfactory, the latter approach can serve as a first, simple and very efficient way to simulate circular plates.

2.5.1 Limitations of polar grids

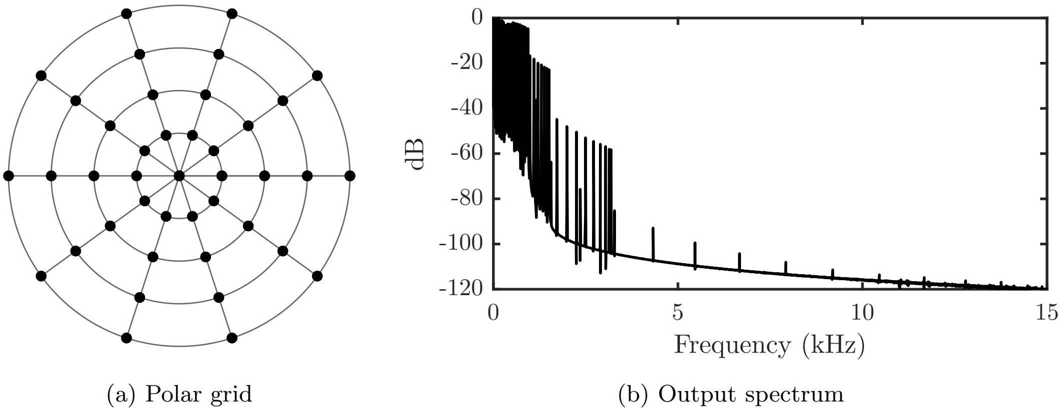
Polar grids are the first option that is considered when discretising a problem defined over a circular domain. Finite difference polar schemes are in fact available for the plate equation (see [19]), and require a grid with N_r and N_θ points along the radial and angular directions, respectively. A typical grid is plotted in Figure 2.20a. As can be easily seen, the spacing is not the same over the entire domain, but adapts to the geometry of the problem. In particular, the points become more and more separated in the angular direction as the distance from the centre is increased. Remembering the discussion on page 46 about the effects of increasing the spatial step of a grid, it is reasonable to expect that the outputs produced with polar grids be severely bandlimited.

Without dwelling upon the implementation details, which can be found in [19], Figure 2.20b shows the spectrum of the output of a circular plate simulation in polar coordinates obtained with an explicit scheme at a sampling rate of 44.1 kHz. The simulated plate has radius $R = 0.5$ m, thickness $H = 1$ mm and $\kappa = 1.5$. As can be seen, the frequencies produced are severely bandlimited, with only a few modes appearing above 5 kHz. As such, this scheme cannot be used in sound synthesis applications.

In order to address this problem, the use of implicit schemes in polar coordinates has been explored by Bilbao [19]. Although this approach can ameliorate the overall behaviour of the system, the distribution of the modes obtained does not match the theoretical values except at low frequencies. This fact and the higher computational cost associated with implicit schemes therefore discourages from using this approach for sound synthesis purposes.

2.5.2 Staircase approximation

One simple alternative approach in the simulation of a circular plate is the use of Cartesian coordinates over a staircase approximation of the domain. Starting from a square grid, as discussed in the previous sections, one can select only the points that lie within a given radius



(a) Polar grid

(b) Output spectrum

Figure 2.20: Left: Typical grid in polar coordinates, with $N_r = 5$ and $N_\theta = 10$ points. Notice the progressive increase in the angular grid spacing. Right: Output spectrum for a simulation at 44.1 kHz. Except for a few spurious modes, the frequencies produced are limited below 5 kHz.

R. The resulting jagged patch is a so-called staircase approximation to the circular region (see Figure 2.21). Although it is evident that refining the grid leads to increasingly accurate approximations to the domain, some numerical artefacts do appear, due to the irregularly terminated grid used for the simulation. An extensive numerical analysis of these errors for the wave equation can be found in the works of Cangellaris *et al.* [44] and of Häggblad *et al.* [93]. Furthermore, for plates in the simply supported case, some inconsistencies arise at the boundaries when a circular domain is approximated with a convex polygonal shape, a problem known as the Sapondzhyan-Babuška paradox [158, 9, 147, 59]. The limiting distribution in this case does not satisfy the boundary conditions in the continuous case.

Despite these limitations, staircase approximations are very attractive for the simplicity of the implementation—the algorithm is only slightly different from that of a rectangular plate. The efficiency of the resulting numerical code, therefore, outweighs the problems deriving from the presence of spurious artefacts, as will be seen shortly. It is clear, however, that this must only be considered as a first step towards the simulation of curved geometries, and more satisfactory solutions are desirable.

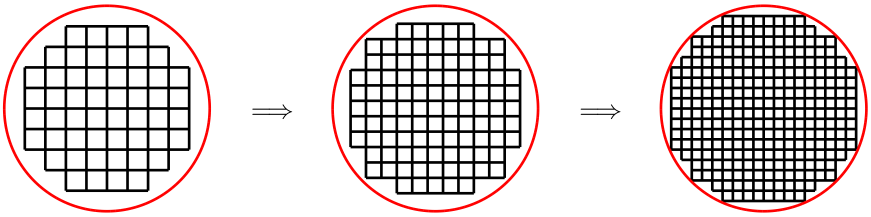


Figure 2.21: Staircase approximation for a circular region using Cartesian coordinates. The radius of the circle is $R = 1$ m, and the grid spacing is $h = 0.2$, $h = 0.15$ and $h = 0.1$ m from left to right.

Construction of a staircased grid

Consider a circular plate of radius R, and start by constructing a square grid of side $2R$, with a central point O. A staircased grid can be obtained by retaining only the points that lie within a radius R from O, and will be denoted by $\mathfrak{C}$. This domain near the boundary is plotted in Figure 2.22, where regular corners and reentrant corners can be seen. In the interior of the grid, nothing changes with respect to the Cartesian case, and the scheme (2.141) can be used. The boundary of this region is given by the set of points with one or two neighbours lying outside the circle. With this definition, reentrant corners, are considered interior points, with a regular update. Since the scheme is defined on a Cartesian grid, boundary conditions can be applied as in the case of the rectangular plate.

Both non-centred and centred conditions can be applied in the clamped case, leading to distinct modal frequencies for the system, which can be compared with the theory. The theoretical non-dimensional frequencies for the clamped case are reported in Table 2.1. Figure 2.23 shows the behaviour of the frequency of the first mode of a clamped plate under non-centred and centred conditions. Contrary to the rectangular case, a non-centred condition is more accurate, assuming that the meshing of Figure 2.21 be used in both cases. As a matter of fact, centred conditions approximate better the actual area of the jagged region. As this is already underestimating the actual area of the circle, centred conditions lead to higher frequencies than non-centred ones. In the free case, instead, only non-centred conditions lead to a stable scheme. In this case, in fact, when centred conditions are applied, instabilities arise at the reentrant corners. These can be explained with a finite volume interpretation of the finite difference scheme, as discussed in [96].

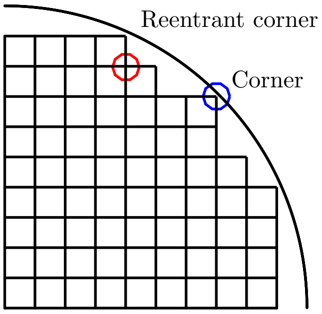


Figure 2.22: Zoomed portion of a staircase approximation of a circular region. A corner and a reentrant corner are indicated in blue and red, respectively.

Analysis of the numerical modes

Consider now the simulation for a circular metal plate with clamped conditions with the following parameters: $R = 0.25$ m, $H = 1$ mm, $\rho = 7800$ kg/m^3. At a sampling rate of 44.1 kHz, the corresponding grid has 43×43 points. The numerical frequencies produced by the staircased finite difference scheme are listed in Table 2.7, together with the corresponding theoretical values. The numerical frequencies of the scheme are obtained from the eigenvalues of the discrete

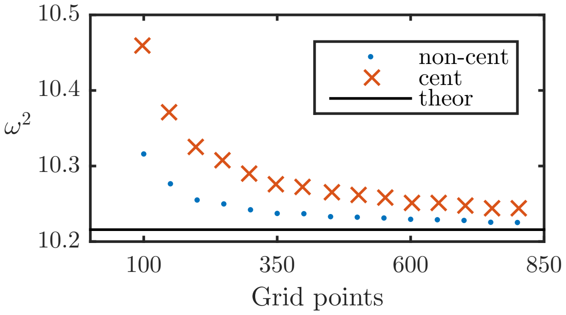


Figure 2.23: Comparison between non-centred (blue dots) and centred (red crosses) conditions for the first mode of a clamped plate. The theoretical frequency, as given in Table 2.1 is marked in black. Non-centred conditions are more accurate.

biharmonic operator and using (2.157):

$$f_{\text{num}} = \frac{1}{\pi k} \sin^{-1}\left(\frac{k\kappa\sqrt{\text{eig}\left(\mathbf{D}_{\Delta,\Delta}\right)}}{2}\right), \tag{2.169}$$

while the theoretical frequencies f_{theor} are calculated from the values for ζ^2 in Table 2.1 according to

$$f_{\text{theor}} = \frac{\kappa\zeta^2}{2\pi R^2}. \tag{2.170}$$

The deviation in cents between the numerical and theoretical values of the frequencies is also calculated.

It is possible to notice from the results that in some cases the degeneracy of the modes with one or more nodal diameters is broken at the numerical level. The identical frequencies predicted by the theory split into a doublet of peaks with slightly different values. This spurious splitting is a well known artefact created by the staircase approximation, and can be attributed to the not perfect symmetry of the domain along different diameters. The most pronounced effects can be observed for the modes with an even number of nodal diameters—the (2,0) and (6,0) modes, e.g., have a relative splitting of 13 and 30 cents, respectively. Another effect which can be noticed is the sign change in the frequency difference between theory and simulation. Contrary to what happens for the rectangular plates, where the frequency values are increasingly underestimated by the finite difference scheme (see Section 2.4.5), here the frequencies of the lowest modes are overestimated by the numerical algorithm. As has been already remarked, in fact, the area of the jagged domain is less than that of the circular domain. This behaviour changes between 750 Hz and 900 Hz. Above this value, the numerical dispersion dominates, and the scheme increasingly underestimates the real frequencies of the system. Increasing the simulation sampling rate has the effect of pushing this sign change towards higher frequencies.

As an additional check, it is possible to analyse the convergence of the various modes with increasing sampling rate. Figure 2.24 shows the behaviour of the difference in cents between the theoretical and numerical frequencies for the first 200 modes for the circular plate whose parameters are given in the previous paragraph. As already noticed from the results in Table 2.7, low frequency modes converge to the theoretical value from above, while at high frequency they converge from below. Regardless of this, however, it is possible to see that increasing the sampling rate progressively reduces the discrepancy between theoretical and simulated values for all the modes. At a sampling rate of 200 kHz, the differences fall between -20 and 20 cents, while at 600 kHz the variations range from -5 to 10 cents.

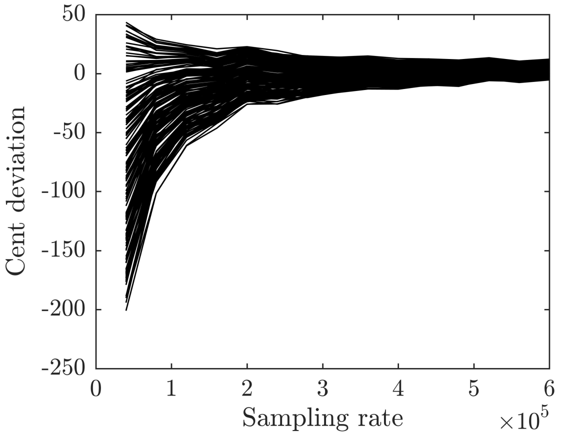


Figure 2.24: Convergence of the first 200 modes for a clamped, circular metal plate with $R = 0.25$ m and $H = 0.001$ m. The values for the simulated frequencies converge to the theoretical ones at high sampling rate. Convergence is from above at low frequencies, and from below at high frequencies.

Energy balance

An energy balance analogous to (2.128) can be written for the linear circular plate under a staircase approximation. The energy expression is similar to (2.124), provided the inner products are evaluated over the region $\mathfrak{C}$. Figure 2.25 shows energy conservation for a circular plate under clamped conditions, subject to an initial deformation. Energy variations in the lossless case are on the order of machine precision.

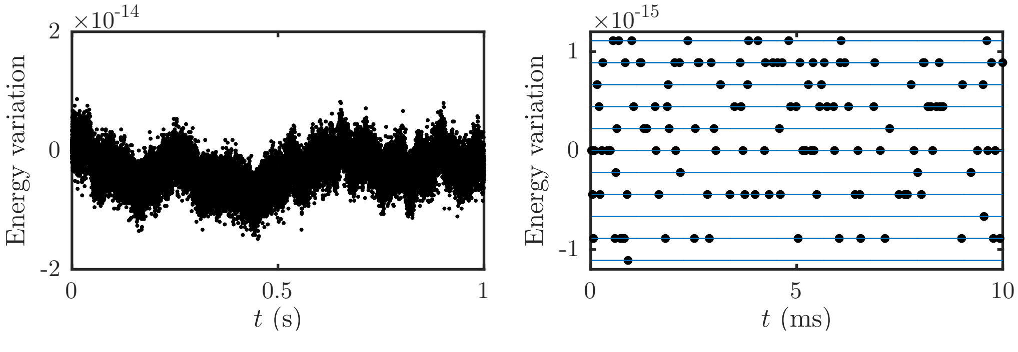


Figure 2.25: Left: Normalised energy variations for the circular plate with clamped boundary conditions under an initial raised cosine deformation over 1 second of output. Right: Zooming in on the initial portion of the graph reveals the machine precision quantisation of the energy jumps. Horizontal lines represent multiples of ϵ. The sampling rate is 44.1 kHz.

Mode (m, n)	**Theoretical**	**Numerical**	**Cent Dev.**
(0,0)	40.28	41.30	42.93
(1,0)	83.83	85.82	40.57
(1,0)	83.83	85.82	40.57
(2,0)	137.53	140.02	31.07
(2,0)	137.53	141.08	44.15
(0,1)	156.83	160.16	36.38
(3,0)	201.22	205.20	33.91
(3,0)	201.22	205.20	33.91
(1,1)	239.86	244.29	31.66
(1,1)	239.86	244.29	31.66
(4,0)	274.71	279.41	29.35
(4,0)	274.71	279.53	30.11
(2,1)	333.53	337.64	21.20
(2,1)	333.53	339.51	30.75
(0,2)	351.36	356.50	25.14
(5,0)	357.81	362.97	24.80
(5,0)	357.81	362.97	24.80
(3,1)	437.78	442.74	19.49
(3,1)	437.78	442.74	19.49
(6,0)	450.37	451.63	4.86
(6,0)	450.37	459.43	34.50
(1,2)	473.50	478.41	17.84
(1,2)	473.50	478.41	17.84
(7,0)	552.28	555.80	11.01
(7,0)	552.28	556.64	13.62
(4,1)	552.48	556.64	12.98
(4,1)	552.48	556.86	13.67
(2,2)	606.53	609.16	7.50
(2,2)	606.53	610.60	11.57
(0,3)	623.76	626.90	8.69
(8,0)	663.43	663.44	0.04
(8,0)	663.43	669.74	16.39
(5,1)	677.46	679.53	5.28
(5,1)	677.46	679.53	5.28
(3,2)	750.41	749.68	-1.69
(3,2)	750.41	749.68	-1.69
(9,0)	783.74	785.52	3.92
(9,0)	783.74	785.52	3.92
(1,3)	784.92	784.22	-1.53
(1,3)	784.92	784.22	-1.53
(6,1)	812.59	804.74	-16.80
(6,1)	812.59	816.93	9.22
(4,2)	905.05	899.96	-9.77
(4,2)	905.05	901.19	-7.41

Table 2.7: Comparison between theoretical and simulated frequencies for a clamped circular plate. The theoretical frequencies are as per Table 2.1, and are labelled according to the number of nodal diameters m and nodal circles n. The value for n does not include the nodal circle at the rim. The simulated frequencies are obtained according to Eq. (2.169) from the eigenvalues of the biharmonic operator constructed using a staircase approximation at a sampling rate of 44.1 kHz and for a plate with radius $R = 0.25$ m.

Final remarks

It is worth now weighing the advantages and disadvantages of the staircase approximation in the light of the analysis performed above. It should be clear that this approach offers a simple strategy to simulate circular plates, which is only slightly different from that of the rectangular plates. It is very easy, in fact, to create the discrete meshing, and no particular care must be taken to ensure the stability of the algorithm, which is inherited from the rectangular case. The presence of a numerical energy is an additional bonus that is useful when this model is coupled to other components (see, e.g., Sec. 6.4).

The weak point of such approach is the accuracy of the simulation, which suffers from the artefacts that have been described in the previous sections. In a sound synthesis framework, however, these may be tolerated in the light of the benefits outlined above, given also the already inharmonic character of the plates' sound. Should a better accuracy be required in the simulation, a higher sampling rate can always be used, as can be seen in Figure 2.24.

For these reasons, this approach will be used in the remainder of this work whenever the domain of interest is circular. It is clear, however, that such approximation is only made in the interest of computational efficiency, and that more work is required towards a more satisfactory solution of the problem.

Chapter 3

Nonlinear plate vibration

Nonlinearities play an important role in percussion instruments, and particularly in gongs and cymbals, even at normal playing conditions. Some experimental studies have been performed on the subject, highlighting the presence of subtle mechanisms of energy transfer between different frequency ranges leading, ultimately, to turbulence. Amongst the first of such studies in the context of musical acoustics, there is the work by Fletcher and Rossing, collected in their classic textbook [84]. In more recent years, the experimental and theoretical work of Touzé, Thomas and Chaigne [187, 174] has focussed on the analysis of gongs and cymbals within the mathematical framework of nonlinear plate and shell vibration. This topic is still a very active field of research [73].

Nonlinear models of plates and shells play are intensively studied in engineering, as these structures are ubiquitous, e.g., in vehicles and buildings design. A brief introduction on the applications of these models can be found in the textbooks by Nayfeh and Pai [137] and Amabili [4], together with a brief account of the historical development of the theory.

The most widely adopted model for nonlinear plate vibration is probably that due to von Kármán [198]. His contribution was to consider only the lowest nonlinear terms in the Green-Lagrange strain tensor $\boldsymbol{\epsilon}$, while retaining a linear stress-strain relation. The model that follows accounts for a nonlinear coupling between in-plane and transverse displacements ($\mathbf{u}$ and w, respectively), and is sometimes referred to as the full von Kármán model [209]. Another important contribution was given by Föppl [85] who, according to Bažant [11, p. 441], was responsible for introducing a stress function which simplifies the equations. The Föppl-von Kármán, or simple von Kármán model describes the behaviour of plates undergoing large amplitude displacements and moderate rotations when subject to static external loads. A similar model was arrived at by Herrmann in 1956 [102] in the dynamic case, and is sometimes referred to as the von Kármán evolution equations [61]. In 1954, a simplified model was proposed by Berger, in which the first invariant of the strain tensor is neglected [14].

Formally, the dynamic von Kármán plate can be described by a system of three PDEs with three variable functions (w and the two components of $\mathbf{u}$), involving several nonlinear terms [102]. When in-plane inertia is neglected and in-plane external forcing is absent, however, a simpler system can be written involving only two variables, the transverse displacement w and an auxiliary variable called Airy's stress function Φ [136], where the nonlinearity is introduced in a very simple and elegant way. This formulation has been validated against measurements with excellent results (see, e.g., [4, 176, 177]), and has thus been adopted in many applications.

This simplified approach will be presented in this work. The Berger model, instead, can be defined by a single equation that can be considered as the 2D extension of the Kirchhoff-Carrier equation for nonlinear string vibration [45].

In the present context of musical acoustics, both models have the ability to produce pitch glides, i.e., variations of the perceived pitch at high striking amplitudes, which are typical of the sounds of gongs and cymbals. The Berger nonlinearity has also been used in sound synthesis applications to describe tension modulation effects in drum membranes [7, 29]. The Berger model, however, is unable to reproduce the most dramatic features of the sound of gongs and cymbals, namely crashes and the migration of energy towards high frequencies, for which the von Kármán model is required.

From a numerical simulation point of view, several approaches have been used in the past. Finite element techniques [3, 144], pseudospectral methods [113] and symplectic methods [126] have been adopted. Modal methods have been successfully used for this problem recently [73]. Energy conserving finite difference methods for the simulation of the simplified [17] and full [25] von Kármán system have also been presented.

In this chapter, the first section will describe the model by Berger together with its numerical implementation, while a second, more extended section will be devoted to the von Kármán model. For the sake of completeness, nonlinear membranes will be presented as an extension of nonlinear plates in the third section.

3.1 The Berger model

A simplified model for nonlinear plate vibration was derived by Berger in 1954 under simplifying assumptions [14]. Over a domain $\mathcal{D}$, Berger's equation takes the form

$$\underbrace{\rho H \partial_t^2 w = -D\Delta^2 w - 2\sigma_0 \rho H \partial_t w}_{\text{linear}} \underbrace{+\mathfrak{T}\Delta w}_{\text{Berger}}. \tag{3.1}$$

The first part of the equation represents the transverse vibration w of a linear plate under frequency independent loss, as discussed in the previous chapter (see Eq. (2.57)). The last term represents the nonlinearity, with Δ the Laplacian operator and $\mathfrak{T}$, a scalar, autonomous nonlinearity, which is an averaged measure of the plate displacement over $\mathcal{D}$:

$$\mathfrak{T} = \xi \left\|\boldsymbol{\nabla} w\right\|_{\mathcal{D}}^2 = \xi \int_{\mathcal{D}} \left|\boldsymbol{\nabla} w\right|^2 \mathrm{d}\sigma, \qquad \xi = \frac{6D}{|\mathcal{D}|H^2}, \tag{3.2}$$

where $|\mathcal{D}|$ is the total area of the plate. Since the Berger term depends on the plate displacement, its contribution disappears in the limit of low amplitude (linear) vibrations. A loss term is included in the equation, such that the initial energy introduced by the blow is dissipated and the amplitude of plate vibration decreases. In this way, the influence of the nonlinear term is gradually attenuated, and a pitch glide effect takes place.

An energy balance for this model can be readily obtained by applying the techniques introduced in the previous chapter. Start by multiplying (3.1) by $\partial_t w$ and integrate over the domain:

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{pl}}}{\mathrm{d}t} = -\mathfrak{Q} + \mathfrak{B}_{\mathrm{pl}} + \mathfrak{T} \int_{\mathcal{D}} \partial_t w \, \Delta w \, \mathrm{d}\sigma, \tag{3.3}$$

where $\mathfrak{H}_{\mathrm{pl}}$ is the energy for the linear plate defined in (2.51), $\mathfrak{B}_{\mathrm{pl}}$ represents boundary conditions (2.49) and

$$\mathfrak{Q} = 2\sigma_0 \rho H \left\|\partial_t w\right\|^2 \tag{3.4}$$

is the loss term. The last term can be easily manipulated to give

$$\begin{aligned}
\mathfrak{T}\int_{\mathcal{D}} \partial_t w\, \Delta w \,\mathrm{d}\sigma &= -\mathfrak{T}\int_{\mathcal{D}} \partial_t \nabla w \cdot \nabla w \,\mathrm{d}\sigma + \mathfrak{T}\oint_{\partial\mathcal{D}} \partial_t w \,\mathbf{n}\cdot\nabla w \,\mathrm{d}s && \text{(3.5a)}\\
&= -\frac{\xi}{2}\left\|\nabla w\right\|_{\mathcal{D}}^2 \frac{\mathrm{d}}{\mathrm{d}t}\left\|\nabla w\right\|_{\mathcal{D}}^2 + \mathfrak{T}\oint_{\partial\mathcal{D}} \partial_t w \,\mathbf{n}\cdot\nabla w \,\mathrm{d}s && \text{(3.5b)}\\
&= -\frac{\xi}{4}\frac{\mathrm{d}}{\mathrm{d}t}\left\|\nabla w\right\|_{\mathcal{D}}^4 + \mathfrak{T}\oint_{\partial\mathcal{D}} \partial_t w \,\mathbf{n}\cdot\nabla w \,\mathrm{d}s && \text{(3.5c)}\\
&= -\frac{\mathrm{d}\mathfrak{U}_{\mathrm{Berger}}}{\mathrm{d}t} + \mathfrak{B}_{\mathrm{Berger}}, && \text{(3.5d)}
\end{aligned}$$

with contributions to the energy and boundary terms due to the Berger nonlinearity given by

$$\mathfrak{U}_{\mathrm{Berger}} = \frac{\xi}{4}\left\|\nabla w\right\|_{\mathcal{D}}^4, \qquad \mathfrak{B}_{\mathrm{Berger}} = \mathfrak{T}(t)\oint_{\partial\mathcal{D}} \partial_t w \,\mathbf{n}\cdot\nabla w \,\mathrm{d}s. \tag{3.6}$$

Thus, the final energy balance in terms of the total energy $\mathfrak{H}_{\mathrm{Berger}}$ can be written as

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{Berger}}}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\mathfrak{H}_{\mathrm{pl}} + \mathfrak{U}_{\mathrm{Berger}}\right) = -\mathfrak{Q} + \mathfrak{B}_{\mathrm{pl}} + \mathfrak{B}_{\mathrm{Berger}}. \tag{3.7}$$

A new boundary term is generated by applying integration by parts to the Berger term. When the plate is clamped or simply supported, however, this term disappears identically, due to the fact that $\partial_t w = 0|_{\partial\mathcal{D}}$ at the boundary. In the case of a free plate, however, boundary conditions (2.56) no longer hold. New conditions, which now depend on $\mathfrak{T}$ and are thus non-local, can be derived in this case in order to obtain an energy conserving scheme (see [19]). However, the validity of Berger's assumptions in the free case have been questioned by several researchers [139, 195], so its use in this case may not be completely justified. As this model will be used in this work primarily in conjunction with drum membranes, which are held fixed at the rim, this issue will not be discussed further.

3.1.1 Discrete approximation

Consider for the moment the infinite domain $\mathbb{Z}^2$. A finite difference scheme for (3.1) is discussed in [19], and can be written as

$$\rho H \delta_{tt} w^n = -D\delta_{\Delta,\Delta} w^n - 2\sigma_0 \rho H \delta_{t\cdot} w^n + \mathfrak{t}^n \delta_{\Delta} w^n, \tag{3.8}$$

where $\mathfrak{t}^n$ represents a discrete approximation to $\mathfrak{T}(t)$. Notice that the explicit time dependence of the scheme has been indicated for clarity. Several possibilities are available for the discretisation of $\mathfrak{t}^n$, but one is particularly useful as it leads to a stable scheme:

$$\mathfrak{t}^n = \xi\, \mu_{t+}\left(\langle \delta_{x-} w^n, s_{t-}\delta_{x-} w^n\rangle_{\mathbb{Z}^2} + \langle \delta_{y-} w^n, s_{t-}\delta_{y-} w^n\rangle_{\mathbb{Z}^2}\right). \tag{3.9}$$

Note that the presence of μ_{t+} makes the scheme implicit, in that $\mathfrak{t}^n$ depends on the unknown quantity w^{n+1}.

A discrete analogue of the energy balance (3.7) can be obtained by multiplying (3.8) by $\delta_{t\cdot}w$ and by taking an inner product over the domain:

$$\delta_{t+}\mathfrak{h}_{\mathrm{rpl}} = -\mathfrak{q} + \mathfrak{t}\,\langle \delta_{t\cdot}w, \delta_{\Delta}w\rangle_{\mathbb{Z}^2}. \tag{3.10}$$

Here, $\mathfrak{h}_{\mathrm{rpl}}$ and $\mathfrak{q}$ are the energy and loss power for the linear plate, as defined in (2.144) and (2.159). The last term in (3.10) can be manipulated in a similar way to the continuous case, giving

$$\begin{aligned}
\mathfrak{t}\,\langle \delta_{t\cdot}w, \delta_{\Delta}w\rangle_{\mathbb{Z}^2} &\overset{(2.109\mathrm{a})}{=} -\mathfrak{t}\left(\langle \delta_{x+}\delta_{t\cdot}w, \delta_{x+}w\rangle_{\mathbb{Z}^2} + \langle \delta_{y+}\delta_{t\cdot}w, \delta_{y+}w\rangle_{\mathbb{Z}^2}\right) && \text{(3.11a)}\\
&\overset{(2.107\mathrm{a})}{=} -\frac{\mathfrak{t}}{2}\,\delta_{t+}\left(\langle \delta_{x+}w, \mathrm{s}_{t-}\delta_{x+}w\rangle_{\mathbb{Z}^2} + \langle \delta_{y+}w, \mathrm{s}_{t-}\delta_{y+}w\rangle_{\mathbb{Z}^2}\right) && \text{(3.11b)}\\
&\overset{(3.9)}{=} -\frac{\xi}{4}\delta_{t+}\left(\langle \delta_{x+}w, \mathrm{s}_{t-}\delta_{x+}w\rangle_{\mathbb{Z}^2} + \langle \delta_{y+}w, \mathrm{s}_{t-}\delta_{y+}w\rangle_{\mathbb{Z}^2}\right)^2 && \text{(3.11c)}\\
&= -\delta_{t+}\mathfrak{u}_{\mathrm{Berger}}, && \text{(3.11d)}
\end{aligned}$$

where the potential $\mathfrak{u}_{\mathrm{Berger}}$ is defined as

$$\mathfrak{u}_{\mathrm{Berger}} = \frac{\xi}{4}\left(\langle \delta_{x+}w, \mathrm{s}_{t-}\delta_{x+}w\rangle_{\mathbb{Z}^2} + \langle \delta_{y+}w, \mathrm{s}_{t-}\delta_{y+}w\rangle_{\mathbb{Z}^2}\right)^2. \tag{3.12}$$

Thus, the discrete energy balance for the total energy $\mathfrak{h}_{\mathrm{Berger}}$ holds, with

$$\delta_{t+}\mathfrak{h}_{\mathrm{Berger}} = \delta_{t+}\left(\mathfrak{h}_{\mathrm{rpl}} + \mathfrak{u}_{\mathrm{Berger}}\right) = -\mathfrak{q}, \tag{3.13}$$

similarly to the continuous case.

Boundary terms

When the domain is limited, boundary conditions are generated by summation by parts. Consider, e.g., the discrete half-plane $\mathfrak{D}_{\mathbb{H}^x}$; the summation by parts in the first line of (3.11), performed with (2.109a), produces an additional term $\mathfrak{b}_{\mathrm{Berger}}$ evaluated at the boundary of the region

$$\mathfrak{b}_{\mathrm{Berger}} = \mathfrak{t}\{\delta_{t\cdot}w, \delta_{x-}w\}_{0,\mathbb{Z}}, \tag{3.14}$$

where the 1D inner product has been defined in (2.110). As in the continuous case, a fixed condition at the boundary (either clamped or simply supported) makes this extra term vanish.

3.1.2 Numerical implementation

Consider now a finite domain $\mathfrak{D}_{N_x,N_y}$. An efficient implementation of (3.8) can be obtained by writing the scheme in matrix-vector form. First, though, it is necessary to recast $\mathfrak{t}^n$ into a different form, as discussed in [19]. Using integration by parts, in fact, it is possible to write

$$\mathfrak{t}^n = -\frac{\xi}{2}\left(\langle w^{n+1}, \delta_{\Delta}w^n\rangle_{\mathfrak{D}_{N_x,N_y}} + \langle w^{n-1}, \delta_{\Delta}w^n\rangle_{\mathfrak{D}_{N_x,N_y}}\right). \tag{3.15}$$

Notice that boundary terms are produced in this case, too. As before, however, these are identically zero when fixed conditions are applied, such that they need not be considered. Now, express w^n and the Laplacian operator δ_Δ as vector and matrix, respectively, as discussed in Section 2.3.4, and let $\mathbf{q}^n$ be defined as

$$\mathbf{q}^n = hk\sqrt{\frac{\xi}{2}}\mathbf{D}_\Delta \mathbf{w}^n, \tag{3.16}$$

where $\mathbf{D}_\Delta$ is the matrix form of the Laplacian operator δ_Δ, see (2.116). Thus, the Berger term may be written in vector form as

$$\mathfrak{t}^n\, \delta_\Delta w^n \rightsquigarrow -\left[\left(\mathbf{q}^n \cdot \mathbf{w}^{n+1}\right)\mathbf{q}^n + \left(\mathbf{q}^n \cdot \mathbf{w}^{n-1}\right)\mathbf{q}^n\right]. \tag{3.17}$$

The additional factors h and k which appear in the definition of $\mathbf{q}^n$ are needed to take into account the factor h^2 that comes from the discrete inner product and the factor k^2 that is present in the operator δ_{tt}. The scheme (3.8) can now be written as

$$\mathbf{A}^n\mathbf{w}^{n+1} = \mathbf{B}\mathbf{w}^n - \mathbf{C}^n\mathbf{w}^{n-1}, \tag{3.18}$$

where the update matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ have this form

$$\mathbf{A} = (1+\sigma_0 k)\mathbb{1} + \mathbf{q}\mathbf{q}^T, \qquad \mathbf{B} = 2\mathbb{1} - \kappa^2 k^2 \mathbf{D}_{\Delta,\Delta}, \qquad \mathbf{C} = (1-\sigma_0 k)\mathbb{1} + \mathbf{q}\mathbf{q}^T. \tag{3.19}$$

Notice that $\mathbf{A}$ and $\mathbf{C}$ depend on the time step n, while $\mathbf{B}$ does not. The update of the scheme requires the solution of a linear system, with the matrix $\mathbf{A}$ depending on known values of $\mathbf{w}^n$ at the previous time step. Furthermore, $\mathbf{A}$ is symmetric positive definite. As such, existence and uniqueness of a solution is guaranteed. An exact expression for the inverse matrix $\mathbf{A}^{-1}$ is given by [160, 130]:

$$\mathbf{A}^{-1} = (1+\sigma_0 k)^{-1}\mathbb{1} - (1+\sigma_0 k)^{-1}\frac{\mathbf{q}\mathbf{q}^T}{1+\sigma_0 k + \mathbf{q}^T\mathbf{q}}. \tag{3.20}$$

Notice that $\mathbf{q}\mathbf{q}^T$ is a matrix, while $\mathbf{q}^T\mathbf{q}$ is a scalar. This expression leads to a particularly efficient implementation, since the $\mathbf{A}^{-1}$ need not be computed explicitly. When applied to a vector $\mathbf{u}$, in fact, the quantity $\mathbf{q}\mathbf{q}^T\mathbf{u}$ can be calculated by performing the vector-vector multiplication $\mathbf{q}^T\mathbf{u}$ first.

3.1.3 Energy conservation

Energy conservation for the Berger plate can be shown in the lossless case. Figure 3.1 shows energy variations in the total energy on the order of machine accuracy.

3.1.4 Pitch glide effect

As anticipated, the Berger model is able to produce pitch glides. An example of this is shown in Figure 3.2, where results obtained with a clamped square plate with $L = 0.5$ m, thickness 1 mm and $\kappa = 1.5$ for three different striking amplitudes are plotted. A long raised cosine centred at $x = (0.075, 0.025)$ m with duration 2 ms is used in all three the simulations. At low

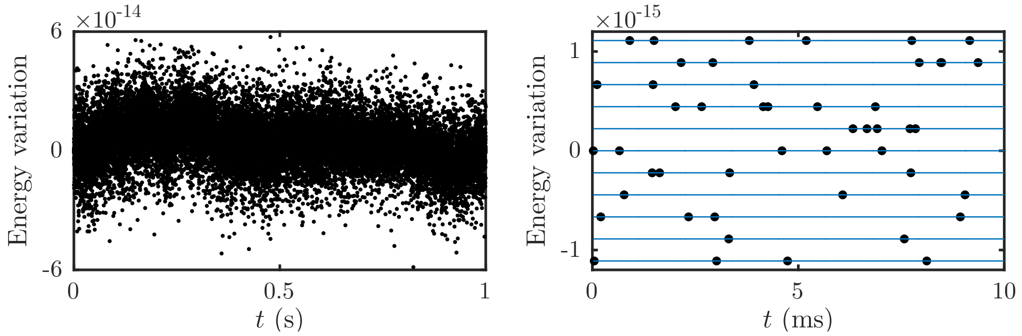


Figure 3.1: Left: Normalised energy variations for a rectangular Berger plate under lossless conditions, initialised with a raised cosine deformation. Right: Zooming in on the initial portion of the graph reveals the machine precision quantisation of the energy jumps. Horizontal lines represent multiples of ϵ. The sampling rate is 44.1 kHz.

excitation amplitudes (here, $F_{exc} = 50$ N), the system behaves as a linear plate. At moderate amplitudes ($F_{exc} = 300$ N), a weak pitch glide can be noticed. A larger effect can be achieved with increasing values of F_{exc}—at $F_{exc} = 600$ N, the frequency bending is evident.

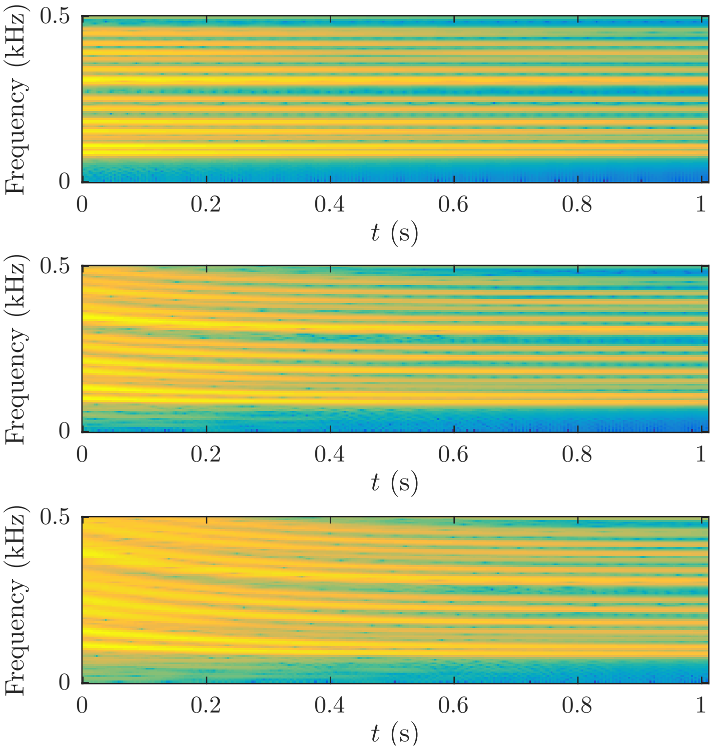


Figure 3.2: Pitch glide effect for the Berger plate. Spectrograms for 1 s of simulation for three different striking forces. Top: Low excitation amplitude ($F_{exc} = 50$ N). Centre: Moderate amplitude ($F_{exc} = 300$ N). Bottom: Large amplitude ($F_{exc} = 600$ N). Increasing values of F_{exc} produce a larger pitch glide, as expected. The sampling rate for the simulations is 44.1 kHz.

3.2 The von Kármán model

3.2.1 Theory

The derivation of the von Kármán equations is based on several hypotheses [175]:

1-4. The classical theory of thin plates is considered (see Section 2.2.1).

5. Only lower order nonlinear terms are considered in the Green-Lagrange strain tensor.
6. In-plane inertia is neglected.
7. The in-plane external forcing is neglected.

By hypothesis 5, the membrane strain tensor $\boldsymbol{\epsilon}$ in the Green-Lagrange strain tensor $\mathbf{E}$ (see (2.36)) is modified as follows

$$\boldsymbol{\epsilon} = \frac{1}{2}\left(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^T + \boldsymbol{\nabla}w \otimes \boldsymbol{\nabla}w\right), \tag{3.21}$$

where the last term represents the lowest nonlinear correction, while the curvature tensor $\boldsymbol{\kappa}$ (see (2.37)) remains the same. The Hookean stress-strain relation (2.38) remains valid, leading by integration of $\boldsymbol{\sigma}$ over the thickness of the plate to the new constitutive relation

$$\mathbf{N} = \frac{EH}{(1-\nu^2)}\,\phi_\nu(\boldsymbol{\epsilon}), \tag{3.22}$$

with $\mathbf{N}$ the membrane forces tensor and ϕ_ν the tensor mapping defined in (2.26).

By applying the variational principle [88], one can arrive at the following equations [174, 73]

$$\rho H \partial_t^2 \mathbf{u} = \nabla \cdot \mathbf{N} \tag{3.23a}$$

$$\rho H \partial_t^2 w = \nabla \cdot (\nabla \cdot \mathbf{M}) + \nabla \cdot (\mathbf{N}\boldsymbol{\nabla}w). \tag{3.23b}$$

When in-plane inertia is neglected, that is when $\partial_t^2\mathbf{u} = \mathbf{0}$ (hypothesis 6), the two equations become

$$\mathbf{0} = \nabla \cdot \mathbf{N} \tag{3.24a}$$

$$\rho H \partial_t^2 w = \nabla \cdot (\nabla \cdot \mathbf{M}) + \mathbf{N} : \boldsymbol{\nabla}\boldsymbol{\nabla}w. \tag{3.24b}$$

The usual approach now is to substitute $\mathbf{N}$ with the so-called Airy stress function, defined implicitly as

$$\Delta\Phi\,\mathbb{1} - \boldsymbol{\nabla}\boldsymbol{\nabla}\Phi = \mathbf{N}, \tag{3.25}$$

where $\mathbb{1}$ is the identity matrix. Inverting (3.22), it is easy to show that

$$\Delta\,\mathrm{tr}\,\boldsymbol{\epsilon} - \nabla \cdot (\nabla \cdot \boldsymbol{\epsilon}) = \frac{1}{EH}\Delta^2\Phi, \tag{3.26}$$

while, at the same time

$$\Delta\,\mathrm{tr}\,\boldsymbol{\epsilon} - \nabla \cdot (\nabla \cdot \boldsymbol{\epsilon}) = \frac{1}{2}\left(\boldsymbol{\nabla}\boldsymbol{\nabla}w : \boldsymbol{\nabla}\boldsymbol{\nabla}w - \Delta w \Delta w\right). \tag{3.27}$$

Finally, this leads to a simplified form of the system (3.23):

$$\rho H \partial_t^2 w = -D\Delta^2 w + \mathcal{L}(w, \Phi) \tag{3.28a}$$

$$\Delta^2 \Phi = -\frac{EH}{2}\mathcal{L}(w, w), \tag{3.28b}$$

which is the well known von Kármán system that is usually found in the literature [136, 4]. As anticipated, this is a system of two PDEs in the unknown scalar functions $w(\mathbf{x}, t)$ and $\Phi(\mathbf{x}, t)$. The nonlinearity is introduced by the bilinear operator $\mathcal{L}(\cdot, \cdot)$, which is sometimes referred to as the "Monge-Ampère form" or simply the "von Kármán operator". For any two functions f_1 and f_2, it is defined as

$$\mathcal{L}(f_1, f_2) = \Delta f_1 \Delta f_2 - \boldsymbol{\nabla}\boldsymbol{\nabla} f_1 : \boldsymbol{\nabla}\boldsymbol{\nabla} f_2. \tag{3.29}$$

More on this operator will be said in the next section.

3.2.2 $\mathcal{L}$ operator

As can be readily seen from the definition (3.29), the von Kármán operator $\mathcal{L}(f_1, f_2)$ is bilinear and symmetric, such that for any constant c one has

$$\mathcal{L}(cf_1, f_2) = c\mathcal{L}(f_1, f_2) = \mathcal{L}(f_1, cf_2), \qquad \mathcal{L}(f_1, f_2) = \mathcal{L}(f_2, f_1). \tag{3.30}$$

One further has

$$\partial_t \mathcal{L}(f_1, f_2) = \mathcal{L}(\partial_t f_1, f_2) + \mathcal{L}(f_1, \partial_t f_2), \tag{3.31}$$

that in the special case $f_1 = f_2 = f$ reduces to

$$\frac{1}{2}\partial_t \mathcal{L}(f, f) = \mathcal{L}(\partial_t f, f). \tag{3.32}$$

Another interesting property which was first noted in [17] for a rectangular plate, and extended to arbitrary domains in [175], is the "triple self-adjointness". This property will be useful in the following energy analysis, and states that for any three scalar functions f, g and r defined over $\mathcal{D}$, one has

$$\int_{\mathcal{D}} \mathcal{L}(f, g) r \, \mathrm{d}\sigma = \int_{\mathcal{D}} f \mathcal{L}(g, r) \, \mathrm{d}\sigma + \mathcal{J}, \tag{3.33}$$

where

$$\mathcal{J} = -\oint_{\partial\mathcal{D}} \mathbf{n} \cdot \left[(\Delta g \mathbb{1} - \boldsymbol{\nabla}\boldsymbol{\nabla} g)(f\boldsymbol{\nabla} r - r\boldsymbol{\nabla} f)\right] \mathrm{d}s. \tag{3.34}$$

By introducing $\mathbf{n}$ and $\boldsymbol{\tau}$, the normal and tangent unit vectors at the boundary, $\mathcal{J}$ can be written as

$$\mathcal{J} = -\oint_{\partial\mathcal{D}} \left[\Delta g - (\boldsymbol{\nabla}\boldsymbol{\nabla} g)_{nn}\right] (f\mathbf{n} \cdot \boldsymbol{\nabla} r - r\mathbf{n} \cdot \boldsymbol{\nabla} f) - (\boldsymbol{\nabla}\boldsymbol{\nabla} g)_{n\tau} (f\boldsymbol{\tau} \cdot \boldsymbol{\nabla} r - r\boldsymbol{\tau} \cdot \boldsymbol{\nabla} f) \, \mathrm{d}s, \tag{3.35}$$

with

$$(\boldsymbol{\nabla\nabla}g)_{nn} = \mathbf{n}\cdot(\boldsymbol{\nabla\nabla}g)\mathbf{n}, \qquad (\boldsymbol{\nabla\nabla}g)_{n\tau} = \mathbf{n}\cdot(\boldsymbol{\nabla\nabla}g)\boldsymbol{\tau}. \tag{3.36}$$

In the simple case when

$$f = 0 \text{ and } \mathbf{n}\cdot\boldsymbol{\nabla}f = 0 \quad \Longrightarrow \quad \boldsymbol{\tau}\cdot\boldsymbol{\nabla}f = 0 \tag{3.37}$$

at the boundary, then the line integral in (3.33) identically vanishes. This will be useful in ensuring the conservation of energy in the model.

3.2.3 Energy

Energy analysis for the nonlinear plate can be carried out in a similar manner to the linear case discussed in Section 2.2.2. Multiplying (3.28a) by $v = \partial_t w$ and integrating over the domain $\mathcal{D}$ gives

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{pl}}}{\mathrm{d}t} = \int_{\mathcal{D}} v\,\mathcal{L}(\Phi, w)\,\mathrm{d}\sigma + \mathfrak{B}_{\mathrm{pl}}, \tag{3.38}$$

where $\mathfrak{H}_{\mathrm{pl}}$ and $\mathfrak{B}_{\mathrm{pl}}$ are the energy and boundary terms for the linear case defined in (2.51) and (2.49), respectively. Continuing with the calculation, using the properties of the $\mathcal{L}$ operator, one has

$$\begin{aligned}
\frac{\mathrm{d}\mathfrak{H}_{\mathrm{pl}}}{\mathrm{d}t} &\overset{(3.33)}{=} \int_{\mathcal{D}} \Phi\,\mathcal{L}(v, w)\,\mathrm{d}\sigma + \mathfrak{B}_{\mathrm{pl}} + \mathfrak{B}_{\mathrm{nonlin}} \\
&\overset{(3.32)}{=} \frac{1}{2}\int_{\mathcal{D}} \Phi\,\partial_t\mathcal{L}(w, w)\,\mathrm{d}\sigma + \mathfrak{B}_{\mathrm{pl}} + \mathfrak{B}_{\mathrm{nonlin}} \\
&\overset{(3.28b)}{=} -\frac{1}{EH}\int_{\mathcal{D}} \Phi\,\partial_t\Delta^2\Phi\,\mathrm{d}\sigma + \mathfrak{B}_{\mathrm{pl}} + \mathfrak{B}_{\mathrm{nonlin}} \\
&= -\frac{1}{EH}\int_{\mathcal{D}} \Delta\Phi\,\partial_t\Delta\Phi\,\mathrm{d}\sigma + \mathfrak{B}_{\mathrm{pl}} + \mathfrak{B}_{\mathrm{nonlin}} + \mathfrak{B}'_{\mathrm{nonlin}} \\
&= -\frac{1}{2EH}\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{D}} |\Delta\Phi|^2\,\mathrm{d}\sigma + \mathfrak{B}_{\mathrm{pl}} + \mathfrak{B}_{\mathrm{nonlin}} + \mathfrak{B}'_{\mathrm{nonlin}}.
\end{aligned} \tag{3.39}$$

Finally, an energy balance of the form

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{vkp}}}{\mathrm{d}t} = \mathfrak{B}_{\mathrm{vkp}} \tag{3.40}$$

is recovered, with total energy $\mathfrak{H}_{\mathrm{vkp}}$ and boundary terms $\mathfrak{B}_{\mathrm{vkp}}$ defined as

$$\mathfrak{H}_{\mathrm{vkp}} = \mathfrak{H}_{\mathrm{pl}} + \frac{1}{2EH}\left\|\Phi\right\|^2_{\mathcal{D}}, \qquad \mathfrak{B}_{\mathrm{vkp}} = \mathfrak{B}_{\mathrm{pl}} + \mathfrak{B}_{\mathrm{nonlin}} + \mathfrak{B}'_{\mathrm{nonlin}}. \tag{3.41}$$

As is apparent, the total energy is a positive quantity, which is conserved when the boundary terms vanish.

Boundary conditions

Boundary conditions for Φ and w can be extracted by examining the boundary terms produced by the energy analysis. In particular, $\mathfrak{B}_{\mathrm{nonlin}}$ is similar to the term $\mathcal{J}$ in (3.35), with Φ playing the same role as f while w and v have the same role as g and r, respectively. The second term $\mathfrak{B}'_{\text{nonlin}}$ comes from integrating by parts the biharmonic operator:

$$\mathfrak{B}_{\text{nonlin}} = \oint_{\partial\mathcal{D}} \left[\Delta w - (\boldsymbol{\nabla\nabla} w)_{nn}\right](v\mathbf{n}\cdot\boldsymbol{\nabla}\Phi - \Phi\mathbf{n}\cdot\boldsymbol{\nabla} v) - \\ (\boldsymbol{\nabla\nabla} w)_{n\tau}(v\boldsymbol{\tau}\cdot\boldsymbol{\nabla}\Phi - \Phi\boldsymbol{\tau}\cdot\boldsymbol{\nabla} v)\,\mathrm{d}s, \quad (3.42)$$

$$\mathfrak{B}'_{\text{nonlin}} = -\frac{1}{EH}\oint_{\partial\mathcal{D}} \mathbf{n}\cdot(\Phi\boldsymbol{\nabla}\Delta\partial_t\Phi - \boldsymbol{\nabla}\Phi\Delta\partial_t\Phi)\,\mathrm{d}s. \quad (3.43)$$

A condition for Φ which is widely accepted in the literature to describe free in-plane vibration of the plate is mathematically expressed as

$$\Phi|_{\partial\mathcal{D}} = 0, \qquad \text{and} \qquad \mathbf{n}\cdot\boldsymbol{\nabla}\Phi|_{\partial\mathcal{D}} = 0, \quad (3.44)$$

at the boundary of the domain, which is in agreement with (3.37). These conditions allow both $\mathfrak{B}_{\text{nonlin}}$ and $\mathfrak{B}'_{\text{nonlin}}$ to vanish identically. No additional condition must be applied to w. It is to be noted that these are not the only possibility, and different conditions can lead to energy conservation [175]. Given their particularly simple expression, however, conditions (3.44) on Φ will be adopted throughout this work.

3.2.4 Rectangular plates

For a rectangular plate defined over $\mathcal{D}_{L_x,L_y}$, Cartesian coordinates are the natural choice. The expression for the von Kármán operator in this case becomes

$$\mathcal{L}(f,g) = \partial_x^2 f\partial_y^2 g + \partial_y^2 f\partial_x^2 g - 2\partial_x\partial_y f\partial_x\partial_y g, \quad (3.45)$$

for any two arbitrary functions $f(x,y,t)$ and $g(x,y,t)$.

Boundary conditions over, e.g., the side of length L_y at $x=0$ can be obtained from (3.42) and (3.43) by substituting $\mathbf{n}\cdot\boldsymbol{\nabla} = -\partial_x$ and $\boldsymbol{\tau}\cdot\boldsymbol{\nabla} = -\partial_y$:

$$\mathfrak{B}_{\text{nonlin}} = -\int_0^{L_y} \left(v\partial_y^2 w\partial_x\Phi + 2\partial_y v\partial_x\partial_y w\Phi + v\partial_x\partial_y^2 w\Phi - \partial_x v\partial_y^2 w\Phi\right)\Big|_{x=0}\mathrm{d}s, \quad (3.46)$$

$$\mathfrak{B}'_{\text{nonlin}} = \frac{1}{EH}\int_0^{L_y} \left(\Phi\partial_x\Delta(\partial_t\Phi) - \partial_x\Phi\Delta(\partial_t\Phi)\right)\big|_{x=0}\mathrm{d}s. \quad (3.47)$$

As is apparent, the conditions (3.44) become

$$\Phi = 0, \qquad \text{and} \qquad \partial_x\Phi = 0 \quad (3.48)$$

over the side, and they allow both boundary integrals above to be identically zero. Analogous conditions can be derived by symmetry over the other sides.

3.2.5 Finite difference scheme

A finite difference discretisation of the von Kármán equations has been proposed by Bilbao in [17]. There exists a family of different possible schemes, depending on two free parameters, which exhibit a diverse variety of behaviours in terms of performance and stability. All of them are implicit (i.e., they require a linear system inversion), given the elliptic character of (3.28b), which is due to the omission of the in-plane inertia terms.

One amongst all possible discretisations which are provably stable is the following:

$$\rho H \delta_{tt} w = -D\delta_{\Delta,\Delta} w + \mathfrak{l}(\mu_{t\cdot}\Phi, w) \tag{3.49a}$$

$$\mu_{t+}\delta_{\Delta,\Delta}\Phi = -\frac{EH}{2}\mathfrak{l}(w, \mathrm{s}_{t+}w), \tag{3.49b}$$

where $\mathfrak{l}$ is the discrete approximation of the von Kármán operator $\mathcal{L}$ defined, for any two grid functions f and g, as

$$\mathfrak{l}(f,g) = \delta_{xx} f \delta_{yy} g + \delta_{yy} f \delta_{xx} g - 2\mu_{x-,y-}(\delta_{x+,y+} f \delta_{x+,y+} g). \tag{3.50}$$

Similar properties to the continuous operator $\mathcal{L}$ can be proven for its discrete counterpart $\mathfrak{l}$ which will eventually lead to energy conservation.

3.2.6 $\mathfrak{l}$ operator and energy conservation

The discrete von Kármán operator (3.50) conserves the bilinearity of its continuous counterpart. As such, a property analogous to (3.32) can be derived using (2.107a):

$$\delta_{t+}\mathfrak{l}(f, \mathrm{s}_{t-} f) = 2\,\mathfrak{l}(f, \delta_{t\cdot} f). \tag{3.51}$$

The discrete approximation of (3.33) over a domain $\mathfrak{D}$ can be written as

$$\langle r, \mathfrak{l}(f,g)\rangle_{\mathfrak{D}} = \langle f, \mathfrak{l}(r,g)\rangle_{\mathfrak{D}} + \mathfrak{j}, \tag{3.52}$$

for any three grid functions f, g, and r, where $\mathfrak{j}$ represents the boundary terms. When $\mathfrak{D} = \mathfrak{D}_{\mathbb{H}^x}$, the semi-infinite plane with $x \geq 0$, this term is equal to

$$\begin{aligned}\mathfrak{j} = {} & \{f, \delta_{x-}(r\delta_{yy} g)\}_{x=0,\mathbb{Z}} - \{\delta_{x-} f, r\delta_{yy} g\}_{x=0,\mathbb{Z}} + \frac{1}{2}\{f, \mathrm{s}_{x-}\delta_{y-}(r\delta_{x+,y+} g)\}_{x=0,\mathbb{Z}} \\ & + \frac{1}{2}\{\mathrm{s}_{x-} f, \delta_{y-}(r\delta_{x-,y+} g)\}_{x=0,\mathbb{Z}} + \frac{1}{2}\{f, \mathrm{s}_{x-}\delta_{y+}(r\delta_{x+,y-} g)\}_{x=0,\mathbb{Z}} + \frac{1}{2}\{\mathrm{s}_{x-} f, \delta_{y+}(r\delta_{x-,y-} g)\}_{x=0,\mathbb{Z}}.\end{aligned} \tag{3.53}$$

Similarly to the continuous case, an energy balance for the discrete system can be arrived at by taking an inner product between (3.49a) and $\delta_{t\cdot} w$:

$$\delta_{t-}\mathfrak{h}_{\mathrm{rpl}}^{n+1/2} = \langle \delta_{t\cdot} w, \mathfrak{l}(\mu_{t\cdot}\Phi, w)\rangle_{\mathfrak{D}_{\mathbb{H}^x}}, \tag{3.54}$$

where $\mathfrak{h}_{\mathrm{rpl}}$ is the discrete energy for the linear plate defined in (2.124). Using the identities discussed above, it is straightforward to perform the following manipulations:

$$\begin{aligned}
\delta_{t-}\mathfrak{h}_{\mathrm{rpl}} &\overset{(3.52)}{=} \langle \mu_{t\cdot}\Phi, \mathfrak{l}(\delta_{t\cdot}w, w)\rangle_{\mathfrak{D}_{\mathbb{H}^x}} + \mathfrak{b}_{\mathrm{rpl}} + \mathfrak{b}_{\mathrm{nonlin}} \\
&\overset{(3.51)}{=} \frac{1}{2}\langle \mu_{t\cdot}\Phi, \delta_{t-}\mathfrak{l}(w, \mathrm{s}_{t+}w)\rangle_{\mathfrak{D}_{\mathbb{H}^x}} + \mathfrak{b}_{\mathrm{rpl}} + \mathfrak{b}_{\mathrm{nonlin}} \\
&\overset{(3.49b)}{=} -\frac{1}{EH}\langle \mu_{t\cdot}\Phi, \delta_{t\cdot}\delta_{\Delta,\Delta}\Phi\rangle_{\mathfrak{D}_{\mathbb{H}^x}} + \mathfrak{b}_{\mathrm{lin}} + \mathfrak{b}_{\mathrm{nonlin}} \qquad (3.55) \\
&= -\frac{1}{EH}\langle \delta_{\Delta}\mu_{t\cdot}\Phi, \delta_{t\cdot}\delta_{\Delta}\Phi\rangle_{\mathfrak{D}_{\mathbb{H}^x}} + \mathfrak{b}_{\mathrm{rpl}} + \mathfrak{b}_{\mathrm{nonlin}} + \mathfrak{b}'_{\mathrm{nonlin}} \\
&= -\frac{1}{2EH}\delta_{t-}\mu_{t+}\left\|\delta_{\Delta}\Phi\right\|^2_{\mathfrak{D}_{\mathbb{H}^x}} + \mathfrak{b}_{\mathrm{rpl}} + \mathfrak{b}_{\mathrm{nonlin}} + \mathfrak{b}'_{\mathrm{nonlin}}.
\end{aligned}$$

Therefore, an energy balance of the form

$$\delta_{t-}\mathfrak{h}^{n+1/2}_{\mathrm{vkp}} = \mathfrak{b}_{\mathrm{vkp}} \qquad (3.56)$$

is recovered, with

$$\mathfrak{h}^{n+1/2}_{\mathrm{vkp}} = \mathfrak{h}^{n+1/2}_{\mathrm{rpl}} + \frac{1}{2EH}\mu_{t+}\left\|\delta_{\Delta}\Phi^n\right\|^2_{\mathfrak{D}_{\mathbb{H}^x}}, \qquad \mathfrak{b}_{\mathrm{vkp}} = \mathfrak{b}_{\mathrm{rpl}} + \mathfrak{b}_{\mathrm{nonlin}} + \mathfrak{b}'_{\mathrm{nonlin}}. \qquad (3.57)$$

The nonlinear contribution to the energy is non-negative, so no modifications need to be made to the stability conditions derived in the linear case.

Boundary conditions for the Airy's function in the discrete case can be readily obtained from (3.53). By choosing

$$\Phi = 0, \quad \text{and} \quad \delta_{x-}\Phi = 0 \qquad (3.58)$$

over the boundary of the domain, $\mathfrak{j}$ identically vanish. Similar conditions can be derived over the other sides in the case of a rectangular plate.

3.2.7 Numerical implementation

In order to write the update equation for the scheme (3.49), first expand all time operators around the time step nk, and bring all unknown functions to the left hand side. This gives

$$w^{n+1} - \frac{k^2}{2\rho H}\mathfrak{l}(\Phi^{n+1}, w^n) = (2 - \kappa^2k^2\delta_{\Delta,\Delta})w^n - w^{n-1} + \frac{k^2}{2\rho H}\mathfrak{l}(\Phi^{n-1}, w^n), \qquad (3.59a)$$

$$\delta_{\Delta,\Delta}\Phi^{n+1} + EH\,\mathfrak{l}(w^{n+1}, w^n) = -\delta_{\Delta,\Delta}\Phi^n. \qquad (3.59b)$$

Notice that the scheme is implicit, as the biharmonic operator and the $\mathfrak{l}$ operator are applied to the unknowns. The latter, in particular, given any grid function w^n, can be considered as a function of a single variable, such that

$$\mathfrak{l}(\,\cdot\,, w^n) = \mathfrak{l}_{w^n}(\,\cdot\,). \qquad (3.60)$$

This newly defined operator inherits the linearity of the original, thus the implicit character of the scheme as a whole is that of a linear system. As is apparent, however, $\mathfrak{l}_{w^n}(\,\cdot\,)$ depends on w^n, and thus it must be recomputed at every time step. One can then foresee some complications in the implementation of the scheme.

Vectorised implementation

Let $\mathbf{w}$ and $\boldsymbol{\Phi}$ be the vector representations of the grid functions w_{l_x,l_y} and Φ_{l_x,l_y}, then concatenate them together in order to create a single vector $\mathbf{x}$, with

$$\mathbf{x} = \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\Phi} \end{bmatrix}, \tag{3.61}$$

that will serve as the combined unknown of the update. It is to be noted, however, that only the non-zero values of this vector must be retained for the calculations. The fixed conditions applied on $\boldsymbol{\Phi}$ imply that all the boundary points must be set to 0, and must then be removed from the linear system. The same holds for $\mathbf{w}$ under clamped and simply supported conditions. The matrices involved in the scheme must also be modified accordingly, by removing the unneeded rows and columns. Regardless of these details, the update for the scheme given by (3.59) can be written as a linear system of the form

$$\mathbf{A}^n_{\text{full}}\mathbf{x}^{n+1} = \mathbf{b}^n \tag{3.62}$$

where

$$\mathbf{A}^n_{\text{full}} = \left[\begin{array}{c|c} \mathbb{1} & -\boldsymbol{\Lambda}^n_{\boldsymbol{\Phi}} \\ \hline \boldsymbol{\Lambda}^n_{\mathbf{w}} & \mathbf{D}_{\Delta,\Delta} \end{array}\right], \quad \mathbf{b}^n \equiv \begin{bmatrix} \mathbf{b}^n_1 \\ \mathbf{b}^n_2 \end{bmatrix} = \begin{bmatrix} \mathbf{B}\mathbf{w}^n - \mathbf{C}\mathbf{w}^{n-1} + \boldsymbol{\Lambda}_{\boldsymbol{\Phi}}\boldsymbol{\Phi}^{n-1} \\ -\mathbf{D}_{\Delta,\Delta}\boldsymbol{\Phi}^n \end{bmatrix}. \tag{3.63}$$

Here, $\mathbf{A}^n_{\text{full}}$ is a square block matrix, with $\mathbb{1}$ the identity matrix and $\mathbf{D}_{\Delta,\Delta}$ the biharmonic operator, and $\mathbf{b}^n$ a vector of quantities known from the previous time step. The blocks $\boldsymbol{\Lambda}^n_{\boldsymbol{\Phi}}$ and $\boldsymbol{\Lambda}^n_{\mathbf{w}}$ are the matrix representations of the operator $\mathfrak{l}_{w^n}(\,\cdot\,)$, acting on $\boldsymbol{\Phi}$ and $\mathbf{w}$, respectively, together with the scaling constants:

$$\frac{k^2}{2\rho H}\mathfrak{l}_{w^n} \text{ acting on } \boldsymbol{\Phi} \longleftrightarrow \boldsymbol{\Lambda}^n_{\boldsymbol{\Phi}}, \qquad EH\mathfrak{l}_{w^n} \text{ acting on } \mathbf{w} \longleftrightarrow \boldsymbol{\Lambda}^n_{\mathbf{w}}.$$

Apart from the multiplicative coefficients, these operators are possibly different depending on the boundary conditions applied. Referring back to the bilinear form of the discrete von Kármán operator defined in (3.50), the new operator $\mathfrak{l}_{w^n}$ can be written as

$$\begin{aligned} \mathfrak{l}_{w^n}(\,\cdot\,) = \delta_{xx}w^n\delta_{yy}(\,\cdot\,) + \delta_{yy}w^n\delta_{xx}(\,\cdot\,) - \frac{1}{2}\big[&\delta_{x+,y+}w^n\delta_{x+,y+}(\,\cdot\,) + \\ &\delta_{x-,y+}w^n\delta_{x-,y+}(\,\cdot\,) + \delta_{x+,y-}w^n\delta_{x+,y-}(\,\cdot\,) + \delta_{x-,y-}w^n\delta_{x-,y-}(\,\cdot\,)\big]\,, \end{aligned} \tag{3.64}$$

where the last term in (3.50) has been expanded into four different terms. In order to cast this operator into matrix form, each discrete operator must be substituted with the corresponding matrix, and since different boundary conditions might in principle be applied to w and Φ, different matrices must be adopted.

Reduced matrix approach

A slightly different approach consists in exploiting the upper left identity block in $\mathbf{A}^n_{\mathrm{full}}$, solving for $\mathbf{w}^{n+1}$ and then obtaining a single equation in $\mathbf{\Phi}^{n+1}$:

$$\mathbf{w}^{n+1} = \mathbf{\Lambda}^n_{\mathbf{\Phi}}\mathbf{\Phi}^{n+1} + \mathbf{b}^n_1 \tag{3.65a}$$

$$\mathbf{A}^n_{\mathrm{rdcd}}\mathbf{\Phi}^{n+1} = \mathbf{b}^n_2 - \mathbf{\Lambda}^n_{\mathbf{w}}\mathbf{b}^n_1, \tag{3.65b}$$

where the new matrix $\mathbf{A}^n_{\mathrm{rdcd}}$ is given by

$$\mathbf{A}^n_{\mathrm{rdcd}} = \mathbf{D}_{\Delta,\Delta} + \mathbf{\Lambda}^n_{\mathbf{w}}\mathbf{\Lambda}^n_{\mathbf{\Phi}}, \tag{3.66}$$

and $\mathbf{b}^n_1$ and $\mathbf{b}^n_2$ are the known quantities defined in (3.63). Once (3.65b) is solved, $\mathbf{\Phi}^{n+1}$ can be inserted into (3.65a) in order to obtain the update for $\mathbf{w}^{n+1}$.

From a computational point of view, this method has the advantage that the system to be inverted is smaller, with half of the initial unknowns, in fact, and has been proposed by the Author in [183].

Iterative methods

As a further step in reducing the computation time of the code, different approaches to the solution of the reduced system (3.65b) can be sought. Direct methods, like Gaussian elimination or its variations, lead to a solution in a finite number of steps (generally on the order of N^3, where N is the size of the matrix). For large values of N, this computation becomes rapidly very expensive. One possible alternative to this approach is given by iterative methods [157]. Starting from an initial guess, the algorithm produces several successive solutions until some specified tolerance is reached. The convergence of such methods can be quite fast, especially when preconditioning is used, but is generally not guaranteed *a priori*. One then needs to test several types of iterative methods in order to find that with the best behaviour.

The field of iterative solution of linear systems is vast, and several different algorithms are available. In order not to interrupt the discussion on nonlinear plate vibration, the presentation of some of them will be delayed to the Appendix C. The interested reader is referred to [157] for a complete discussion.

In the present case, methods based on Krylov subspaces are particularly useful, and can speed up the computation considerably, as will be discussed in Section 3.2.10. The idea is to use the reduced implementation discussed before, and to solve (3.65b) with the Biconjugate Gradient Stabilised (BiCGSTAB) method [192]. At every time step, the algorithm produces a vector that is as close to the exact solution of the system as desired by the user, at the price of increasing the number of iterations required. To reduce the complexity of the calculation, a less stringent tolerance can be used (typically 10^{-9} instead of machine ϵ), in order to find an approximate solution that still retains the perceptual features of the exact numerical solution. Notice that this comes at the price of altering the energy conservation properties of the scheme.

One difficulty with methods based on Krylov subspaces is that the convergence of the algorithm is not guaranteed. This problem, however, can be reduced with the use of preconditioning (see Appendix C). In this case, a Cholesky decomposition of the biharmonic matrix $\mathbf{D}_{\Delta,\Delta}$ is the choice that gives the best results in terms of accuracy and computational speed.

Summary of the different methods

It is perhaps worth pausing the discussion for a moment to recapitulate the different approaches proposed to solve the linear system required in the implementation of the von Kármán scheme. Three methods have been discussed in this section. The first consists in solving the system (3.62) with a direct method, such as Gaussian elimination, obtaining simultaneously the update for w^{n+1} and Φ^{n+1}. This approach will be referred to as the full matrix implementation. A second strategy, labelled reduced matrix implementation, requires the system (3.65b) to be solved first in order to find the updated value for Φ^{n+1}, and then to substitute it back into (3.65a) in order to find w^{n+1}. In this case, the solution of the linear system is performed with a direct method. Finally, the iterative approach is similar to the reduced one, except that the solution of the linear system (3.65b) is found with an iterative method, typically BiCGSTAB. In this case, the user can specify the tolerance with which the system is solved.

These three methods have different properties which are analysed in the following sections.

3.2.8 Energy conservation

The three implementations described above show different properties in terms of energy conservation. The behaviour of the scheme, in fact, is very sensitive to the type of the linear system solution chosen.

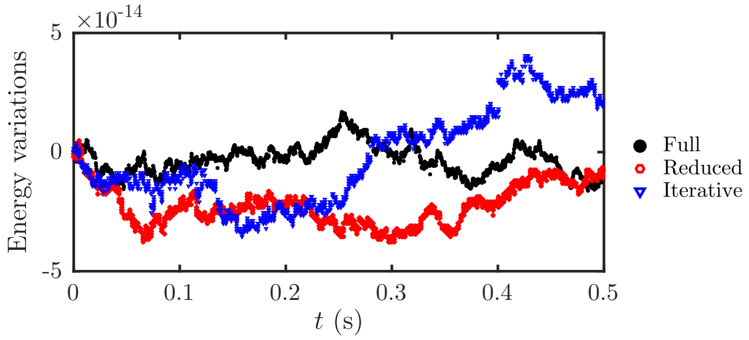
Consider, first of all, a square metal plate with thickness $H = 1$ mm, $\kappa = 1.5$ and $L_x = L_y = 0.3$ m under lossless conditions, initialised with a raised cosine impulsive excitation with velocity 0.01 m/s. The maximum displacement produced by the strike in this case is about the same as the thickness. Simulation results in terms of outputs and energy can be compared for the three different implementations described before. Figure 3.3a shows a comparison of the normalised energy variations for the three implementations. As is apparent, compatible patterns are produced in all cases.

Consider now the same plate, but initialised with a velocity of 0.5 m/s, corresponding to maximum displacement of about eleven times the thickness. In this case, the energy calculation starts to drift for the reduced and iterative implementations. Energy variation patterns are plotted in Figure 3.3b, with the iterative method exhibiting larger variations than the reduced method. This might be hinting at a poor convergence of these two implementations, and possibly to a tendency to larger numerical errors.

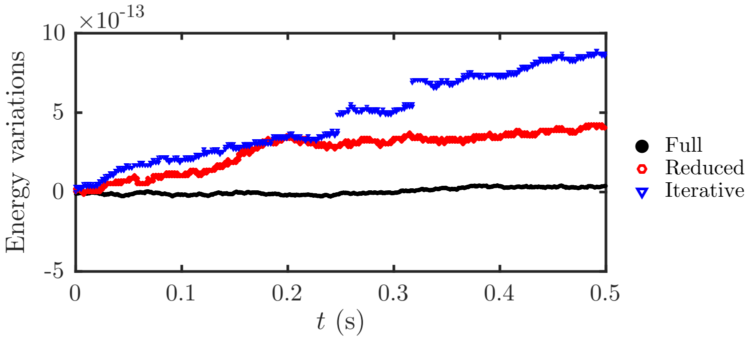
Statistical analysis of the energy

The behaviour of the three implementations may be explored in a statistical way over a wide range of displacement to thickness ratios. Consider therefore the normalised step by step energy variations $\mathfrak{e}_{\triangle}$ defined in (2.152) in terms of the normalised energy variations $\mathfrak{e}$. As discussed in Section 2.4.4, for a typical simulation the values of $\mathfrak{e}_{\triangle}$ should average to zero. Large variations in this expected behaviour could hint at a possible problem with the numerical scheme. The average $\text{avg}(\mathfrak{e}_{\triangle})$ over the entire simulation can thus be considered as a first parameter to analyse the behaviour of the implementations.

Figure 3.4 shows the results of this analysis over the range of output/thickness ratios between 0.1 and 100. A set of 76 random velocities have been chosen and used to initialise identical simulations with the three different implementations. Each simulation lasted 0.2 s (8810 steps



(a) Energy variations at low excitations.



(b) Energy variations at high excitations.

Figure 3.3: Comparison of the three implementations of the linear system for the von Kármán system in terms of energy conservation. Top: Plate excited with a raised cosine velocity of 0.01 m/s, corresponding to an output to thickness ratio of one. Energy variations are on the order of machine precision. Bottom: Plate excited with a raised cosine velocity of 0.5 m/s, corresponding to an output to thickness ratio of eleven. A linear drift can be noticed in the reduced and iterative patterns. The simulations were run with identical parameters for the plate and at a sampling rate of 44.1 kHz.

at a sampling rate of 44.1 kHz). The average variations $\mathrm{avg}(\mathfrak{e}_\triangle)$ in multiples of machine ϵ are plotted for each run as a function of the output to thickness ratio. Note that a correct normalisation of the energy variations is necessary in order to be able to compare the results of the simulations over a wide range of initial conditions.

As can be seen, the full implementation shows the least jitter in the energy average. The reduced and iterative implementations, instead, show a similar behaviour up to a ratio of ten, when they start to exhibit a clear drift. The reduced implementation, in particular, shows a negative drift between ratios of 10 and 35, while for larger ratios a positive drift can be noticed. A similar behaviour can be seen for the iterative implementation, but in this case much larger drifts can be observed.

From this discussion, it is clear that the full implementation exhibits the best behaviour over a wide range of output to thickness ratios. Even at ratios of a hundred, where the validity of the physical model itself is questionable, this numerical implementation is stable and produces good results. The reduced and iterative implementations, on the other hand, can be safely used up to ratios of ten, with analogous results to the full implementation. This can constitute a major advantage from the computational point of view, as will be explained shortly.

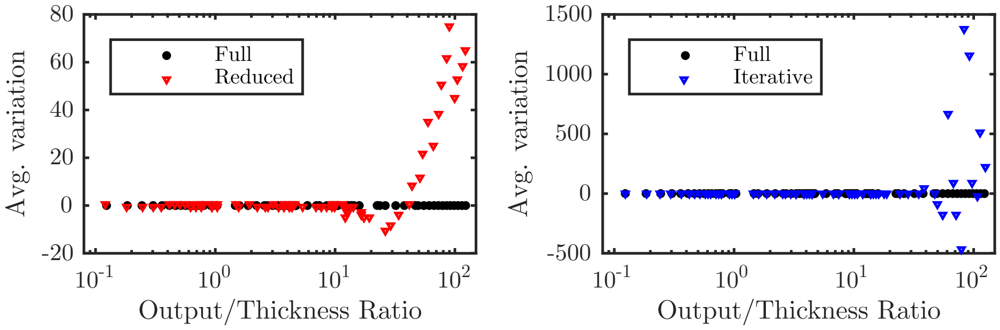


Figure 3.4: Average energy variations for the various implementations in multiples of ϵ. Left: Comparison between the full and reduced implementations. Right: Comparison between the full and iterative implementations. Larger variations can be seen for both the reduced and iterative implementations. The three approaches, however, give similar results for output to thickness ratios up to ten. The simulations were run at a sampling rate of 44.1 kHz and for 0.2 s.

3.2.9 Outputs

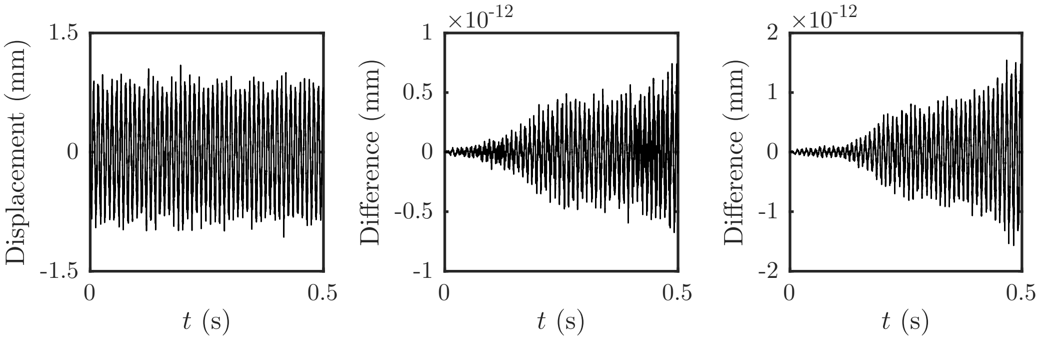
The outputs corresponding to the simulations of Figure 3.3 are plotted in Figure 3.5. At low excitation amplitudes (Figure 3.5a), the displacement at the centre of the plate obtained with the full implementation is on the order of the thickness, as anticipated. The differences between the output calculated with the full method and the other two approaches are also plotted. In this case, the differences are tiny, at least twelve orders of magnitude smaller than the amplitude of vibration.

Consider now the case of high amplitude excitations described in the previous section. The output for the full implementation is plotted in Figure 3.5b, together with the difference between this trajectory and those produced with the reduced and iterative implementations. As anticipated, the maximum displacement in this case is about eleven times the thickness. Here, the differences between the outputs calculated with the different methods can even become larger than the output itself!

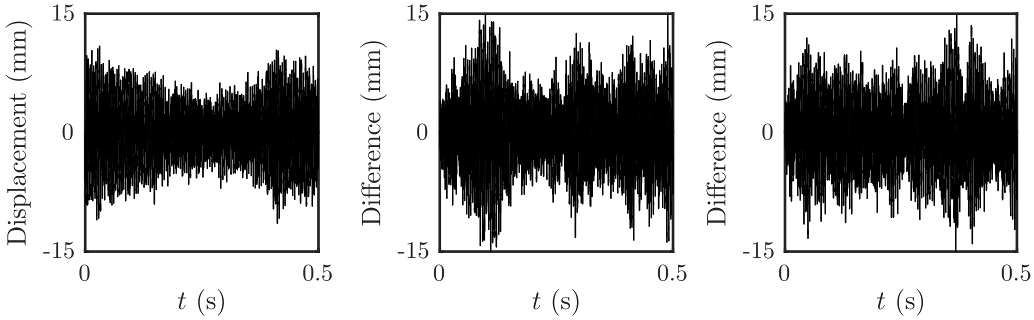
Despite the previous discussion on energy conservation for the three implementations, it is difficult to attribute the different outputs totally to numerical errors. The dynamics of the von Kármán model, in fact, is inherently turbulent. This means that small differences in the initial conditions produce trajectories that separate at a rate that is proportional to the inverse of the largest Lyapunov exponent of the system [189]. Therefore, as soon as small numerical variations are produced in the solution of the linear system, they start to increase, eventually producing totally different behaviours. The rate at which this happens depends on how strongly the nonlinearity is activated.

Musical acoustics simulations

The numerical issues broached in the previous sections, though important when it comes to numerical analysis, seem less urgent in sound synthesis applications. The discrepancy between the three implementations, in fact, is large when impulsive initial conditions are used, but when smoother excitations are adopted, the differences between the various outputs take more time to build up, and are not perceptually relevant.



(a) Low amplitude excitation.



(b) High amplitude excitation.

Figure 3.5: Comparison of the outputs obtained with the three different implementations at low and high excitation amplitudes. Left: Output obtained with the full implementation. Centre: Difference between the outputs calculated with the full and reduced implementations. Right: Difference between the outputs calculated with the full and iterative implementations. The sampling rate of the simulations was 44.1 kHz.

Consider a square steel plate with $L_x = L_y = 0.5$ m with thickness $H = 1$ mm, excited with a long raised cosine strike, as defined in Section 6.5.1. The outputs from such simulations are presented in Figure 3.6. It is possible to see that the differences between the two implementations are very limited. As a further step, the tolerance used in the iterative method can be dropped from the original machine ϵ in double precision to a less stringent value (for example 10^{-9}). In all three cases, the outputs generated are very similar—the differences, although present, are several orders of magnitude smaller than the signals themselves. The huge benefit of using the reduced implementation, however, comes from the increase in computation speed, which is illustrated in the next section.

3.2.10 Computational cost

The main drawback of the full implementation is its computational cost, with the solution of the linear system (3.62) being the dominant part of the code. The reduced and iterative implementations presented before can somehow alleviate this problem by speeding up the computation.

As a crude measure of the increase in computation speed obtained with the reduced and iterative implementations, one can compare the run times for identical simulations, differing only in the solution of the linear system. Table 3.1 shows the computation times for the simulations presented in Figure 3.6, i.e., the times required to produce 1 second of output at the sampling rate of 44.1 kHz. The full implementation is the most computationally expensive—it is 1.6

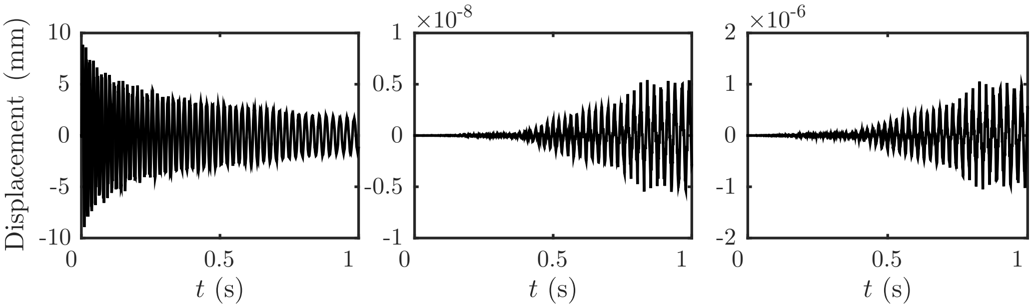


Figure 3.6: Comparison between the outputs obtained with the three implementations (full, reduced, and iterative with a tolerance of 1×10^{-9}). Left: Output obtained with the full implementation. Centre: Difference between out_{full} and out_{rdcd}. Right: Difference between out_{full} and out_{iter}. All the displacements are expressed in millimeters. Notice that the difference between the various outputs is several orders of magnitude smaller than the outputs themselves, thus justifying the use of the reduced and iterative implementations. All simulations are run for a square steel plate with $L_x = L_y = 0.5$ m and thickness 1 mm, under lossy conditions ($\sigma_0 = 1.34$, $\sigma_1 = 0.001$) and with a long raised cosine strike with duration 4 ms and force 2000 N.

times slower than the reduced implementation and more than 2 times slower than the iterative implementation. A tolerance of 10^{-9} has been used in the iterative solution.

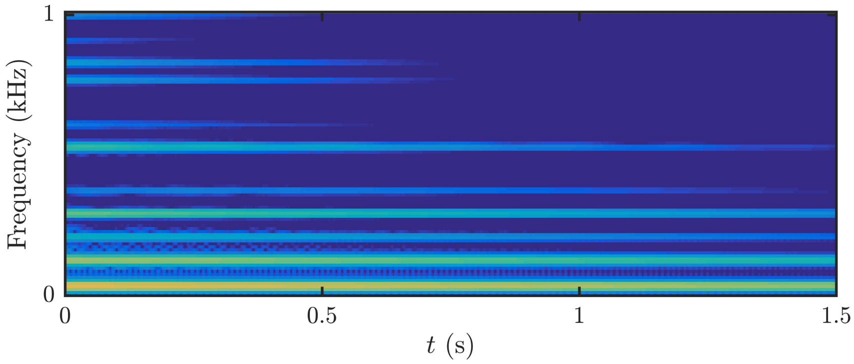
Although in some sense this is not a fair comparison, as the results obtained are different, it can nonetheless give an idea of the benefits of using the iterative method in musical acoustics applications. The advantage in terms of reduction of computation time becomes more and more pronounced with increasing state size of the system. This corresponds to a lower grid step, or equivalently to a higher sampling rate. Considerable savings of time can be obtained for nonlinear membranes, as discussed in [183]. See also Chapter 6.

	Full	**Reduced**	**Iterative**
Computation time	839 s	512 s	361 s
Speedup	×1	×1.6	×2.3

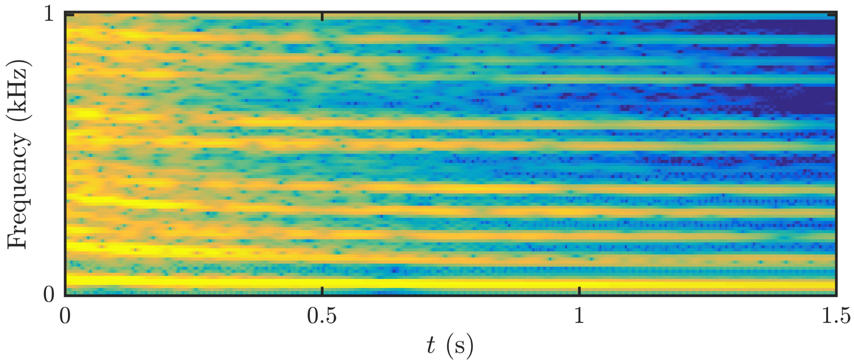
Table 3.1: Computation times for the simulations presented in Figure 3.6. The codes used are identical, except for the lines that concern the solution of the linear system. Speedups are calculated with respect to the full implementation. Simulations have been run on an Intel Core i7 CPU with processor speed of 2.8 GHz.

3.2.11 Pitch glide and crashes

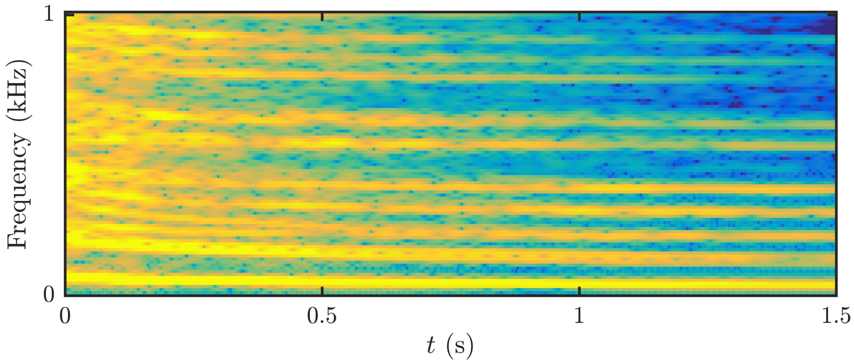
One of the most prominent features of the von Kármán model is the ability to produce pitch glides and crashes at high excitation amplitude. Pitch glides have already been discussed in the Berger case (see Section 3.1.4), and a similar phenomenon can be seen in Figure 3.7. Here, an external force f_{ext} with a raised cosine shape (see Section 6.5.1) has been adopted for the excitation of a square steel plate with $L_x = L_y = 0.5$ m, thickness $H = 1$ mm and $\kappa = 1.5$. The duration of the strike is 4 ms in all cases, whereas the maximum force $F_{\max}$ is as indicated in the captions. Pitch glides are not the only effect that is produced by von Kármán equations. At large excitation amplitudes, a migration of energy towards the high frequencies is activated, which produces crashes typical of gongs and cymbals. This is apparent from the spectrograms of Figure 3.8, which refer to the same simulations discussed above.



(a) $F_{\max} = 10$ N.



(b) $F_{\max} = 200$ N.

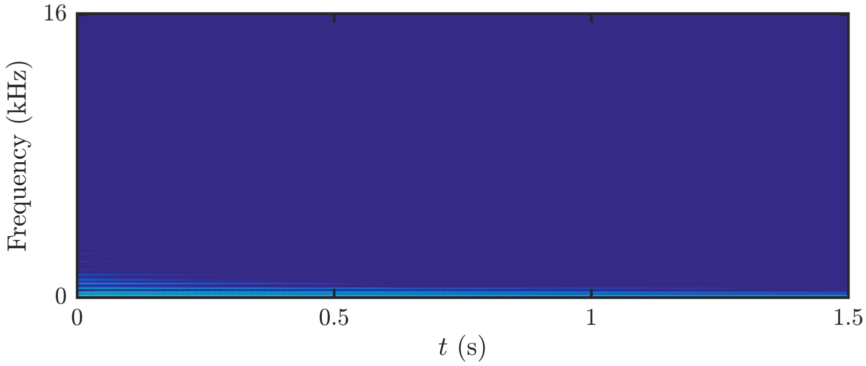


(c) $F_{\max} = 1000$ N.

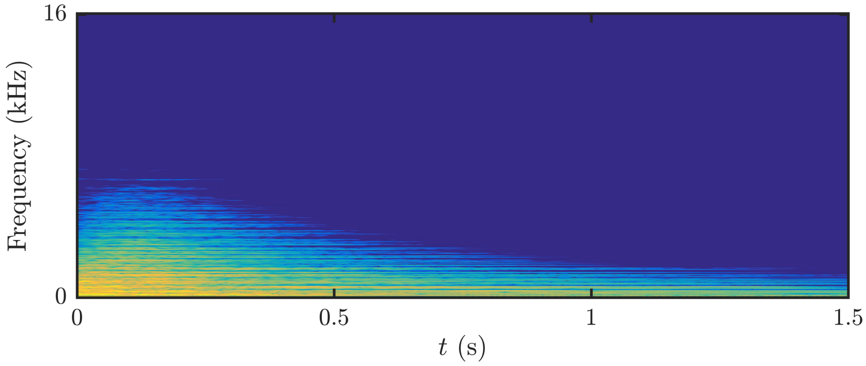
Figure 3.7: Spectrograms for the outputs of a square steel plate with von Kármán nonlinearity at excitation amplitudes as indicated. Pitch glide effects can be noticed at high excitation amplitudes. The simulation sampling rate is 44.1 kHz.

3.2.12 Quasi-periodic and turbulent behaviour

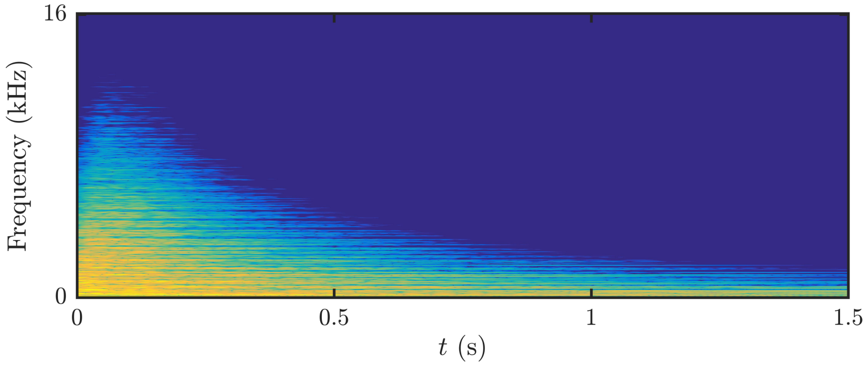
One of the most interesting features of the von Kármán model is the richness of phenomena that it can produce. Fletcher and Rossing [84] studied the vibration of cymbals subject to a harmonic excitation, and they reported various behaviours depending on the amplitude of the signal injected. More complex phenomena, ultimately involving turbulent behaviour, have



(a) $F_{\max} = 10$ N.



(b) $F_{\max} = 200$ N.



(c) $F_{\max} = 1000$ N.

Figure 3.8: Spectrograms for the same simulations as Figure 3.7, highlighting crashes phenomena at increasing excitation amplitude.

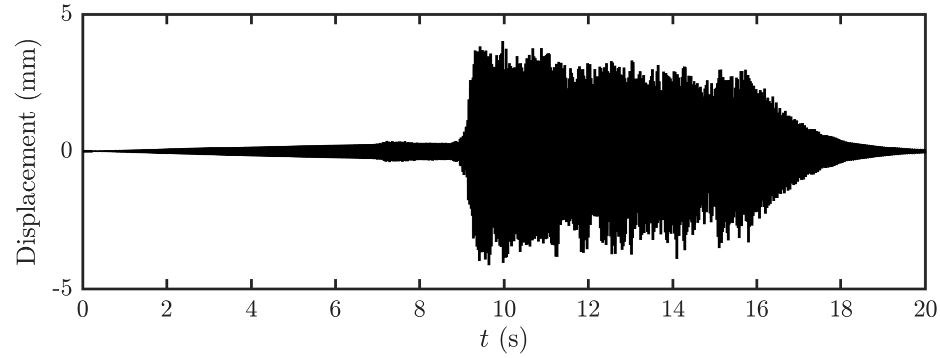
been observed experimentally by Touzé, Thomas, Chaigne *et al.* [187, 54, 190], with bifurcations appearing at particular frequencies and for particular amplitudes of the injected signal.

Numerical simulations performed with the scheme presented before are able to reproduce the kind of phenomena demonstrated by the experiments. Similar studies have been reported before in [188]. Recent works [43, 73] based on modal methods are also available. Consider, for example, a plate excited with a sinusoidal force with frequency f_{exc} and increasing amplitude A_{exc}. The response of the plate depends on the force applied, and generally falls into three different regimes: periodic, quasi-periodic and turbulent. At very low amplitudes, the system behaves almost linearly, and a single frequency (the excitation) appears in the spectrum. A weakly nonlinear behaviour is soon noticeable, which is characterised by spectral lines at odd multiples of f_{exc}. This behaviour is attributed to the cubic nonlinearity that appears in the von Kármán equations [188]. When the excitation is increased, more energy is available, which can then be shared between different modes that have particular relations between their frequencies. A bifurcation to a quasi-periodic regime is said to happen at this stage. Finally, at larger amplitudes, a turbulent regime is reached, where the entire spectrum of the plate is activated. At this point, it is difficult to distinguish single peaks, and more complex mathematical techniques must be applied in order to extract information from the spectra [187]. There are however some exceptions to this general behaviour. In some circumstances, especially at low f_{exc}, a direct transition to turbulence can be observed, without the quasi-periodic regime [188].

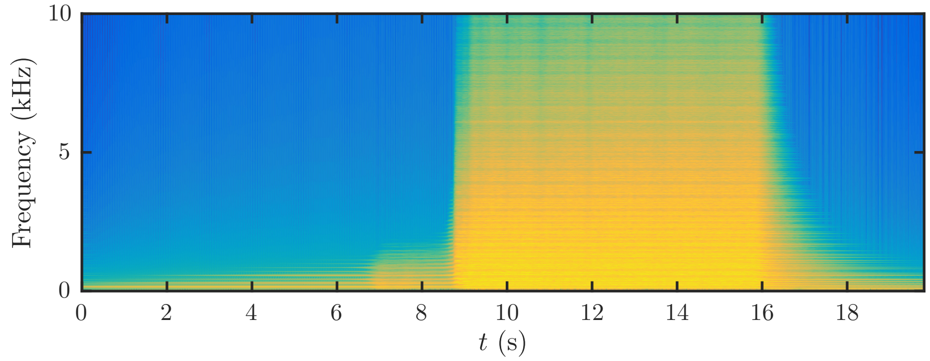
Figure 3.9 shows the behaviour of a simply supported metal square plate with $L = 0.5$ m, $\kappa = 1.5$ and thickness $H = 1$ mm excited with a sinusoidal force with $f_{exc} = 69$ Hz and variable amplitude A_{exc} from 0 to 50 N. This particular frequency is not very close to any eigenfrequency of the system, the (2,1) and (2,2) modes being at 49 Hz and 78 Hz, respectively. Losses are included in the model, with $\sigma_0 = 0.75$ and $\sigma_1 = 0.001$. A linearly increasing force is applied from $t = 0$ to $t = 16$ s, when it suddenly stops. At that point, the plate vibrates freely for 4 s, for a total duration of the simulation of 20 s.

The displacement of the plate can be seen in Figure 3.9a. The sudden jump in signal is evident at about 9 s, corresponding to an excitation amplitude of about 28 N. This represents the onset of turbulent behaviour. The spectrum of the velocity signal up to 10 kHz is plotted in Figure 3.9b. The activation of the entire frequency range here is apparent. A slow build up of high frequency energy is also noticeable, which is typical of the turbulent behaviour.

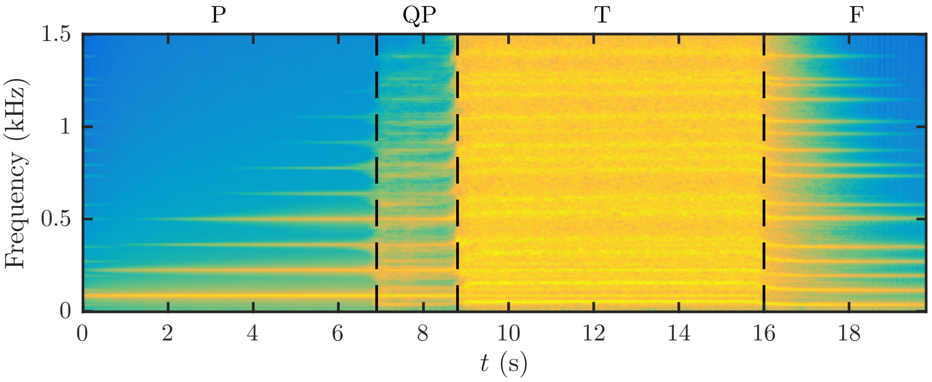
Additional information can be gathered by looking at a zoomed portion of the spectrum up to about 1500 Hz. Here the four regimes described above can be clearly distinguished. The initial portion represents the periodic regime, where lines associated with odd multiples of the excitation frequency (f_{exc}, $3f_{exc}$, $5f_{exc}$, $\dots$) are visible. At about 7 s, subharmonic frequencies appear, which signal the onset of the quasi-periodic regime. A second bifurcation takes place at $t = 8.8$ s, when a turbulent regime sets in. At $t = 16$ s, the external forcing is switched off, and the plate vibrates freely. The energy of the system is dissipated by the loss terms in the equations, giving rise to a downward pitch glide that can be clearly noticed.



(a) Displacement of the central point of the plate



(b) Spectrum of the velocity signal up to 10 kHz.



(c) Spectrum of the velocity signal up to 1.5 kHz.

Figure 3.9: Numerical simulation of the nonlinear behaviour of a square plate under an external sinusoidal force with increasing amplitude. The sampling rate is 44.1 kHz. In the last figure, the various regimes are indicated (Periodic, Quasi-Periodic, Turbulent and Free).

3.3 Extension of the nonlinear models to membranes

In this section, the nonlinear models for plate vibration are extended to membranes. Although short, this section is fundamental for the modelling of drums described in the later chapters.

3.3.1 Berger nonlinearity

The linear behaviour of membranes has been briefly presented in Section 2.2.3. In order to account for nonlinear phenomena in drum membranes, some authors [7, 29] have included in the stiff membrane equation (2.62) a tension modulation term similar to the Berger nonlinearity discussed in Section 3.1. In this case, the equation of motion becomes

$$\rho H \partial_t^2 w = \mathcal{O}_m w + \mathfrak{T}\Delta w, \tag{3.67}$$

where $\mathcal{O}_m$ is the linear operator defined in (2.64) and $\mathfrak{T}$ has been defined in (3.2). It is clear that, in this case, the additional Berger term is responsible for changing the tension of the membrane, such that

$$T \longrightarrow T + \mathfrak{T} \tag{3.68}$$

thus increasing the equilibrium value T by a quantity $\mathfrak{T}$ that depends on the displacement of the membrane.

One possible finite difference scheme for (3.67) can be written in terms of the schemes presented earlier, and is simply

$$\rho H \delta_{tt} w^n = \mathfrak{O}_m w^n + \mathfrak{t}^n \delta_\Delta w^n, \tag{3.69}$$

where $\mathfrak{O}_m$ is the discrete linear operator defined in (2.165) and $\mathfrak{t}^n$ is the tension modulation term defined in (3.9). The discrete energy for the system in this case can be written as

$$\mathfrak{h}_{\text{tot}} = \mathfrak{h}_{\text{sm}} + \mathfrak{u}_{\text{Berger}}, \tag{3.70}$$

where the energy $\mathfrak{h}_{\text{sm}}$ for a stiff membrane under linear conditions has been defined in (2.164) and the expression for $\mathfrak{u}_{\text{Berger}}$ can be found in (3.12). This energy is conserved under lossless conditions.

3.3.2 von Kármán nonlinearity

A different approach, which is often used in engineering applications [109, 108], but has never been adopted so far in acoustical applications, is the inclusion in the membrane equation of a nonlinear term of von Kármán type. This leads to a system similar to (3.28), but with the additional membrane term. The new equations read

$$\rho H \partial_t^2 w = \mathcal{O}_m w + \mathcal{L}(w, \Phi) \tag{3.71a}$$

$$\Delta^2 \Phi = -\frac{EH}{2}\mathcal{L}(w, w). \tag{3.71b}$$

A similar system applied to membranes has been proposed by Yasuda and Asano [207] and by Nayfeh [136], although in both cases the fourth order linear term was discarded. Given the fixed conditions used for the membranes, no additional boundary terms are needed besides those presented in this chapter.

A finite difference scheme for the above system can be easily written by combining the schemes presented earlier, and reads

$$\rho H \delta_{tt} w^n = \mathfrak{D}_m w^n + \mathfrak{l}(\mu_{t\cdot}\Phi^n, w^n) \tag{3.72a}$$

$$\mu_{t+}\delta_{\Delta,\Delta}\Phi^n = -\frac{EH}{2}\mathfrak{l}(w^n, \mathrm{s}_{t+}w^n), \tag{3.72b}$$

following the energy conserving scheme used in (3.49).

The total energy in this case can be written as

$$\mathfrak{h}_{\mathrm{tot}} = \mathfrak{h}_{\mathrm{sm}} + \frac{1}{2EH}\mu_{t+}\left\|\delta_{\Delta}\Phi\right\|^2_{\mathfrak{D}}, \tag{3.73}$$

where the norm in the last term is calculated over the discrete domain $\mathfrak{D}$.

3.3.3 Further remarks

One interesting question that arises is if the two models lead to comparable results when applied to membrane vibration. In the case of plates, it has been seen that although both models are able to replicate pitch glide effects, the von Kármán model produces more complex and richer phenomena, which ultimately lead to a more realistic synthetic sound. It is therefore important to evaluate up to which point the simple Berger model can be used in the case of nonlinear membrane vibration, and when the von Kármán model is needed, instead.

A preliminary study [186] regarding this particular issue has been performed by the Author in collaboration with M. Newton. Some of the results of this paper are presented in Appendix A.

Chapter 4

Collisions

Many musical instruments rely on collisions for sound generation. Typical examples are reeds in wind instruments, the hammer-string interaction in pianos or the slapping gesture in a bass guitar. Percussion instruments, however, are probably the most salient example of this type of phenomena. In drums and gongs, for example, the sound is generally produced by striking these instruments with a mallet. A physical model must then take into account the subtle phenomena that occur during the short period (generally a few milliseconds) of contact. Experimental investigations of collisions in tom tom drums have been performed by Dahl *et al.* [66, 200], and they confirm the complex shape of the collision force during the impact.

When it comes to numerical simulation, collisions play an important role in many branches of science and engineering. The main challenge is to detect an interaction between two objects given some physical and geometrical constraints, and to correctly model the ensuing reaction. To this end, two different approaches are generally adopted. The first method consists in including the constraint in the total energy of the system through Lagrange multipliers. A second approach, generally referred to as penalty or regularisation method, allows some interpenetration between the objects, and includes in the equations of motion a repelling force term proportional to it. In this work, this latter approach will be adopted, and its description in the particularly simple case of a mass colliding against a rigid barrier is discussed in Section 1. More information on both methods and on collision simulation can be found in [206].

In acoustical applications, penalty based methods have been used extensively to describe collisions in musical instruments. These generally depend on a collision force f_c that can be defined as

$$f_c = K\,[\zeta]_+^{\alpha},$$

where ζ is a measure of the interpenetration between two objects (e.g., a hammer and a string, or a mallet and a membrane), and K and $\alpha > 1$ are two parameters, usually determined experimentally. The force is active only when a positive interpenetration is detected (hence the symbol $[\,\cdot\,]_+$, see Section 4.1). This penalty method can be viewed within the mathematical framework originated by Hertz and his contact law [120]. The original formulation set down for a sphere can be extended to more complex systems, and the validity of this approach can be confirmed experimentally *a posteriori*. At first, many works used this approach with piano hammers [171, 36, 95, 49, 50], combining both experimental analysis and numerical simulation. This has soon been extended to collision modelling in many different cases, such as xylophones

[51], plates [53, 119] and drums [146]. Since then, this approach has been adopted extensively for a wide variety of systems (see [27] for a recent review).

The interpretation of the collision force f_c and its physical model can be twofold. On the one hand, the interpenetration ζ can be thought of as an elastic deformation appearing on the material as a consequence of the collision, which disappears as soon as the two objects separate. This corresponds to the original Hertzian formulation. Hysteresis effects can also be taken into account [36, 170, 8]. On the other hand, f_c can be interpreted as a mathematical device to obtain rigid body collisions. In a strict physical sense, allowing interpenetration to occur in rigid bodies clashes with their very definition as non-deformable objects. It can be shown, however, that the value of ζ can be made arbitrarily small with a sensible choice of K and α, thus leading to a rigid collision in the limit. This particular framework is useful in the case of percussion instruments, where the mallets and sticks used are generally made of solid wood. It is also relevant in the distributed collisions between the strings and the membrane that can be found in snare drums [153]. In this case, in fact, it would be hard to justify deformation or interpenetration of the materials. Therefore, this mathematical interpretation is adopted in this work.

One drawback in the use of a power law collision force in a numerical scheme is stability. Explicit and semi-implicit schemes, which are particularly simple to implement [19], suffer however from the lack of an associated energy. As such, the stability of the numerical scheme cannot be guaranteed. A possible alternative is the use of a collision potential Π which leads directly to a conserved energy for the system (see [110] for an analysis of this method in the continuous case). This, however, comes at the price of complicating the numerical implementation, by requiring the solution of a nonlinear equation in order to update the scheme.

This approach has first been proposed by Chaigne and collaborators in [52], in the case of a lumped mallet colliding against a timpani membrane. The important issue of existence and uniqueness of a solution is discussed and solved therein in the discrete case. A similar approach has been adopted by Chabassier for a piano hammer striking a string [46], and by Chatziioannou and van Walstijn to model lumped collisions [56] and string/bridge interactions [58]. This method has been applied to the triple interaction between string, fretboard and player's finger by Bilbao *et al.* [26] and to bowed string instruments by Desvages *et al.* [69]. Extensions of these ideas to the case of distributed collisions in 2D, as in the case of a string striking against a membrane, have been presented by the Author in [185, 184]. See also [27] for a recent review analysing all these cases.

4.1 Collision of a mass against a rigid barrier

Consider a lumped object of mass M and vertical position $z(t)$ at time $t \geq 0$, colliding from below against a rigid barrier positioned at $z = 0$. A penalty approach prescribes the introduction of a force $f(t)$ that acts whenever some penetration is detected, and that can be defined as

$$f = K[z]_+^\alpha, \qquad [z]_+ = \frac{1}{2}\left(z + |z|\right). \tag{4.1}$$

This expression for the collision force is particularly useful, and has been adopted in many previous works (see [27] for a recent review). Here, the parameters K and α are obtained experimentally, and depend on the particular model at hand. The symbol $[\,\cdot\,]_+$ represents the positive part of the argument. The action of f can be understood as follows. Whenever the position of the mass lies above the barrier level, the positive part of z is different from zero, hence the collision potential is activated. A repelling force is then generated, which pushes the mass downwards, away from the barrier. At this point, $[z]_+ = 0$ and the collision force is no longer active.

For a free mass, Newton's law can be written as

$$M \frac{\mathrm{d}^2 z}{\mathrm{d}t^2} = -f, \tag{4.2}$$

and this is the equation that will be simulated numerically. In implementation, it is convenient to introduce the non-negative potential function $\Pi(z) \geq 0$ from which f is generated by differentiation, such that

$$f = \frac{\mathrm{d}\Pi}{\mathrm{d}z} = \frac{\mathrm{d}\Pi/\,\mathrm{d}t}{\mathrm{d}z/\,\mathrm{d}t}, \qquad \Pi = \frac{K}{\alpha+1}[z]_+^{\alpha+1}. \tag{4.3}$$

When the object is deformable, the penetration can be interpreted as a compression without loss of energy. When the collision is meant to be perfectly rigid, however, this model is physically inconsistent, at least at the philosophical level, as it contradicts the definition of rigid bodies as non-deformable objects. Nonetheless, in practical applications the parameters K and α can be chosen such that the amount of spurious penetration be controlled and be made arbitrarily small (at least within the limits of numerical computation). More on this will be said later.

4.1.1 Energy balance

In order to find an energy balance for the system, one might simply multiply (4.2) by $\mathrm{d}z/\,\mathrm{d}t$,

$$M \frac{\mathrm{d}z}{\mathrm{d}t} \frac{\mathrm{d}^2 z}{\mathrm{d}t^2} = -\frac{\mathrm{d}z}{\mathrm{d}t} f = -\frac{\mathrm{d}z}{\mathrm{d}t} \frac{\mathrm{d}\Pi}{\mathrm{d}z} = -\frac{\mathrm{d}\Pi}{\mathrm{d}t}\,. \tag{4.4}$$

This directly leads to an energy balance of the form

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{tot}}}{\mathrm{d}t} = 0, \qquad \mathfrak{H}_{\mathrm{tot}} = \mathfrak{H}_{\mathrm{M}} + \Pi, \qquad \mathfrak{H}_{\mathrm{M}} = \frac{M}{2}\left(\frac{\mathrm{d}z}{\mathrm{d}t}\right)^2, \tag{4.5}$$

where $\mathfrak{H}_{\mathrm{tot}}$ is the total energy of the system and $\mathfrak{H}_{\mathrm{M}}$ is the (kinetic) energy of the mallet. It is now possible to derive some bounds on the velocity and position of the mass, by exploiting the fact that the energy is non-negative:

$$\mathfrak{H}_{\mathrm{M}} \leq \mathfrak{H}_{\mathrm{tot}}(0) \quad \Rightarrow \quad \left|\frac{\mathrm{d}z}{\mathrm{d}t}\right| \leq \sqrt{\frac{2\mathfrak{H}_{\mathrm{tot}}(0)}{M}} \tag{4.6a}$$

$$\Pi \leq \mathfrak{H}_{\mathrm{tot}}(0) \quad \Rightarrow \quad z(t) \leq \sqrt[\alpha+1]{\frac{(\alpha+1)\mathfrak{H}_{\mathrm{tot}}(0)}{K}} \tag{4.6b}$$

Thus, when a rigid collision is to be simulated, the amount of penetration allowed can be controlled by a sensible choice of α and K.

4.1.2 Finite difference scheme

A finite difference scheme for (4.2) can be written as follows. Let k be the discrete time step, and z^n an approximation for $z(t)$ at time step nk. The discrete equation for the colliding mass is

$$M\delta_{tt}z^n = -f^n \tag{4.7}$$

where f^n represents the finite difference approximation of the collision force $f(t)$ defined as

$$f^n = \frac{\delta_{t-}\Pi^{n+1/2}}{\delta_{t\cdot}z^n}, \qquad \Pi^{n+1/2} = \mu_{t+}\Pi(z^n). \tag{4.8}$$

As before, the discrete collision force depends on the potential Π defined in (4.3) evaluated for the discrete variable z^n. This scheme is similar to that used by Chatziioannou and van Walstijn in [56].

4.1.3 Discrete energy balance and bounds

An energy balance for the system can be obtained by multiplying (4.7) by $\delta_{t\cdot}z$:

$$\frac{M}{2}\delta_{t\cdot}z^n\,\delta_{tt}z^n = -\delta_{t\cdot}z^n\,f^n = -\delta_{t\cdot}z^n\frac{\delta_{t-}\Pi^{n+1/2}}{\delta_{t\cdot}z^n} = -\delta_{t-}\Pi^{n+1/2}, \tag{4.9}$$

which leads directly to

$$\delta_{t-}\mathfrak{h}_{\text{tot}}^{n+1/2} = 0, \qquad \mathfrak{h}_{\text{tot}}^{n+1/2} = \mathfrak{h}_{\text{M}}^{n+1/2} + \mu_{t+}\Pi^n \geq 0, \qquad \mathfrak{h}_{\text{M}}^{n+1/2} = \frac{M}{2}\left(\delta_{t+}z^n\right)^2, \tag{4.10}$$

where $\mathfrak{h}_{\text{tot}}$ is the discrete total energy of the system and $\mathfrak{h}_{\text{M}}$ is the discrete energy of the mallet. Thus, $\mathfrak{h}_{\text{tot}}$ is conserved and is non-negative.

As in the continuous case, a bound can be found on the velocity and position of the mass

$$\mathfrak{h}_{\text{M}}^{n+1/2} \leq \mathfrak{h}_{\text{tot}}^{0} \quad \Rightarrow \quad |\delta_{t+}z^n| \leq \sqrt{\frac{2\mathfrak{h}_{\text{tot}}^{0}}{M}} \tag{4.11a}$$

$$\mu_{t+}\Pi^n \leq \mathfrak{h}_{\text{tot}}^{0} \quad \Rightarrow \quad z^n \leq \sqrt[\alpha+1]{\frac{2(\alpha+1)\mathfrak{h}_{\text{tot}}^{0}}{K}} \tag{4.11b}$$

Notice that the factor $2^{1/\alpha+1}$ makes the second bound slightly wider than that of the corresponding continuous case.

The particular choice for the discretisation of $\Pi^{n+1/2}$ given in (4.8) is not unique. One could use, in fact

$$\Pi^{n+1/2} = \Pi(\mu_{t+}z^n), \tag{4.12}$$

where the time averaging operator now acts directly on z^n, or even a parameterised combination of the two expressions. The energy balance for the scheme would not be modified by this, but different and less stringent bounds on the solution would derive from the new expression[1].

---

[1] S. Bilbao, private communication.

4.1.4 Implementation

The numerical implementation of (4.7) can be obtained by expanding it as

$$\begin{aligned} z^{n+1} &= -2z^n + z^{n-1} - \frac{k^2}{M} f^n & (4.13a)\\ &\overset{(4.8)}{=} -2z^n + z^{n-1} - \frac{\delta_{t-}\Pi^{n+1/2}}{\delta_{t\cdot} z} & (4.13b)\\ &\overset{(4.8)}{=} -2z^n + z^{n-1} - \frac{\Pi(z^{n+1}) - \Pi(z^{n-1})}{z^{n+1} - z^{n-1}}. & (4.13c) \end{aligned}$$

Thus, the update of the scheme requires the solution of a nonlinear equation of the form

$$\Gamma(r) = r + m\frac{\Pi(r+a) - \Pi(a)}{r} + b = 0, \tag{4.14}$$

where the new variables

$$r = z^{n+1} - z^{n-1}, \qquad a = z^{n-1}, \qquad b = -2z^n + 2z^{n-1}, \qquad m = \frac{k^2}{M} \tag{4.15}$$

have been introduced to simplify the notation. Once r is determined, the scheme can be updated immediately. As is always the case with nonlinear equations, the existence of a solution is not generally guaranteed. In this particular case, however, a proof of existence and uniqueness can be derived [52]. First of all, it can be remarked that $\Gamma(r)$ is well behaved at $r = 0$. One has, in fact

$$\lim_{r\to 0} \Gamma(r) = m\Pi'(a) + b \tag{4.16}$$

where

$$\Pi'(x) = \begin{cases} K\,x^{\alpha}, & \text{for } x \geq 0\\ 0, & \text{for } x < 0 \end{cases} \tag{4.17}$$

Hence, $\Gamma(0)$ is continuous and well defined at $r = 0$, given the continuity of the function Π'. Furthermore, the derivative Γ' of (4.14) can be calculated as

$$\Gamma'(r) = 1 + \frac{m}{r^2}\left[-\Pi(r+a) + \Pi(a) + r\Pi'(r+a)\right]. \tag{4.18}$$

This function is also well behaved at $r = 0$, as long as $\alpha \geq 1$, with

$$\lim_{r\to 0} \Gamma'(r) = 1 + m\Pi''(a). \tag{4.19}$$

where

$$\Pi''(x) = \begin{cases} \alpha K\,x^{\alpha-1}, & \text{for } x \geq 0\\ 0, & \text{for } x < 0 \end{cases} \tag{4.20}$$

The heart of the uniqueness proof lies in the fact that $\Gamma'(r) \geq 1$, meaning that Γ is bounded away from zero. As such, the solution to (4.14), if exists, must be unique. This is equivalent to showing that

$$r\Pi'(r+a) \geq \Pi(r+a) - \Pi(a), \tag{4.21}$$

which holds given the particular choice of Π.

4.1.5 Newton-Raphson method

As discussed in the previous section, the implementation of the collision model just presented relies on the solution of a nonlinear equation. This is a problem frequently encountered in every branch of science: given a function $f(x)$, one wishes to compute its roots, that is the solutions $\bar{x}$ to the equation $f(x) = 0$.

One possible way to tackle this problem is the use of an iterative method such as Newton-Raphson [100]. Starting from a guess solution x_0, one can compute successive approximations to $\bar{x}$ as

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}, \tag{4.22}$$

where f' is the derivative of f. In order to find the solution of $f(x) = 0$, then, the derivative of f must be known beforehand.

For the particular choice of Π (4.3), the derivative for $r \neq 0$ is available, as discussed in the previous section. Starting from an initial value r_0, Newton-Raphson method applied to (4.14) gives successive approximations to the solution as

$$r_{j+1} = r_j - \frac{\Gamma(r_j)}{\Gamma'(r_j)}. \tag{4.23}$$

In implementation, particular care must be taken when evaluating Γ and its derivative at $r = 0$. In this case, the alternative expressions (4.16) and (4.19) must be used, instead.

4.1.6 Simulation results

Consider a mass $M = 10$ g, with initial position $z_0 = -8$ mm and velocity $v_0 = \sqrt{3}$ m/s, colliding against a barrier positioned at $z = 0$. The trajectory followed by the mass is plotted in Figure 4.1, together with the normalised energy variations of the scheme for $K = 10^8$ and $\alpha = 2.5$ at sampling rate of 44.1 kHz. The activation of the collision force is marked in red. As can be noticed from the trajectory, the position of the mass lies above the barrier for a period of about 3.5 ms, meaning that penetration takes place, as expected. Energy variations are exactly zero before the contact. Then, small quantised jumps in the discrete energy can be seen when the collision is taking place. Other types of progressively bigger energy jumps can be noticed when the mass moves away from the barrier and is not in contact with it. These can be interpreted as a loss of accuracy in the calculation of the kinetic energy due to the increasingly larger value of z, as the mass moves away from the barrier. Notice, also, that such a phenomenon does not occur when the mass approaches the barrier. More details can be found in Appendix B.4.

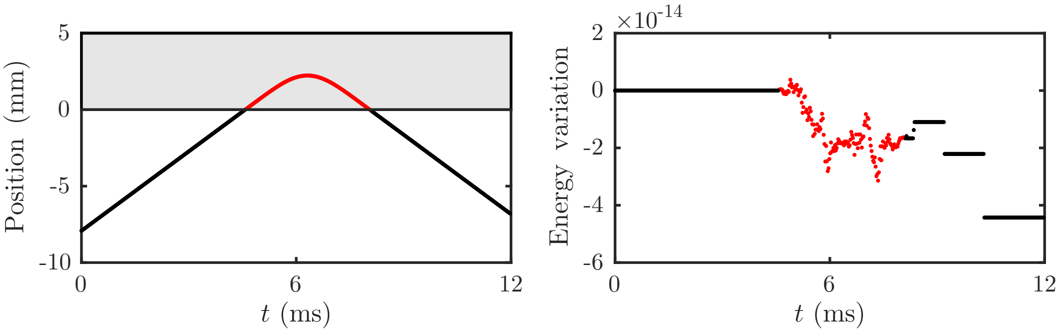


Figure 4.1: Left: Trajectory of a mass $M = 10$ g colliding against a barrier positioned at $z = 0$. Right: Normalised energy variations for the numerical scheme. The color red in both plots signals that collision is taking place. The sampling rate is 44.1kHz.

4.2 Lumped 2D collision

The case of a mallet colliding against a membrane or a plate is particularly useful in the context of percussion instruments. Consider, as a more general situation, a stiff linear membrane as described in Section 2.2.3, defined over a region $\mathcal{D}$ endowed with Cartesian coordinates, together with a lumped object of mass M acting as a mallet from above. This system can be described by the following set of equations:

$$\rho H \partial_t^2 w = \mathcal{O}_m w - \mathcal{F}_c \tag{4.24a}$$

$$M \frac{\mathrm{d}^2 z}{\mathrm{d}t^2} = f_c, \tag{4.24b}$$

where as usual, w denotes the transverse vibration of the membrane, ρ and H are the density and thickness, $\mathcal{O}_m$ represents some combination of 2D linear operators (see (2.64)), and z represents the position of the mallet. $\mathcal{F}_c$ and f_c represent the collision force acting on the membrane and on the mallet, respectively. The term "force" for $\mathcal{F}_c$ is a slight abuse of language, since it is really a force density. These two quantities are related by the condition

$$\mathcal{F}_c = g\, f_c, \tag{4.25}$$

where $g(x, y)$ represents a spatial distribution centred around the impact point on the 2D surface. It is useful to normalise g to unity, such that

$$\int_{\mathcal{D}} g \,\mathrm{d}\sigma = 1. \tag{4.26}$$

Notice the different sign of the collision forces, given the fact that in one case the force points upwards, and in the other downwards.

The definition of the collision force is in terms of a potential Π

$$f_c = \frac{\mathrm{d}\Pi}{\mathrm{d}\zeta}, \qquad \zeta = \int_{\mathcal{D}} g\, w \,\mathrm{d}\sigma - z \tag{4.27}$$

where ζ is now a measure of the penetration of the mallet—notice that ζ is positive when the mallet lies below the portion of the 2D surface averaged over the distribution function g. The potential Π can be chosen as a nonlinear power law

$$\Pi(\zeta) = \frac{K}{\alpha+1}\,[\zeta]_+^{\alpha+1}\,, \tag{4.28}$$

where K and α are the collision coefficient and exponent, respectively. A similar approach has been used by Rhaouti *et al.* in [146].

As in case of the mass colliding against a rigid barrier, this system possesses an energy which is conserved in the lossless case. The usual multiplication by the velocity and integration over the domain leads to

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{sm}}}{\mathrm{d}t} = -\int_{\mathcal{D}} \partial_t w\,\mathcal{F}_c\,\mathrm{d}\sigma \tag{4.29a}$$

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{M}}}{\mathrm{d}t} = \frac{\mathrm{d}z}{\mathrm{d}t}\,f_c, \tag{4.29b}$$

where $\mathfrak{H}_{\mathrm{sm}}$ is the energy for a stiff membrane defined in (2.63) and $\mathfrak{H}_{\mathrm{M}}$ is the kinetic energy of the mallet. Summing the two in order to obtain the energy for the entire system gives

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{sm}}}{\mathrm{d}t} + \frac{\mathrm{d}\mathfrak{H}_{\mathrm{M}}}{\mathrm{d}t} \overset{(4.25)}{=} -\int_{\mathcal{D}} \partial_t w\, g\, f_c\,\mathrm{d}\sigma + \frac{\mathrm{d}z}{\mathrm{d}t}\,f_c \tag{4.30a}$$

$$\overset{(4.27)}{=} -\frac{\mathrm{d}\zeta}{\mathrm{d}t}\,\frac{\mathrm{d}\Pi}{\mathrm{d}\zeta} \tag{4.30b}$$

$$= -\frac{\mathrm{d}\Pi}{\mathrm{d}t} \tag{4.30c}$$

Thus, an energy balance is obtained with

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{tot}}}{\mathrm{d}t} = 0, \qquad \mathfrak{H}_{\mathrm{tot}} = \mathfrak{H}_{\mathrm{sm}} + \mathfrak{H}_{\mathrm{M}} + \Pi. \tag{4.31}$$

4.2.1 Finite difference scheme

The system (4.24) can be discretised as follows

$$\rho H \delta_{tt} w = \mathfrak{O}_m w - g f_c \tag{4.32a}$$

$$M \delta_{tt} z = f_c, \tag{4.32b}$$

where g is a discrete distribution normalised such that $\langle g, 1\rangle_{\mathfrak{D}} = 1$, and $\mathfrak{O}_m$ is the discrete counterpart of the operator $\mathcal{O}_m$ given in (2.165). The discrete collision force f_c can be written as

$$f_c^n = \frac{\delta_{t-}\Pi^{n+1/2}}{\delta_{t\cdot}\zeta^n}, \qquad \zeta^n = \langle g, w^n\rangle_{\mathfrak{D}} - z^n, \qquad \Pi^{n+1/2} = \mu_{t+}\Pi^n. \tag{4.33}$$

A discrete energy balance is easily arrived at through the usual techniques. Summing together the energy for the mallet and for the membrane, one obtains

$$\delta_{t-}\mathfrak{h}_{\mathrm{sm}} + \delta_{t-}\mathfrak{h}_{\mathrm{M}} = -\langle \delta_{t\cdot} w, g f_c\rangle_{\mathfrak{D}} + \delta_{t\cdot} z f_c \tag{4.34a}$$

$$\overset{(4.33)}{=} -\delta_{t\cdot}\zeta\,\frac{\delta_{t-}\mu_{t+}\Pi}{\delta_{t\cdot}\zeta} \tag{4.34b}$$

$$= -\delta_{t-}\mu_{t+}\Pi, \tag{4.34c}$$

leading to an energy balance analogous to the continuous case of the form

$$\delta_{t-}\mathfrak{h}_{\rm tot} = 0, \qquad \mathfrak{h}_{\rm tot} = \mathfrak{h}_{\rm sm} + \mathfrak{h}_{\rm M} + \Pi. \tag{4.35}$$

The implementation of this scheme requires the solution of a nonlinear equation. First, write the update for the system (4.32):

$$w^{n+1} = \widetilde{w} + \frac{k^2}{\rho H(1+\sigma_0 k)} g f_c^n \tag{4.36a}$$

$$z^{n+1} = \widetilde{z} + \frac{k^2}{M} f_c^n, \tag{4.36b}$$

where $\widetilde{w}$ and $\widetilde{z}$ represent the quantities known from the previous time steps, see (2.167), and σ_0 is the loss parameter defined in Section 2.2.3. Notice now that f_c^n depends on ζ^{n+1}, which in turn depends on both w^{n+1} and z^{n+1}. Taking an inner product of the first equation with g and subtracting the second equation, leads to a nonlinear equation in the unknown $r = \zeta^{n+1} - \zeta^{n-1}$ of the form

$$\Gamma(r) = r + m\frac{\Pi(r+a) - \Pi(a)}{r} + b = 0, \tag{4.37}$$

which is formally identical to (4.14). In this case, however, the various quantities are defined as

$$a = \zeta^{n-1}, \quad b = \zeta^n + \widetilde{z} - \langle g, \widetilde{w}\rangle_{\mathfrak{D}}, \quad m = \frac{k^2}{\rho H(1+\sigma_0 k)}\, \|g\|_{\mathfrak{D}}^2 + \frac{k^2}{M}. \tag{4.38}$$

4.2.2 Simulation results

Consider a linear steel plate with thickness 1 mm and $\kappa = 1.5$, excited with a lumped mallet of mass 0.1 kg. Figure 4.2 shows the results of a simulation with $K = 10^{10}$ and $\alpha = 2.5$. The position of the mallet and the averaged striking location on the plate is presented, together with the profile of the collision force generated by the impact, the energy curves for the various components and the normalised variations of the total energy. For illustration purposes, a particularly large value for α has been chosen, which is far from a perfectly elastic collision—the maximum interpenetration here (the difference between the mallet and the membrane position) is about 0.8 mm. A smaller value can be achieved with smaller values for α. It is interesting to analyse the force profile obtained. The initial portion of the signal resembles that of a raised cosine strike that has been used before. However, rather than reaching zero after the maximum, the profile shows two additional bumps, that considerably lengthen the duration of the impact.

4.3 Distributed 2D collisions

In this section, the interaction between a distributed 1D object, like a string, and a 2D surface is studied. First, however, it is necessary to make a short detour in order to introduce the equation for a stiff string. The energy techniques adopted for the plates and membranes can be successfully applied in this case, leading to an energy balance and to the determination of suitable boundary conditions.

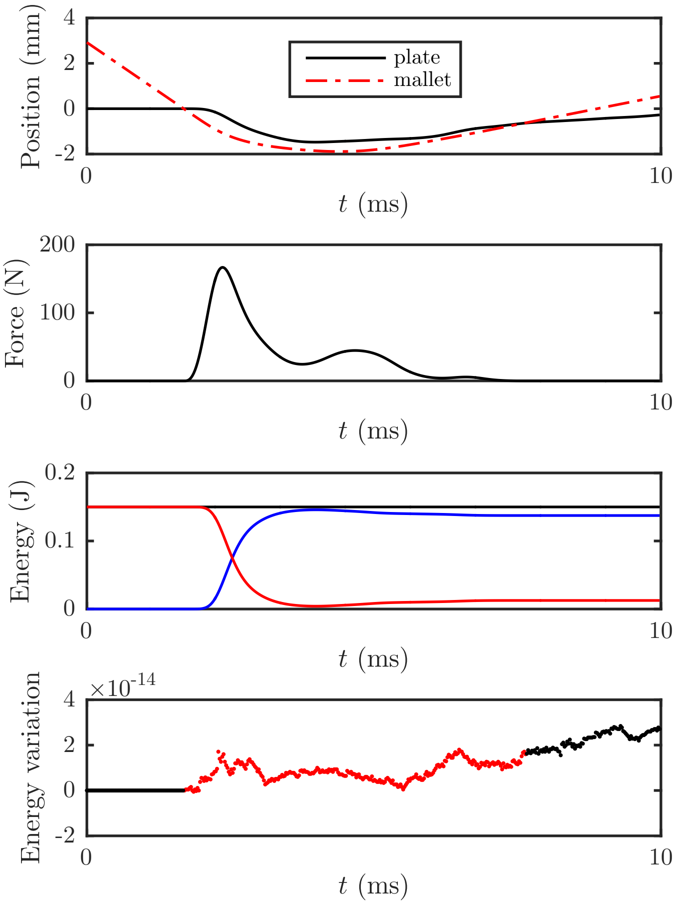


Figure 4.2: Simulation results for an interacting system of a mallet ($M = 0.1$ kg), and a linear rectangular steel plate ($L_x = 0.4$ m, $L_y = 0.7$ m, thickness $H = 1$ mm, $\kappa = 1.5$) under lossless conditions. From top to bottom: Displacement of the mallet and of the corresponding striking location on the plate; profile of the generated force; contributions to the total discrete energy (black) from the mallet (red) and plate (blue) energy; normalised energy variations for the numerical scheme. The sampling rate is 44.1 kHz.

4.3.1 Stiff string equation

Consider the transverse displacement $u(\chi, t)$ of a stiff lossy string defined over a 1D region $\mathcal{D}_s = \{\chi \in [0, L]\}$ and with $t \in \mathbb{R}^+$. The equation of motion can be written as

$$\rho_s A \partial_t^2 u = \mathcal{O}_s u, \qquad \mathcal{O}_s = T_s \partial_\chi^2 - E_s I_s \partial_\chi^4 - 2\sigma_{0,s}\rho_s A \partial_t + 2\sigma_{1,s}\rho_s A \partial_\chi^2 \partial_t \tag{4.39}$$

with ρ_s the density, A the cross sectional area, T_s the tension, E_s Young's modulus, I_s the moment of inertia, and $\sigma_{0,s}$ and $\sigma_{1,s}$ the loss coefficients for the string. The linear operator $\mathcal{O}_s$ will considerably simplify the notation in the next sections. Notice that this equation is formally very similar to that of a stiff membrane. Similar models for the string have been used before, particularly by Hall [95] and Chaigne and Askenfelt [49, 50]. With respect to the latter case, a different choice has been made here for the frequency dependent loss term. This is to avoid the use of third order derivatives, which often lead to instabilities in the numerical scheme.

An energy balance can be shown to exist for this particular system as well. Multiplying by the velocity $\partial_t u$ and integrating over the domain $\mathcal{D}_s$ gives [19]

$$\frac{\mathrm{d}\mathfrak{H}_\mathrm{s}}{\mathrm{d}t} = -\mathfrak{Q}_\mathrm{s} + \mathfrak{B}_\mathrm{s}\Big|_0^L, \tag{4.40}$$

with

$$\mathfrak{H}_\mathrm{s} = \frac{\rho_s A}{2}\left\|\partial_t u\right\|^2_{\mathcal{D}_s} + \frac{T_s}{2}\left\|\partial_\chi u\right\|^2_{\mathcal{D}_s} + \frac{E_s I_s}{2}\left\|\partial^2_\chi u\right\|^2_{\mathcal{D}_s} \tag{4.41a}$$

$$\mathfrak{Q}_\mathrm{s} = 2\sigma_{0,s}\rho_s A\left\|\partial_t u\right\|^2_{\mathcal{D}_s} + 2\sigma_{1,s}\rho_s A\left\|\partial_{t\chi} u\right\|^2_{\mathcal{D}_s} \tag{4.41b}$$

$$\mathfrak{B}_\mathrm{s} = T_s\partial_t u\,\partial_\chi u - E_s I_s\partial_t u\,\partial^3_\chi u + E_s I_s\partial_{t\chi} u\,\partial^2_\chi u + 2\sigma_{1,s}\rho_s A\partial_t u\,\partial_\chi u. \tag{4.41c}$$

Here, $\mathfrak{H}_\mathrm{s}$ represents the total energy of the system and $\mathfrak{Q}_\mathrm{s}$ the dissipated power. $\mathfrak{B}_\mathrm{s}$ groups together the boundary terms, which must be evaluated at the end points. As in the case of the plate, this fourth order equation requires two conditions at each end, and several possibilities are available. For the purpose of this work, only the case of simply supported conditions at both ends is considered, which can be written as

$$u\big|_0 = 0, \quad \partial^2_\chi u\big|_0 = 0, \qquad u\big|_L = 0, \quad \partial^2_\chi u\big|_L = 0. \tag{4.42}$$

The interested reader is referred to [19] for other types of conditions.

Consider now a finite difference implementation of this equation. Let u_l^n be a discrete approximation to $u(x,t)$, with $n \in \mathbb{Z}^+$ and $0 \leq l \leq N$ positive integers, over a 1D domain $\mathfrak{D}_s$ with spacing h. The time step is indicated with k, as usual. A finite difference scheme simulating (4.39) can be written as follows

$$\rho_s A\delta_{tt} u = \mathfrak{O}_s u, \qquad \mathfrak{O}_s = T_s\delta_{\chi\chi} - E_s I_s\delta_{\chi\chi\chi\chi} - 2\sigma_{0,s}\rho_s A\delta_{t\cdot} + 2\sigma_{1,s}\rho_s A\delta_{t-}\delta_{\chi\chi} \tag{4.43}$$

where $\delta_{\chi\chi}$ and $\delta_{\chi\chi\chi\chi}$ are the second order and fourth order difference operators, see (2.102) and (2.104). As in the continuous case, the introduction of the operator $\mathfrak{O}_s$ will considerably simplify the notation in the next sections.

The energy balance for this scheme is

$$\delta_{t-}\mathfrak{h}_\mathrm{s} = -\mathfrak{q}_s - \mathfrak{b}_0 + \mathfrak{b}_N \tag{4.44}$$

where the various quantities are

$$\mathfrak{h}_\mathrm{s} = \frac{\rho_s A}{2}\left\|\delta_{t+} u\right\|^2_{\mathfrak{D}_s} + \frac{T_s}{2}\left\|\delta_{\chi+} u\right\|^2_{\mathfrak{D}_s} + \frac{E_s I_s}{2}\left\|\delta_{\chi\chi} u\right\|^2_{\mathfrak{D}_s} - \frac{\sigma_{1,s}\rho_s A k}{2}\left\|\delta_{t+}\delta_{\chi+} u\right\|^2_{\mathfrak{D}_s} \tag{4.45a}$$

$$\mathfrak{q}_s = 2\rho_s A\left(\sigma_{0,s}\left\|\delta_{t\cdot} u\right\|^2_{\mathfrak{D}_s} + \sigma_{1,s}\left\|\delta_{t\cdot}\delta_{\chi+} u\right\|^2_{\mathfrak{D}_s}\right) \tag{4.45b}$$

$$\mathfrak{b}_0 = T_s\delta_{t\cdot} u_0\delta_{\chi-} u_0 - E_s I_s\delta_{t\cdot} u_0\delta_{\chi-}\delta_{\chi\chi} u_0 + E_s I_s\delta_{t\cdot}\delta_{\chi+} u_0\delta_{\chi\chi} u_0 + 2\sigma_{1,s}\rho_s A\delta_{t\cdot} u_0\delta_{t-}\delta_{\chi-} u_0 \tag{4.45c}$$

$$\mathfrak{b}_N = T_s\delta_{t\cdot} u_N\delta_{\chi+} u_N - E_s I_s\delta_{t\cdot} u_N\delta_{\chi+}\delta_{\chi\chi} u_N + E_s I_s\delta_{t\cdot}\delta_{\chi-} u_N\delta_{\chi\chi} u_N + 2\sigma_{1,s}\rho_s A\delta_{t\cdot} u_N\delta_{t-}\delta_{\chi+} u_N. \tag{4.45d}$$

Notice that an additional energy term with negative sign has appeared, due to the non-centred discretisation of the time operator in the frequency dependent loss term.

When the discrete boundary conditions

$$u = 0, \qquad \delta_{\chi\chi} u = 0 \tag{4.46}$$

are applied at both ends of the string, the boundary terms $\mathfrak{b}_0$ and $\mathfrak{b}_N$ disappear identically. These correspond to the conditions (4.42) given in the continuous case.

A stability conditions for the scheme can be obtained either via energy techniques or von Neumann analysis. In both cases, the minimum grid step is given by [19]

$$h^2 \geq h_{\min}^2 = \frac{T_s k^2}{2\rho_s A} + \frac{k\sigma_{1,s}}{2} + \frac{k}{2}\sqrt{\left(\frac{T_s k}{\rho_s A} + 4\sigma_{1,s}\right)^2 + \frac{16 E_s I_s}{\rho_s A}}. \tag{4.47}$$

4.3.2 String/membrane collision

The interaction between a string and a stiff membrane is described in this section. Consider a string defined over the 1D region $\mathcal{D}_s$ with coordinate $\chi \in [0, L_s]$, with transverse displacement $u(\chi, t)$, and a membrane defined over a region $\mathcal{D}_m$ with Cartesian coordinates $\mathbf{x} = (x, y)$, with transverse displacement $w(\mathbf{x}, t)$, and $t \in \mathbb{R}^+$. The string interacts with the membrane by colliding against it from below. It is also assumed that the string be very thin, such that the region of collision can be considered one-dimensional. The equations for the system can be written as

$$\rho_m H \partial_t^2 w = \mathcal{O}_m w + \int_{\mathcal{D}_s} g(\chi, \mathbf{x}) \mathcal{F} \, \mathrm{d}\chi \tag{4.48a}$$

$$\rho_s A \partial_t^2 u = \mathcal{O}_s u - \mathcal{F}, \tag{4.48b}$$

where the operators $\mathcal{O}_m$ and $\mathcal{O}_s$ have been defined in (2.64) and (4.39), respectively. By setting $T_m = 0$ or $D = 0$ in the membrane equation one can simulate either an ideal membrane or a linear plate, without changing the core of the following derivation. The same holds for T_s and E_s in the string equation. This system of equations must be complemented with suitable boundary conditions. Boundary conditions for the stiff membrane have been presented in Section 3.3, and they are still valid in this case. Although the conditions for the string in isolation have been discussed in Section 4.3.1, the present case is more delicate, and is given special attention below. As a matter of fact, different possibilities arise depending on where the ends of the string are fixed. One can either fix them at the rim of the membrane, or attach them directly to the interior of $\mathcal{D}_m$. Another possibility is to introduce a bridge that connects the string to the rim of the membrane, as in the case of the snare drum, but this option would require an even more elaborate model, and will therefore not be considered here. The term $\mathcal{F} = \mathcal{F}(\chi)$ appearing in (4.48) is the distributed collision force density defined along the string, and $g(\chi, \mathbf{x})$ is a distribution function that maps the points along the string onto the membrane, and vice versa. This requires the introduction of a function $\boldsymbol{\pi} = (\pi_x, \pi_y) : \mathcal{D}_s \to \mathcal{D}_m$ that projects every point of the string onto the corresponding point on the membrane, as shown in Figure 4.3. The natural choice for the distribution g then becomes

$$g(\chi, \mathbf{x}) = \delta^2(\mathbf{x} - \boldsymbol{\pi}(\chi)), \tag{4.49}$$

where δ^2 is the two dimensional Dirac delta. The integral in (4.48a) is necessary in order to take into account the contribution from the entire length of the string. The collision force density $\mathcal{F}$ may be related to a distributed potential $\Pi(\zeta)$ in a similar way to the lumped cases

$$\mathcal{F} = \frac{\mathrm{d}\Pi}{\mathrm{d}\zeta}\,, \qquad \zeta = u - \int_{\mathcal{D}_m} g\, w \,\mathrm{d}\sigma, \tag{4.50}$$

where $\zeta(\chi)$ is a measure of the string/membrane interpenetration at position χ along the string. The integral in the second expression is required to select the point on the membrane corresponding to the projection $\boldsymbol{\pi}(\chi)$, such that

$$\int_{\mathcal{D}_m} g(\chi, \mathbf{x})\, w(\mathbf{x}, t)\,\mathrm{d}\sigma = \int_{\mathcal{D}_m} \delta^2(\mathbf{x} - \boldsymbol{\pi}(\chi)) w(\mathbf{x}, t)\,\mathrm{d}\sigma = w(\boldsymbol{\pi}(\chi), t). \tag{4.51}$$

An energy balance for the system can be obtained by multiplying each equation in (4.48) by the velocity and integrating over the domain

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{sm}}}{\mathrm{d}t} + \mathfrak{Q}_m = \int_{\mathcal{D}_m} \partial_t w \left(\int_{\mathcal{D}_s} g\,\mathcal{F}\,\mathrm{d}\chi \right) \mathrm{d}\sigma \tag{4.52a}$$

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{s}}}{\mathrm{d}t} + \mathfrak{Q}_s = -\int_{\mathcal{D}_s} \partial_t u\,\mathcal{F}\,\mathrm{d}\chi. \tag{4.52b}$$

Here, $\mathfrak{H}_{\mathrm{sm}}$ is the energy for a stiff membrane defined in (2.63), $\mathfrak{H}_{\mathrm{s}}$ is the string energy (4.41a), while $\mathfrak{Q}_m$ and $\mathfrak{Q}_s$ represent the loss terms produced by the energy analysis of the membrane and the string, respectively, see (2.59) and (4.41b). Summing together these two expressions, one then obtains

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{sm}}}{\mathrm{d}t} + \frac{\mathrm{d}\mathfrak{H}_{\mathrm{s}}}{\mathrm{d}t} + \mathfrak{Q}_m + \mathfrak{Q}_s = \int_{\mathcal{D}_m} \left(\int_{\mathcal{D}_s} \partial_t w\, g\,\mathcal{F}\,\mathrm{d}\chi \right) \mathrm{d}\sigma - \int_{\mathcal{D}_s} \partial_t u\,\mathcal{F}\,\mathrm{d}\chi \tag{4.53a}$$

$$= \int_{\mathcal{D}_s} \left(\int_{\mathcal{D}_m} g\,\partial_t w\,\mathrm{d}\sigma \right) \mathcal{F}\,\mathrm{d}\chi - \int_{\mathcal{D}_s} \partial_t u \mathcal{F}\,\mathrm{d}\chi \tag{4.53b}$$

$$\overset{(4.50)}{=} -\int_{\mathcal{D}_s} \partial_t \zeta\, \frac{\mathrm{d}\Pi}{\mathrm{d}\zeta}\,\mathrm{d}\chi \tag{4.53c}$$

$$= -\frac{\mathrm{d}\Pi}{\mathrm{d}t}\,, \tag{4.53d}$$

where the order of integration in the second equation has been switched. An energy balance can be recovered with the form

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{tot}}}{\mathrm{d}t} = -\mathfrak{Q}_{\mathrm{tot}}, \qquad \mathfrak{H}_{\mathrm{tot}} = \mathfrak{H}_{\mathrm{sm}} + \mathfrak{H}_{\mathrm{s}} + \Pi, \qquad \mathfrak{Q}_{\mathrm{tot}} = \mathfrak{Q}_{\mathrm{sm}} + \mathfrak{Q}_{\mathrm{s}}. \tag{4.54}$$

4.3.3 Finite difference scheme

A finite difference scheme for the system (4.48) can be written as

$$\rho_m H \delta_{tt} w = \mathfrak{D}_m w + \langle g, \mathfrak{F} \rangle_{\mathfrak{D}_s} \tag{4.55a}$$

$$\rho_s A \delta_{tt} u = \mathfrak{D}_s u - \mathfrak{F} \tag{4.55b}$$

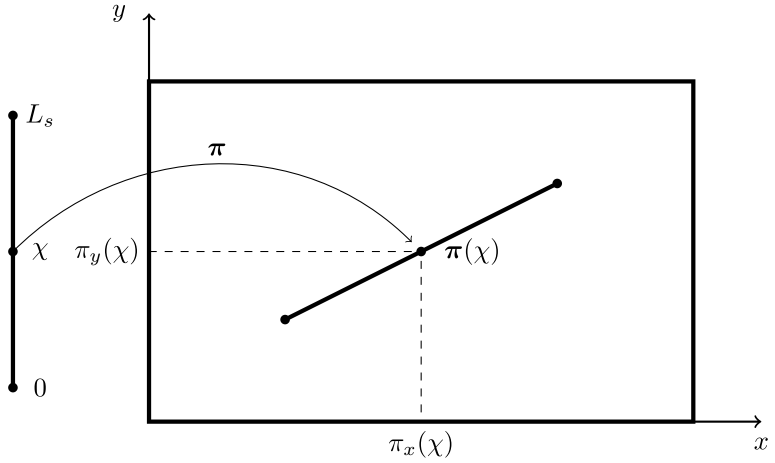


Figure 4.3: Function $\boldsymbol{\pi}$ projecting a point of the string onto the corresponding point on the membrane.

where the operators $\mathfrak{O}_m$ and $\mathfrak{O}_s$ have been defined in (2.165) and (4.43), respectively. The integral in (4.48a) has been replaced by a discrete inner product, and $\mathfrak{F}$ is the discrete counterpart of the collision force density $\mathcal{F}$. As for the explicit expression for the discrete distribution g, several possibilities are available, and they will be discussed shortly. Regardless of the particular choice of g, however, this inner product can be expressed as a linear operator

$$\langle g, \cdot \rangle_{\mathfrak{D}_s} = \mathfrak{i}^{\mathrm{1D}\rightarrow\mathrm{2D}}. \tag{4.56}$$

The operator $\mathfrak{i}^{\mathrm{1D}\rightarrow\mathrm{2D}}$ maps the variable $\mathfrak{F}$ defined over the 1D string domain $\mathfrak{D}_s$ onto a quantity which is defined over (a portion of) the 2D membrane domain $\mathfrak{D}_m$.

As before, the collision force density can be defined as

$$\mathfrak{F}^n = \frac{\delta_{t-}\Pi^{n+1/2}}{\delta_{t\cdot}\zeta^n}, \qquad \zeta^n = u^n - \mathfrak{i}^{\mathrm{2D}\rightarrow\mathrm{1D}} w^n, \qquad \Pi^{n+1/2} = \mu_{t+}\Pi^n, \tag{4.57}$$

where the operator $\mathfrak{i}^{\mathrm{2D}\rightarrow\mathrm{1D}}$ is a shorthand for the discrete inner product with g over the domain $\mathfrak{D}_m$,

$$\langle g, \cdot \rangle_{\mathfrak{D}_m} = \mathfrak{i}^{\mathrm{2D}\rightarrow\mathrm{1D}}. \tag{4.58}$$

Energy techniques can be applied also in this case, in order to arrive at an energy balance of the form

$$\delta_{t-}\mathfrak{h}_{\mathrm{tot}} = -\mathfrak{q}_{\mathrm{tot}}, \tag{4.59}$$

where $\mathfrak{h}_{\mathrm{tot}}$ represents the total energy of the system and $\mathfrak{q}_{\mathrm{tot}}$ is the total dissipated power.

Summing together the contributions from the membrane and the string gives

$$\begin{aligned}
\delta_{t-}\mathfrak{h}_{\mathrm{sm}} + \delta_{t-}\mathfrak{h}_{\mathrm{s}} + \mathfrak{q}_m + \mathfrak{q}_s &= \langle \delta_{t\cdot} w, \mathfrak{i}^{\mathrm{1D}\to\mathrm{2D}}\mathfrak{F}\rangle_{\mathfrak{D}_m} - \langle \delta_{t\cdot} u, \mathfrak{F}\rangle_{\mathfrak{D}_s} && \text{(4.60a)}\\
&= \langle \delta_{t\cdot} w, \left(\langle g, \mathfrak{F}\rangle_{\mathfrak{D}_s}\right)\rangle_{\mathfrak{D}_m} - \langle \delta_{t\cdot} u, \mathfrak{F}\rangle_{\mathfrak{D}_s} && \text{(4.60b)}\\
&= \langle \left(\langle \delta_{t\cdot} w, g\rangle_{\mathfrak{D}_m}\right), \mathfrak{F}\rangle_{\mathfrak{D}_s} - \langle \delta_{t\cdot} u, \mathfrak{F}\rangle_{\mathfrak{D}_s} && \text{(4.60c)}\\
&= -\langle \delta_{t\cdot}\zeta, \frac{\delta_{t-}\mu_{t+}\Pi}{\delta_{t\cdot}\zeta}\rangle_{\mathfrak{D}_s} && \text{(4.60d)}\\
&= -\delta_{t-}\langle 1, \mu_{t+}\Pi\rangle_{\mathfrak{D}_s} && \text{(4.60e)}
\end{aligned}$$

where the change in the order of the inner products descends from the properties of a finite sum, and corresponds to the change of order of integration in the continuous case. Hence, an energy balance similar to (4.54) is recovered.

4.3.4 Implementation

The implementation of the string/membrane interaction using the finite difference scheme described in the previous section requires a matrix/vector representation of the variables and of the operators. This useful representation in the 2D case has been discussed in Section 2.3.4. The system (4.55) can be written as

$$\begin{aligned}
\mathbf{w}^{n+1} &= \widetilde{\mathbf{w}} + \eta_m \mathbf{I}^{\mathrm{1D}\to\mathrm{2D}}\mathbf{f}^n && \text{(4.61a)}\\
\mathbf{u}^{n+1} &= \widetilde{\mathbf{u}} - \eta_s \mathbf{f}^n && \text{(4.61b)}
\end{aligned}$$

where the vectors **w** and **u** represent the membrane and string displacements and **f** is the vector of the collision force. The term $\widetilde{\mathbf{w}}$ represents the quantities known from the previous time step in the update for $\mathbf{w}^{n+1}$, see (2.167), and analogously for $\widetilde{\mathbf{u}}$. Finally, the matrix $\mathbf{I}^{\mathrm{1D}\to\mathrm{2D}}$ represents the action of the operator $\mathfrak{i}^{\mathrm{1D}\to\mathrm{2D}}$ and the coupling coefficients η_m and η_s are given by

$$\eta_m = \frac{k^2}{\rho_m H(1+\sigma_{0,m}k)}, \quad \eta_s = \frac{k^2}{\rho_s A(1+\sigma_{0,s}k)}. \tag{4.62}$$

The collision force $\mathfrak{F}$ and the interpenetration ζ become

$$\mathbf{f}^n = \frac{\delta_{t-}\mathbf{\Pi}^{n+1/2}}{\delta_{t\cdot}\boldsymbol{\zeta}^n}, \qquad \boldsymbol{\zeta}^n = \mathbf{u}^n - \mathbf{I}^{\mathrm{2D}\to\mathrm{1D}}\mathbf{w}^n \tag{4.63}$$

where $\mathbf{I}^{\mathrm{2D}\to\mathrm{1D}}$ is the matrix representation of the operator $\mathfrak{i}^{\mathrm{2D}\to\mathrm{1D}}$. The division between the two vectors in the definition of $\mathbf{f}^n$ is to be interpreted as a pointwise operation involving the entries of the vectors. Notice that, as in the previous cases, the collision force depends on the values for $\mathbf{w}^{n+1}$ and $\mathbf{u}^{n+1}$ yet to be calculated. In order to solve (4.61), one can left multiply (4.61a) by $\mathbf{I}^{\mathrm{2D}\to\mathrm{1D}}$ and subtract this from (4.61b). This leads to the following equation:

$$\mathbf{u}^{n+1} - \mathbf{I}^{\mathrm{2D}\to\mathrm{1D}}\mathbf{w}^{n+1} = \widetilde{\mathbf{u}} - \mathbf{I}^{\mathrm{2D}\to\mathrm{1D}}\widetilde{\mathbf{w}} - \mathbf{G}\mathbf{f}^n, \tag{4.64}$$

where

$$\mathbf{G} = \eta_m \mathbf{I}^{\mathrm{2D}\to\mathrm{1D}}\mathbf{I}^{\mathrm{1D}\to\mathrm{2D}} + \eta_s \mathbb{1}, \tag{4.65}$$

where $\mathbb{1}$ is the identity matrix. Rearranging all the terms on the left hand side, it is possible to arrive at the following nonlinear vector equation in the unknown $\mathbf{r} = \boldsymbol{\zeta}^{n+1} - \boldsymbol{\zeta}^{n-1}$:

$$\boldsymbol{\Gamma}(\mathbf{r}) = \mathbf{r} + \mathbf{G}\frac{\boldsymbol{\Pi}(\mathbf{r}+\mathbf{a}) - \boldsymbol{\Pi}(\mathbf{a})}{\mathbf{r}} + \mathbf{b} = \mathbf{0}, \tag{4.66}$$

which is formally similar to (4.37). Here, the various quantities are defined as

$$\mathbf{a} = \boldsymbol{\zeta}^{n-1}, \quad \mathbf{b} = -\widetilde{\mathbf{u}} + \mathbf{I}^{2\mathrm{D}\to 1\mathrm{D}}\widetilde{\mathbf{w}} + \boldsymbol{\zeta}^{n-1}. \tag{4.67}$$

The solution of this equation is discussed in the next section.

4.3.5 Newton-Raphson method for a vector equation

The key to the numerical simulation of the string-membrane interaction resides in the solution of the nonlinear equation (4.66). Newton-Raphson method can be adopted in this case as well, but it requires a slight variation with respect to the 1D approach presented before. Starting from an initial guess $\mathbf{r}_0$ for the solution of $\boldsymbol{\Gamma}(\mathbf{r}) = \mathbf{0}$, successive approximations can be obtained as

$$\mathbf{r}_{j+1} = \mathbf{r}_j - \mathbf{J}^{-1}(\mathbf{r}_j)\boldsymbol{\Gamma}(\mathbf{r}_j), \tag{4.68}$$

where $\mathbf{J}(\mathbf{r}_j)$ represents the Jacobian of $\boldsymbol{\Gamma}$.

The problem of existence and uniqueness of a solution in the vector case has been discussed in [27]. The matrix $\mathbf{G}$ defined in (4.65) is symmetric and positive definite, therefore invertible. The problem of solving $\boldsymbol{\Gamma} = \mathbf{0}$ is thus equivalent to the following:

$$\widetilde{\boldsymbol{\Gamma}}(\mathbf{r}) = \mathbf{G}^{-1}\mathbf{r} + \boldsymbol{\omega} + \mathbf{G}^{-1}\mathbf{b} = \mathbf{0}, \tag{4.69}$$

with

$$\boldsymbol{\omega} = \frac{\boldsymbol{\Pi}(\mathbf{r}+\mathbf{a}) - \boldsymbol{\Pi}(\mathbf{a})}{\mathbf{r}}. \tag{4.70}$$

The system $\widetilde{\boldsymbol{\Gamma}}$ possesses a unique solution if its Jacobian $\widetilde{\mathbf{J}}$ is positive definite. This, in turn, can be written as

$$\widetilde{\mathbf{J}} = \mathbf{G}^{-1} + \boldsymbol{\Omega}, \tag{4.71}$$

where $\boldsymbol{\Omega}$ represents the Jacobian of $\boldsymbol{\omega}$ and is a diagonal matrix. Each diagonal entry is either positive or zero, such that $\boldsymbol{\Omega}$ as a whole is non-negative. Thus, the Jacobian $\widetilde{\mathbf{J}}$ is positive definite, and the solution is unique.

4.3.6 String boundary conditions

One thing that so far has been overlooked is the boundary conditions for the string. One possibility is to attach the string at the rim of the membrane, such that fixed conditions (4.42) may be applied. In this case, the displacement of the string at those points is simply zero, and does not need to be updated.

A more delicate case is when the string is attached directly to the membrane, as this requires a more involved treatment. The equation of motion for the membrane needs to be modified in order to take into account the pointwise forces f_0 and f_{L_s} exerted at the end points of the string:

$$\rho_m H \partial_t^2 w = \mathcal{O}_m w + \int_{\mathcal{D}_s} g(\chi, \mathbf{x}) \mathcal{F} \, \mathrm{d}\chi + g_0 f_0 + g_{L_s} f_{L_s}, \tag{4.72}$$

where $g_0(x, y)$ and $g_{L_s}(x, y)$ are the distribution functions normalised to unity that represent the point of contact. The explicit expression for the force terms can be derived by applying once more energy methods. Summing together the contributions to the energy from the two components, the new energy balance reads

$$\frac{\mathrm{d}\mathfrak{H}_{\text{tot}}}{\mathrm{d}t} = -\mathfrak{Q}_{\text{tot}} + \langle \partial_t w, g_0 f_0 + g_{L_s} f_{L_s} \rangle_{\mathcal{D}_m} + \mathfrak{B}_s \Big|_0^{L_s} . \tag{4.73}$$

Now, the string boundary terms and the pointwise force terms at both ends must sum to zero, in order for the balance to be consistent. Considering for the moment only the end point at $\chi = 0$, one has

$$\langle \partial_t w, g_0 f_0 \rangle_{\mathcal{D}_m} - \Big(T_s \partial_t u \, \partial_\chi u - E_s I_s \partial_t u \, \partial_\chi^3 u + E_s I_s \partial_{t\chi} u \, \partial_\chi^2 u + 2\sigma_{1,s} \rho_s A \partial_t u \, \partial_\chi u \Big) \Big|_0 = 0. \tag{4.74}$$

As a first condition, one can impose $\partial_{\chi\chi} u \Big|_0 = 0$ for the string, in line with the simply supported condition applied before. This removes the third member in the string boundary term, and one is left with

$$\langle \partial_t w, g_0 f_0 \rangle_{\mathcal{D}_m} - \Big(T_s \partial_t u \, \partial_\chi u - E_s I_s \partial_t u \, \partial_\chi^3 u + 2\sigma_{1,s} \rho_s A \partial_t u \, \partial_\chi u \Big) \Big|_0 = 0. \tag{4.75}$$

As a second condition, it is legitimate to assume that, when the string is attached to the membrane, its position at $\chi = 0$ is the same as the contact point on the membrane, or

$$u\big|_0 = \langle w, g_0 \rangle_{\mathcal{D}_m} \quad \Rightarrow \quad \partial_t u\big|_0 = \langle \partial_t w, g_0 \rangle_{\mathcal{D}_m}. \tag{4.76}$$

This condition determines the value of f_0 in terms of the string position as

$$f_0 = T_s \partial_\chi u - E_s I_s \partial_\chi^3 u + 2\sigma_{1,s} \rho_s A \partial_\chi u. \tag{4.77}$$

When these three conditions are used, (4.74) is identically zero. Analogous conditions may be derived at the end point at $\chi = L_s$.

This analysis can guide in the implementation of a stable connection in the numerical case. The vector equation (4.61a) may be augmented with the contact forces f_0^n and $f_{L_s}^n$ such that

$$\mathbf{w}^{n+1} = \widetilde{\mathbf{w}} + \eta_m \mathbf{I}^{\mathrm{1D}\to\mathrm{2D}} \mathbf{f}^n + \eta_m \mathbf{g}_0 f_0^n + \eta_m \mathbf{g}_{L_s} f_{L_s}^n, \tag{4.78}$$

where $\mathbf{g}_0$ and $\mathbf{g}_{L_s}$ are vectors that represent the constant distribution functions g_0 and g_{L_s}. Considering now the end point at $\chi = 0$, the discrete counterparts of the three conditions presented before for the continuous case now read

$$\delta_{\chi\chi} u_0^n = 0 \tag{4.79a}$$

$$u_0^n = \langle w^n, g_0 \rangle_{\mathfrak{D}_m} \tag{4.79b}$$

$$f_0^n = T_s \delta_{\chi-} u_0^n - E_s I_s \delta_{\chi-} \delta_{\chi\chi} u_0^n + 2\sigma_{1,s} \rho_s A \delta_{t-} \delta_{\chi-} u_0^n. \tag{4.79c}$$

These can be easily derived through energy analysis techniques, similarly to the continuous case. Since the expression for f_0^n involves ghost points, its value cannot be calculated yet, but can formally be treated as an unknown.

The first step in the implementation of the numerical scheme is to consider (4.61b). At the end point, the update for u_0^{n+1} requires access to ghost points, which can be set in terms of (4.79a) and (4.79c). Notice that no collision force acts on this point, as it is attached to the membrane. The update for this point can thus be written as

$$u_0^{n+1} = -\frac{\eta_s}{h_s} f_0^n + v_0^n, \tag{4.80}$$

where v_0^n represents the terms known from the previous time step and the factor of h_s comes from the identity $\delta_{\chi\chi} = (\delta_{\chi+} - \delta_{\chi-})/h_s$. Using (4.79b), u_0^{n+1} can be written in terms of w^{n+1}, such that

$$h_m^2 \mathbf{g}_0^T \mathbf{w}^{n+1} = -\frac{\eta_s}{h_s} f_0^n + v_0^n, \tag{4.81}$$

where h_m is the grid step for the membrane grid and the inner product in (4.79b) has been rewritten in terms of a vector multiplication. Assuming that the end points of the string are sufficiently far apart and that $\mathbf{g}_0^T \mathbf{g}_{L_s} = 0$, one possible strategy is to left multiply (4.78) by $h_m^2 \mathbf{g}_0^T$ and to compare it with (4.81). Removing the left hand side from both equations, it is possible to express f_0^n in terms of the collision force $\mathbf{f}^n$, arriving at the following relation:

$$f_0^n = -\frac{1}{\overline{\eta}_0} \left(\boldsymbol{\nu}_0^T \mathbf{f}^n + \phi_0^n \right), \qquad \overline{\eta}_0 = \frac{\eta_s}{h_s} + \eta_m h_m^2 \left| \mathbf{g}_0 \right|^2, \tag{4.82}$$

where $\boldsymbol{\nu}_0$ is a constant vector and ϕ_0^n is a scalar that depends on known values from the previous time steps. A similar relation between $f_{L_s}^n$ and $\mathbf{f}^n$ can also be written. These new expressions for f_0^n and $f_{L_s}^n$ can be substituted into (4.78), and at this point the process outlined in Section 4.3.4 can be repeated. One arrives at the following nonlinear vector equation:

$$\boldsymbol{\Gamma}^\dagger(\mathbf{r}) = \mathbf{r} + \mathbf{G}^\dagger \frac{\boldsymbol{\Pi}(\mathbf{r} + \mathbf{a}) - \boldsymbol{\Pi}(\mathbf{a})}{\mathbf{r}} + \mathbf{b}^\dagger = \mathbf{0}, \tag{4.83}$$

with a new constant vector $\mathbf{b}^\dagger$ that depends on additional terms coming from the definitions of the end forces f_0^n and $f_{L_s}^n$ and with a new matrix $\mathbf{G}^\dagger$ defined as

$$\mathbf{G}^\dagger = \mathbf{I}^{2\text{D}\to 1\text{D}} \left(\eta_m \mathbf{I}^{1\text{D}\to 2\text{D}} - \frac{1}{\overline{\eta}_0} \mathbf{g}_0 \boldsymbol{\nu}_0^T - \frac{1}{\overline{\eta}_{L_s}} \mathbf{g}_{L_s} \boldsymbol{\nu}_{L_s}^T \right) + \eta_s \mathbb{1}. \tag{4.84}$$

The solution of (4.83) can now be tackled with the same techniques described in Section 4.3.5.

4.3.7 Simulation results

Consider an ideal square membrane with $L_x = L_y = 0.3$ m and wave speed $c_m = \sqrt{T_m/\rho_m H} = 62$ m/s with attached an ideal string with $L_s = 0.2$ m and wave speed $c_s = \sqrt{T_s/\rho_s A} = 50$ m/s. The parameters $K = 10^{11}$ and $\alpha = 1.5$ are used for the collision potential. The system is lossless, is simulated at a sampling rate of 44.1 kHz, and is initialised with a very short raised cosine strike (duration 0.3 ms) on the membrane. The evolution of the system in the very first instants is plotted in Figure 4.4, together with a close up of the string and of the portion of the membrane in contact with it. At first, the wave front generated by the raised cosine starts to propagate, and reaches the string after about 1 ms. When this happens, the string acts as an obstacle, with the wave bouncing back, while at the same time launching downwards a portion of the string. After 2.3 ms, the string is completely detached from the membrane, except of course for the two end points, where it is possible to notice the small ripples created by the downward pull of the string. Notice that the profile of the wave front is still very regular. At $t = 5.9$ ms, the string in its entirety collides against the membrane, generating a very complex pattern on the surface. The collision pushes the string away from the membrane again, reaching a maximum displacement at about 8 ms, with a new collision happening at $t = 10.4$ ms.

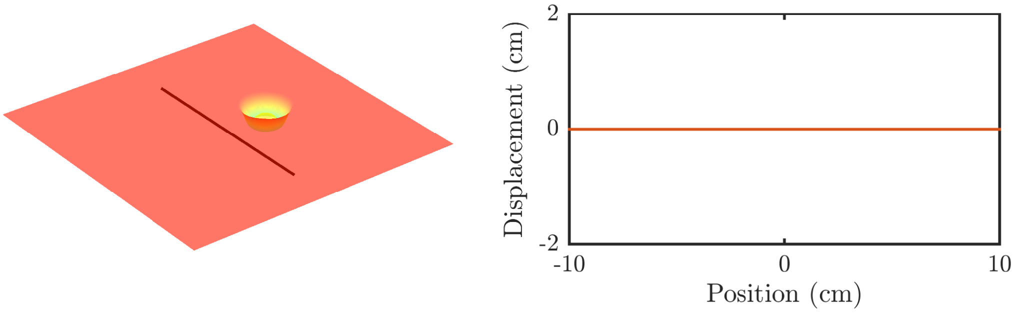
Energy conservation

From an energetic point of view, the system exhibits a continuous exchange of energy between the different components (membrane, string and collision potential), while the total energy remains constant to machine accuracy. This is shown in Figure 4.5, where the results for the first 0.1 s of simulation for the system described above are presented. Notice, in particular, that some portions of the initial part of the collision potential signal are missing, indicating that collision is not occurring. This is confirmed by the behaviour presented in Figure 4.4.

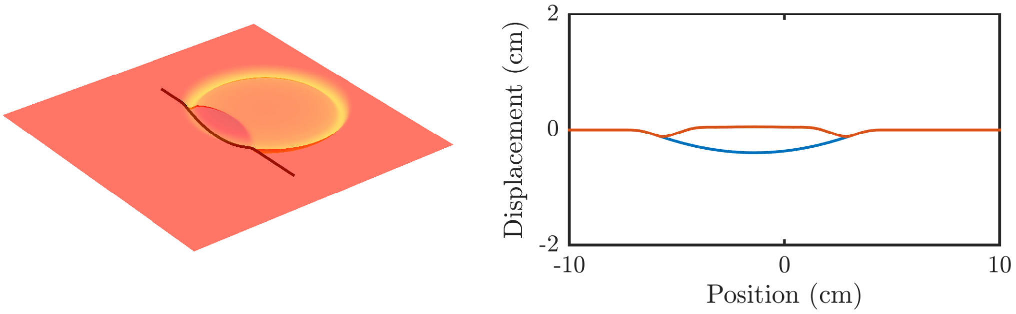
Spectrograms

The inclusion of a snare in contact with the drum membrane dates back at least to the Middle Ages, when the *tabor*, a forefather of the modern snare drum, was commonly adopted [31]. This expedient adds a buzzy and noisy timbre to the sound of the membrane in isolation. The rattling of the snare against the drumhead creates high frequency components that can be easily seen in the spectrogram.

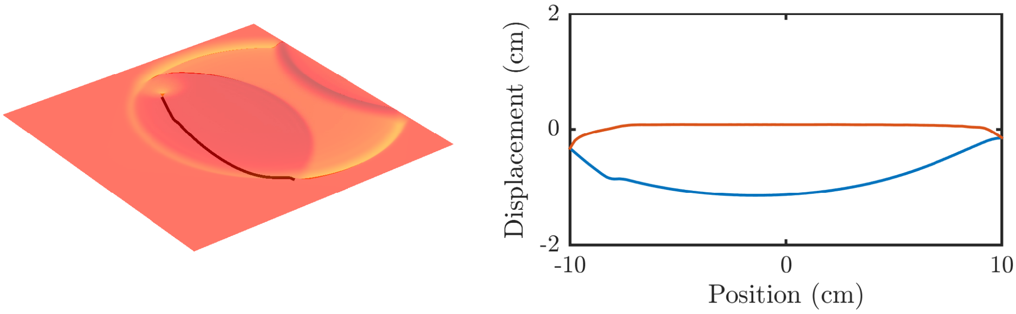
Consider, therefore, the system presented above, in the lossy case, and with a raised cosine excitation (see Section 6.5.1) with duration 3 ms and $F_{\max} = 100$ N. Figure 4.6 shows the spectrogram of the output for the single membrane and for the membrane/snare system. As is apparent, a noisy sound is produced when the snare is present.



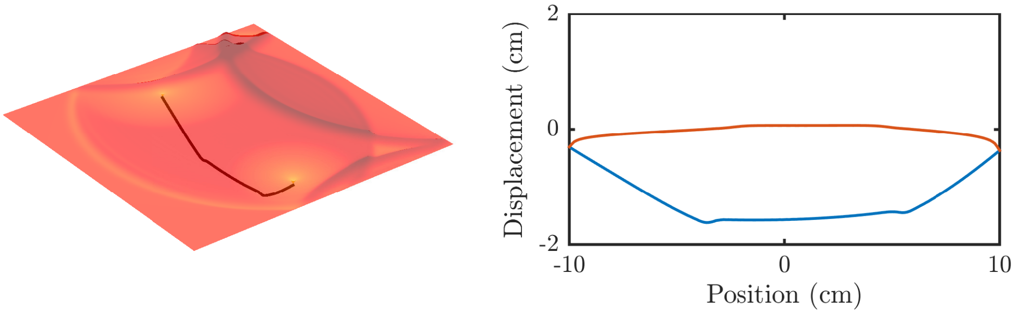
(a) $t = 0.5$ ms



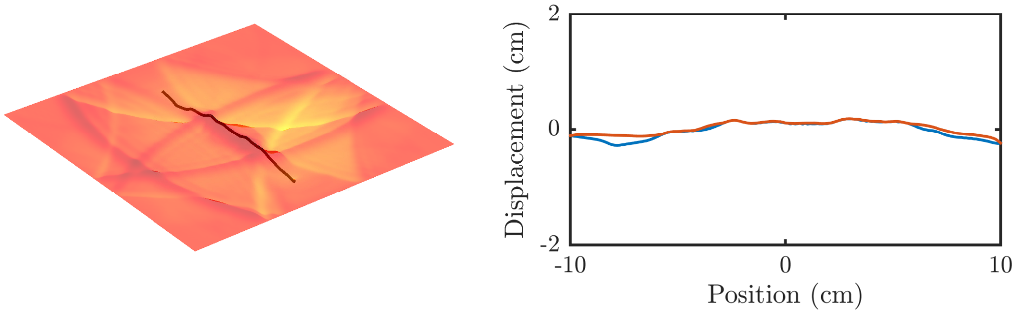
(b) $t = 1.4$ ms



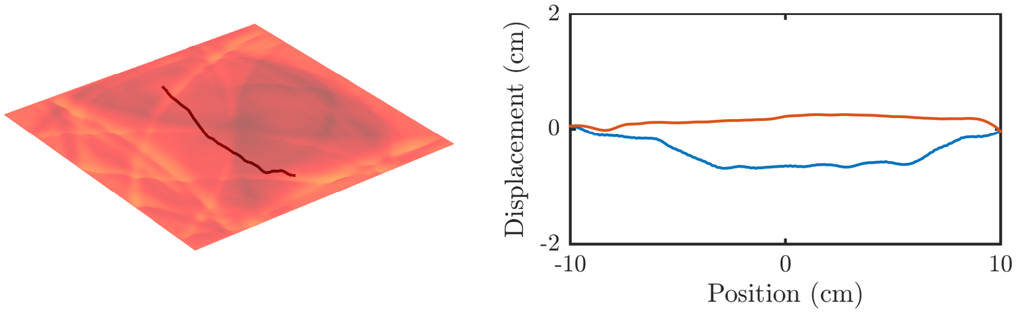
(c) $t = 2.3$ ms



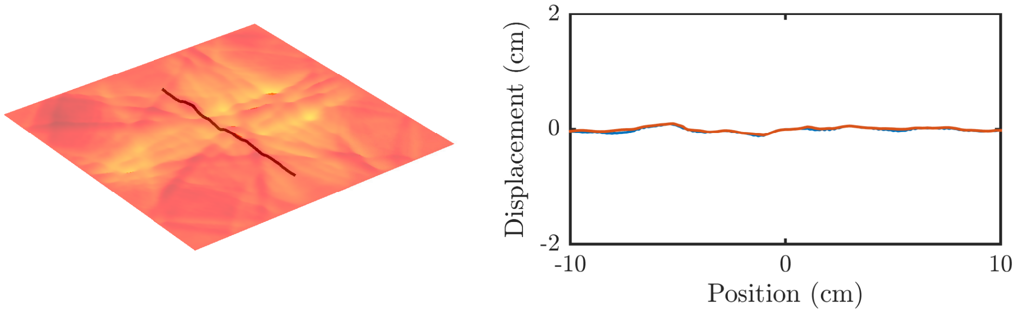
(d) $t = 3.2$ ms



(e) $t = 5.9$ ms



(f) $t = 7.7$ ms



(g) $t = 10.4$ ms

Figure 4.4: Evolution of the system described in Section 4.3.7, at times as indicated. Snapshots of the entire systems (left) are accompanied by the close up of the string deformation (right), together with the displacement of the membrane portion in contact with it. The simulation is run at a sampling rate of 44.1 kHz.

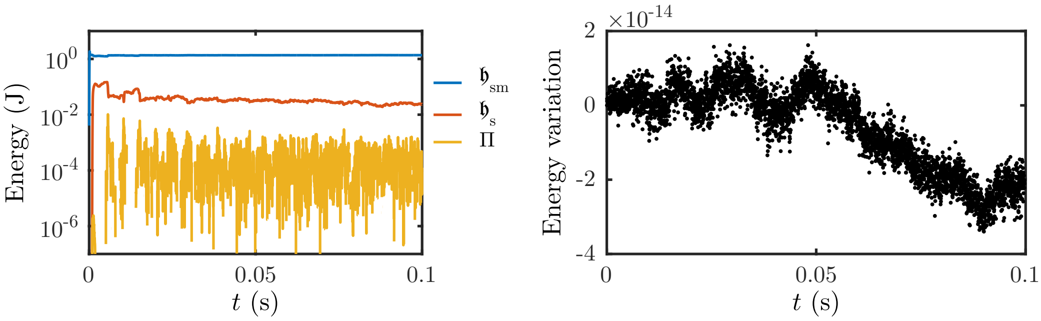
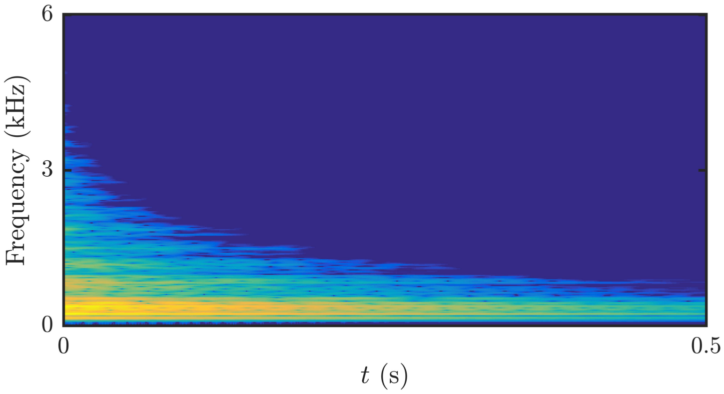
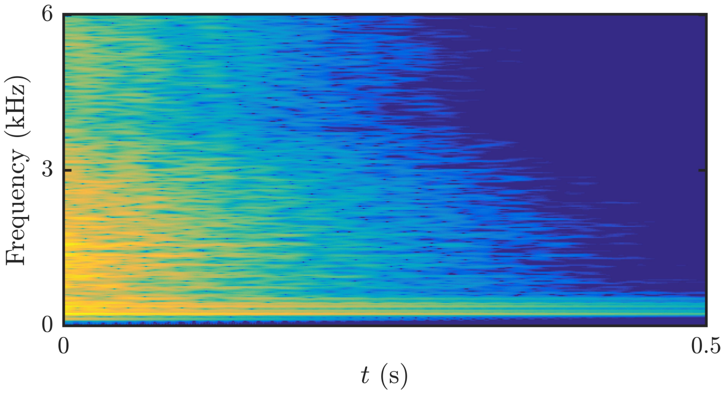


Figure 4.5: Left: Energy contributions from the various components of the string-membrane system under lossless conditions. Notice the different orders of magnitude of the energy values. Collisions can be detected by looking at the presence of the collision potential signal. Right: Normalised energy variations for the total energy of the system, calculated right after the end of the initial raised cosine excitation.



(a) Isolated membrane.



(b) Snare/membrane system.

Figure 4.6: Spectrograms of the output for the isolated membrane (above) and for the snare/membrane system (below). Including the snare in the system adds a noisy character to the membrane sound. The sampling rate of the simulation is 44.1 kHz.

Chapter 5

Coupling with the acoustic field in 3D

Musical instruments are never heard in isolation, and always rely on the presence of air for sound generation and propagation. The study of sound propagation in 3D is thus an important subject in musical acoustics. Several experimental investigations have been conducted on the sound radiation of various musical instruments, including drums. Studies conducted by Fleischer (as reported by Fletcher and Rossing in [84]) on timpani drums and by Rossing *et al.* on snare drums [153] confirm that the various modes produce different radiation patterns. These, in turn, are related to the efficiency with which energy is dissipated, and thus determine the decay rate of the various modes.

The presence of air coupling, besides providing a means for the instrument to radiate the sound, has also an influence on the frequencies produced. It is well known, in fact, that the modal frequencies of a circular membrane in isolation are not harmonically related. This, however, is in contrast with the sound that a timpani drum is able to produce, which has a determined pitch. Rayleigh was probably the first to recognise such phenomenon [169]. In this case, air loading effects and the presence of the cavity are responsible for altering the ratios between the $(m, 0)$ modes[1], with $m = 1, 2, 3\ldots$, bringing them close to a harmonic series. This has been the subject of a theoretical and experimental work by Christian *et al.* [60].

Numerical simulation has been adopted to study air coupling effects. This requires the instrument at hand to be modelled within a 3D acoustic space, with which it is coupled. Such process can be referred to as embedding. Not only does the acoustic field influence the behaviour of the embedded instrument, but also is the sound so generated more realistic. One of the first contributions focussing on the coupling between a 2D plate and the acoustic field is due to Botteldooren [32]. More complex interacting systems were later proposed, starting from the timpani drum model by Chaigne, Rhaouti and Joly [146, 52]. Here, the air/membrane coupling is implemented using the fictitious domain method. Other examples of coupling between percussion and air can be found within the framework of digital waveguides in the works of van Duyne [193], Laird [117] and Aird [2]. A modular percussion environment was proposed by Avanzini [6] to simulate a snare drum, where the interaction between the air and the membrane

---

[1] The modes that have m nodal diameters and a single nodal circle at the rim. A similar notation has been used in Section 2.2.5 for the modes of circular plates.

is simulated with a piston model. In recent years, other drum models have been proposed by Bilbao [20] and Bilbao and Webb [30] in the context of finite differences.

This chapter starts with a preliminary discussion on 3D notation and a brief introduction on the 3D wave equation and its numerical implementation. The coupling between a 2D structure, such as a stiff membrane, and the acoustic field is presented next—the linear case is discussed first, with a detailed analysis of the finite difference discretisation and the numerical implementation. This model is then extended to more complex systems involving the presence of nonlinearities and collisions.

5.1 Preliminaries III: 3D notation

In this section, the notation used in 2D is extended to 3D. Both the continuous and discrete cases are discussed here.

5.1.1 Continuous case

A 3D region of the Euclidean space $\mathbb{R}^3$ is denoted by $\mathcal{V}$. In this work, only regular, rectangular domains with sides L_x, L_y and L_z are considered, and a set of Cartesian coordinates $(x, y, z) \in \mathbb{R}^3$ is adopted. The Laplacian operator in 3D is defined as

$$\Delta^{(3)} = \partial_x^2 + \partial_y^2 + \partial_z^2. \tag{5.1}$$

As done in the previous chapters, whenever the dimension of the space is clear from the context, the superscript (3) will be dropped from the Laplacian symbol. The divergence and gradient operators can be defined in an analogous way to 2D (see (2.15)) for any vector $\mathbf{v} = (v_x, v_y, v_z)^T$ and function $f(x, y, z)$, respectively, as

$$\nabla \cdot \mathbf{v} = \partial_x v_x + \partial_y v_y + \partial_z v_z, \qquad \boldsymbol{\nabla} f = (\partial_x f, \partial_y f, \partial_z f)^T. \tag{5.2}$$

The 3D inner product and its associated norm for any two functions $f(x, y, z, t)$ and $g(x, y, z, t)$ are defined as

$$\ll f, g \gg_{\mathcal{V}} = \int_{\mathcal{V}} f \, g \; \mathrm{d}\mathbf{x}, \qquad \|f\|_{\mathcal{V}} = \sqrt{\ll f, f \gg_{\mathcal{V}}}, \tag{5.3}$$

where $\mathrm{d}\mathbf{x}$ is the infinitesimal volume element.

5.1.2 Discrete case

Numerical simulation of 3D spaces using the finite difference method can be performed using an analogous notation to the 2D case, see Section 2.3. Here, a 3D discrete function $f^n_{l_x,l_y,l_z}$ represents an approximation of $f(x, y, z, t)$ at position $(x = l_x h, y = l_y h, z = l_z h)$ and time $t = nk$, where h and k are the usual notation for the spatial and temporal grid steps, respectively. The indices (l_x, l_y, l_z) are restricted to the set of integers such that $(l_x h, l_y h, l_z h) \in \mathcal{V}$, and the finite difference domain is denoted by $\mathfrak{V}$. Let $N_x + 1$, $N_y + 1$ and $N_z + 1$ be the number of grid points along each direction.

New shift operators s_{z+} and s_{z-} defined as

$$\mathrm{s}_{z+} f^n_{l_x,l_y,l_z} = f^n_{l_x,l_y,l_z+1}, \qquad \mathrm{s}_{z-} f^n_{l_x,l_y,l_z} = f^n_{l_x,l_y,l_z-1}, \tag{5.4}$$

can be combined to obtain the various difference operators in the z direction. The discrete counterpart of the Laplacian operator $\Delta^{(3)}$ can be defined as

$$\delta^{(3)}_{\Delta} = \delta_{xx} + \delta_{yy} + \delta_{zz} = \frac{\mathrm{s}_{x-} + \mathrm{s}_{y-} + \mathrm{s}_{z-} - 6 + \mathrm{s}_{x+} + \mathrm{s}_{y+} + \mathrm{s}_{z+}}{h^2}. \tag{5.5}$$

The 3D discrete inner product and associated norm can be defined for any two grid functions $f^n_{l_x,l_y,l_z}$ and $g^n_{l_x,l_y,l_z}$ over the domain $\mathfrak{V}$ as

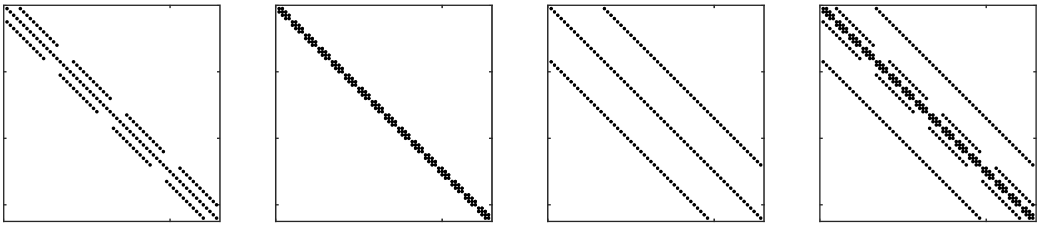
$$\ll f, g \gg_{\mathfrak{V}} = \sum_{(l_x,l_y,l_z)\in\mathfrak{V}} h^3 f^n_{l_x,l_y,l_z} g^n_{l_x,l_y,l_z}, \qquad \|f\|_{\mathfrak{V}} = \sqrt{\ll f, f \gg_{\mathfrak{V}}}. \tag{5.6}$$

Notice the presence of the factor h^3 that takes into account the volume of a grid cell.

In implementation, it is sometimes convenient to recast the schemes into matrix-vector form, as has been explained in Section 2.3.4. The vector form for the discrete grid variable can be obtained by rearranging the values into a column vector, and the matrix representation of the finite difference operators can be constructed using the Kronecker product of simple matrices. Starting from the simple matrix $\mathbf{D}_{xx}$ given in (2.114), and constructing $\mathbf{D}_{yy}$ and $\mathbf{D}_{zz}$ in a similar way, the operators δ_{xx}, δ_{yy}, δ_{zz} and $\delta^{(3)}_{\Delta}$ acting on a 3D grid can be transformed into matrix form as

$$\mathbf{D}^{(3)}_{xx} = \mathbb{1}_{N_z+1} \otimes (\mathbf{D}_{xx} \otimes \mathbb{1}_{N_y+1}) \tag{5.7a}$$
$$\mathbf{D}^{(3)}_{yy} = \mathbb{1}_{N_z+1} \otimes (\mathbb{1}_{N_x+1} \otimes \mathbf{D}_{yy}) \tag{5.7b}$$
$$\mathbf{D}^{(3)}_{zz} = \mathbf{D}_{zz} \otimes (\mathbb{1}_{N_x+1} \otimes \mathbb{1}_{N_y+1}) \tag{5.7c}$$
$$\mathbf{D}^{(3)}_{\Delta} = \mathbf{D}^{(3)}_{xx} + \mathbf{D}^{(3)}_{yy} + \mathbf{D}^{(3)}_{zz}. \tag{5.7d}$$

Notice that the terms inside the parentheses in the definitions of $\mathbf{D}^{(3)}_{xx}$ and $\mathbf{D}^{(3)}_{yy}$ are the 2D matrices $\mathbf{D}^{(2)}_{xx}$ and $\mathbf{D}^{(2)}_{yy}$ defined in (2.115). The sparsity patterns for these matrices are plotted in Figure 5.1.



(a) $\mathbf{D}^{(3)}_{xx}$ (b) $\mathbf{D}^{(3)}_{yy}$ (c) $\mathbf{D}^{(3)}_{zz}$ (d) $\mathbf{D}^{(3)}_{\Delta}$

Figure 5.1: Sparsity plots for the finite difference operators in matrix form for a 3D grid with 4 × 4 × 4 points.

5.2 Basic equations

Consider a 3D space with Cartesian coordinates (x, y, z). It is customary to describe the acoustic field by two variables, $p(x, y, z, t)$ and $\mathbf{v}(x, y, z, t)$, a scalar field and a vector field, respectively, both functions of space and time. The former is the pressure variation from atmospheric, while the latter represents the particle velocity of air. The equations of motion relating these two variables are given in all classic textbooks [135, 84], and can be written as

$$\partial_t p = -\rho_a c_a^2 \nabla \cdot \mathbf{v} \tag{5.8a}$$

$$\rho_a \partial_t \mathbf{v} = -\boldsymbol{\nabla} p, \tag{5.8b}$$

where ρ_a is the density of air and c_a is the sound speed. In this work, the values $\rho_a = 1.21$ kg/m^3 and $c_a = 340$ m/s will be used. These two equations can be combined together in a single equation, involving p only:

$$\partial_t^2 p = c_a^2 \Delta p, \tag{5.9}$$

where Δ is the 3D Laplacian operator (5.1). This model can be obtained as a simplified version of a more complex system, involving temperature as an extra variable [135, 40], but it is an adequate starting point for many applications, including the present work. Following Morse and Ingard [135], it is useful to introduce the velocity potential $\Psi(x, y, z, t)$, a scalar function of space and time, whose relation with p and $\mathbf{v}$ is given by

$$\mathbf{v} = -\boldsymbol{\nabla}\Psi, \qquad p = \rho_a \partial_t \Psi. \tag{5.10}$$

The equation of motion (5.9) in terms of the velocity potential Ψ has still the form of a wave equation

$$\partial_t^2 \Psi = c_a^2 \Delta \Psi, \tag{5.11}$$

and will be used in the remainder of this work as it considerably simplifies the implementation of coupling conditions.

5.2.1 Energy and lossless boundary conditions

When a finite 3D domain is considered, (5.11) must be complemented with suitable boundary conditions. Two types of conditions, reflective and absorbing, will be considered in this work. In the first case, a wave travelling towards a wall of the domain is completely reflected, while in the second case it is totally absorbed. This latter case is useful when one is to model an infinite, boundless space with numerical methods. In real spaces, a combination of both behaviours is found, with sound waves being partially reflected and partially absorbed by the walls. This obviously produces several challenges from a simulation point of view, and the research on this topic is still very active [115, 21, 96].

Energy methods can be applied to derive suitable boundary conditions. Consider a 3D domain $\mathcal{V}$, with boundary surface $\partial\mathcal{V}$. In order to find an energy balance for the system, multiply (5.11) by $(\rho_a/c_a^2)\partial_t\Psi$ (the factor ρ_a/c_a^2 is added to obtain the correct physical dimensions for energy) and integrate over $\mathcal{V}$

$$\frac{\rho_a}{c_a^2}\int_{\mathcal{V}} \partial_t\Psi\partial_t^2\Psi\,\mathrm{d}\mathbf{x} = \int_{\mathcal{V}} \rho_a\partial_t\Psi\Delta\Psi\,\mathrm{d}\mathbf{x}. \tag{5.12}$$

The left hand side of this equation gives

$$\text{l.h.s.} = \frac{\rho_a}{c_a^2}\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{V}} |\partial_t\Psi|^2\,\mathrm{d}\mathbf{x}, \tag{5.13}$$

while the right hand side can be manipulated in the following way:

$$\begin{aligned}\text{r.h.s} &= -\int_{\mathcal{V}} \rho_a\partial_t\boldsymbol{\nabla}\Psi\cdot\boldsymbol{\nabla}\Psi\,\mathrm{d}\mathbf{x} + \oint_{\partial\mathcal{V}} \rho_a\mathbf{n}\cdot(\partial_t\Psi\boldsymbol{\nabla}\Psi)\,\mathrm{d}\sigma && \text{(5.14a)}\\ &= -\frac{\rho_a}{2}\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{V}} |\boldsymbol{\nabla}\Psi|^2\,\mathrm{d}\mathbf{x} + \mathfrak{B}_a, && \text{(5.14b)}\end{aligned}$$

where $\mathbf{n}$ is the unit vector normal to the boundary surface and pointing outwards. Combining the results, it is possible to write the energy balance

$$\frac{\mathrm{d}\mathfrak{H}_a}{\mathrm{d}t} = \mathfrak{B}_a, \tag{5.15}$$

with the energy of the acoustic field and boundary terms given by

$$\mathfrak{H}_a = \frac{\rho_a}{2c_a^2}\left\|\partial_t\Psi\right\|_{\mathcal{V}}^2 + \frac{\rho_a}{2}\left\|\boldsymbol{\nabla}\Psi\right\|_{\mathcal{V}}^2 \tag{5.16}$$

and

$$\mathfrak{B}_a = \rho_a\oint_{\partial\mathcal{V}} \mathbf{n}\cdot(\partial_t\Psi\,\boldsymbol{\nabla}\Psi)\,\mathrm{d}\sigma. \tag{5.17}$$

In order for the energy to be conserved, $\mathfrak{B}_a$ must identically vanish over the boundary $\partial\mathcal{V}$. To this end, either a Dirichlet or Neumann type condition satisfies this requirement

$$\begin{aligned}\Psi|_{\partial\mathcal{V}} &= 0 \qquad \text{Dirichlet}, && \text{(5.18a)}\\ \mathbf{n}\cdot\boldsymbol{\nabla}\Psi|_{\partial\mathcal{V}} &= 0 \qquad \text{Neumann}. && \text{(5.18b)}\end{aligned}$$

The reflective condition (5.18a) will be often used in the remainder of this work to check the energy conservation of the numerical scheme. Neumann condition can be applied to simulate the presence of a rigid wall, and will be used for modelling the cavity of drums in Chapter 6.

5.2.2 Absorbing boundary conditions

In some circumstances, the simulation of an unbounded domain is required, while only finite spaces are amenable to numerical computation. In these cases, the classic approach is to use a bounded domain, and apply some conditions at the walls such that waves travelling towards the boundary are ideally totally absorbed, without being reflected back. One popular choice in acoustics and electrodynamics is the use of perfectly matched layers (PMLs), due in their original formulation to Berenger [13]. Another widely adopted approach is the use of Enguist-Majda absorbing boundary conditions (ABCs) [78]. Consider the real half space $\mathbb{R}^{3,x+}$, such that $x \geq 0$, and define the d'Alembert operator as

$$\Box = \partial_x^2 + \partial_y^2 + \partial_z^2 - \frac{1}{c^2}\partial_t^2. \tag{5.19}$$

The 3D wave equation (5.11) is then equivalent to the following:

$$\Box\Psi = 0. \tag{5.20}$$

The d'Alembert operator can be factored into two components,

$$\Box = \Box^+ \Box^-, \quad \text{with} \quad \Box^\pm = \partial_x \pm \frac{\partial_t}{c}\sqrt{1 - \left(\frac{\partial_y}{\partial_t/c}\right)^2 - \left(\frac{\partial_z}{\partial_t/c}\right)^2}. \tag{5.21}$$

Over the surface at $x = 0$, the operator $\Box^-$ completely absorbs any wave travelling towards the boundary, no matter the angle of incidence [172]. Thus,

$$\Box^- \Psi|_{x=0} = 0 \tag{5.22}$$

is an exact analytical absorbing boundary condition, and corresponds to Sommerfeld's radiation condition [166]. For a finite box, this condition can be easily extended by symmetry considerations to the various faces.

First order approximation

The presence of a square root in the definition of $\Box^-$ prevents it from being implemented straightforwardly in a numerical simulation [172]. The operator, in fact, is non-local both in space and time [78]. The idea of Engquist and Majda [78] was to consider a family of increasingly accurate conditions based on different approximations of the square root. As well known, a Taylor series expansion of the function $\sqrt{1-x^2}$ around zero gives

$$\sqrt{1-x^2} \approx 1 - \frac{x^2}{2} - \frac{x^4}{8} + O(x^6). \tag{5.23}$$

A first order approximation to (5.22) at $x = 0$ can be obtained by retaining only the first term in the square root expansion, that is

$$\left(\partial_x - \frac{\partial_t}{c_a}\right)\Psi|_{x=0} = 0. \tag{5.24}$$

Over the face at $x = 0$, the boundary term (5.17) produced by the energy analysis performed above can be written as

$$\mathfrak{B}_a = -\rho_a \int_{x=0} \partial_x\Psi\, \partial_t\Psi\, \mathrm{d}\sigma. \tag{5.25}$$

When the first order condition (5.24) is employed, $\mathfrak{B}_a$ becomes a perfectly dissipative term,

$$\mathfrak{B}_a = -\frac{\rho_a}{c_a}\left\|\partial_t\Psi\right\|_{x=0}^2. \tag{5.26}$$

Higher order absorbing conditions can also be adopted [78]. In this case, however, it is difficult to devise a way to implement them within an energetic framework, where the resulting boundary terms $\mathfrak{B}_a$ are manifestly dissipative.

5.3 Numerical simulation

5.3.1 Finite difference scheme and implementation

The simplest finite difference scheme for the 3D wave equation (5.11) is the following [86]:

$$\delta_{tt}\Psi = c_a^2 \delta_{\Delta}^{(3)} \Psi \tag{5.27}$$

where $\delta_{\Delta}^{(3)}$ represents an approximation for the Laplacian operator. Various stencils can be adopted [116], the simplest of which is the standard leapfrog scheme defined in (5.5).

In order to simplify the implementation, the 3D discrete variable Ψ^n can be rearranged into a vector, which will be denoted by $\mathbf{\Psi}^n$, as discussed in Section 2.3.4. The update for the scheme can thus be written as

$$\mathbf{\Psi}^{n+1} = \mathbf{B}\mathbf{\Psi}^n - \mathbf{\Psi}^{n-1}, \qquad \mathbf{B} = 2\,\mathbb{1} + c_a^2 k^2 \mathbf{D}_{\Delta}^{(3)}, \tag{5.28}$$

where $\mathbb{1}$ is the identity matrix and $\mathbf{D}_{\Delta}^{(3)}$ is the 3D Laplacian matrix defined in (5.7d).

5.3.2 Energy and boundary conditions

The numerical energy associated with this scheme can be obtained with the usual technique, by multiplying (5.27) by $\delta_{t\cdot}\Psi$ and taking an inner product over the entire space. For the infinite domain $\mathbb{Z}^3$, this leads to an energy balance of the form

$$\delta_{t+}\mathfrak{h}_{\mathrm{a}} = 0 \tag{5.29}$$

with discrete energy

$$\begin{aligned}\mathfrak{h}_{\mathrm{a}} = \frac{\rho_a}{2c_a^2}\left\|\delta_{t-}\Psi\right\|_{\mathbb{Z}^3}^2 + \frac{\rho_a}{2} \ll \delta_{x-}\Psi, \mathrm{s}_{t-}\delta_{x-}\Psi \gg_{\mathbb{Z}^3} \\ + \frac{\rho_a}{2} \ll \delta_{y-}\Psi, \mathrm{s}_{t-}\delta_{y-}\Psi \gg_{\mathbb{Z}^3} + \frac{\rho_a}{2} \ll \delta_{z-}\Psi, \mathrm{s}_{t-}\delta_{z-}\Psi \gg_{\mathbb{Z}^3} .\end{aligned} \tag{5.30}$$

When the domain is finite, however, boundary conditions appear. Consider, e.g., the semi-infinite space $\mathbb{Z}^{3,x+} = \{(l_x, l_y, l_z) \in \mathbb{Z}^3 | l_x \geq 0\}$. The new energy balance in this case becomes

$$\delta_{t+}\mathfrak{h}_{\mathrm{a}} = -\rho_a \langle \delta_{t\cdot}\Psi, \delta_{x-}\Psi \rangle_{\mathbb{Z}^2, l_x=0} = \mathfrak{b}_{\mathrm{a}}, \tag{5.31}$$

where $\mathfrak{b}_{\mathrm{a}}$ is the new boundary term. Two conditions can now be applied, corresponding to the Dirichlet (5.18a) and Neumann (5.18b) conditions given in the continuous case

$$\delta_{t\cdot}\Psi|_{l_x=0} = 0 \qquad \text{discrete Dirichlet,} \tag{5.32a}$$

$$\delta_{x-}\Psi|_{l_x=0} = 0 \qquad \text{discrete Neumann.} \tag{5.32b}$$

Notice, in fact, that the discrete Dirichlet condition is equivalent to $\Psi|_{l_x=0} = 0$ when the initial acoustic potential is zero at the boundaries. In both cases, the resulting scheme is perfectly lossless. These types of conditions will be referred to as reflective boundaries. The stability condition for the scheme can be derived with either energy methods or extensions of von Neumann analysis. In either case, the grid step h_a must satisfy the relation

$$h_a \geq \sqrt{3} c_a k. \tag{5.33}$$

Energy conservation

Consider the 3D scheme (5.27) over a box with $L_x = L_y = 0.4$ m and $L_z = 0.3$ m, initialised with a Dirac delta at the central point. Figure 5.2 shows the normalised energy variations for one second of simulation at sampling rate of 44.1 kHz with Dirichlet and Neumann boundary conditions. As expected, energy variations exhibit a random walk behaviour in both cases. In order to avoid rounding errors, however, care must be taken in implementation. See Appendix B.5 for further details.

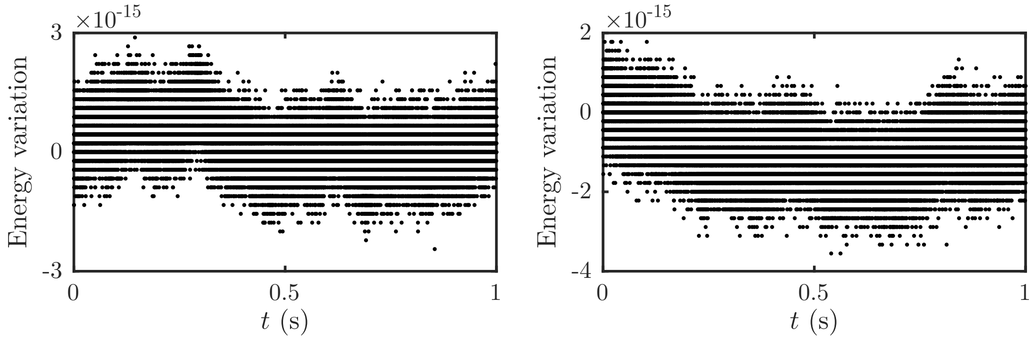


Figure 5.2: Energy conservation for the lossless 3D scheme (5.27) under Dirichlet (left) and Neumann (right) boundary conditions, see (5.32a) and (5.32b), respectively. The sampling rate is 44.1 kHz.

5.3.3 Absorbing conditions

A finite difference implementation of the first order absorbing boundary condition (5.24) can be written as

$$\left(c_a \delta_{x-}\Psi - \delta_{t\cdot}\Psi\right)\big|_{l_x=0} = 0. \tag{5.34}$$

Similarly to the continuous case, when this condition is inserted into the boundary term defined in (5.31), the total energy of the system is monotonically decreasing, given the negativity of $\mathfrak{b}_a$:

$$\delta_{t+}\mathfrak{h}_a = -\frac{\rho_a}{c_a} \left\|\delta_{t\cdot}\Psi\right\|^2_{l_x=0}. \tag{5.35}$$

5.3.4 Rigid boundaries

Rigid boundary conditions can be approximated by applying the Neumann condition (5.32b) to the points near the boundary. When the domain to be modelled has a circular geometry, as in the case of a drum shell (see Chapter 6), a staircase approximation can be used, similar to that described in Section 2.5.2. Figure 5.3 illustrates the problem. The update at the point labelled with A requires access to the points that lie outside of the domain, which are marked with a cross. The values of the acoustic potential Ψ^n at those points can be set by applying the conditions

$$\delta_{x-}\Psi^n_{\mathrm{A}} = 0, \qquad \text{and} \qquad \delta_{y+}\Psi^n_{\mathrm{A}} = 0 \tag{5.36}$$

at A.

Alternative approaches, like the fictitious domain method [52] and the finite volume method [21] can also be adopted in order to achieve more accurate approximations.

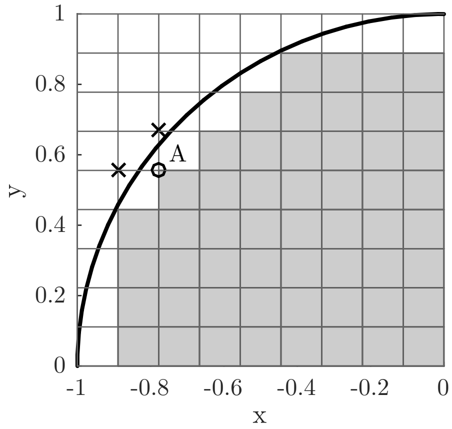


Figure 5.3: Staircase approximation for the acoustic field within a rigid circular boundary. The update for the point labelled with A requires two ghost points that lie outside of the cavity. Neumann conditions specify the values of these ghost points in terms of the value at A.

5.4 Embeddings

In the following sections, the coupling between the acoustic field and 2D structures is presented.

5.4.1 Air coupling with a linear membrane

Energy techniques can be applied in order to describe the coupling between a 2D flat structure, like a plate or a membrane, and the surrounding 3D acoustic field. A similar approach has been adopted by Rhaouti *et al.* [146] and by Bilbao [20, 29]. A lossy stiff membrane can be considered as the most general case, see Section 2.2.3. Let such 2D surface be defined for the moment over the infinite plane parallel to the xy region at vertical coordinate z_0. The equation of motion can be written as

$$\rho H \partial_t^2 w = \mathcal{O}_m w + p^+ + p^-, \tag{5.37}$$

where the left hand side is the inertial term of the equation, with ρ the density and H the thickness. The symbol $\mathcal{O}_m$ groups together all the linear terms acting on w, and has been defined in (2.64). Finally, p^+ and p^- represent the air pressure acting above and below the structure, respectively. The geometry of the problem is schematically represented in Figure 5.4.

The energy $\mathfrak{H}_{\mathrm{sm}}$ for the stiff membrane can be obtained using the techniques described in Chapter 2 by multiplying (5.37) by $\partial_t w$ and integrating over the entire plane:

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{sm}}}{\mathrm{d}t} = \int_{\mathbb{R}^2} \partial_t w \left(p^+ + p^-\right) \mathrm{d}\sigma. \tag{5.38}$$

The energy variation for the membrane is not zero in this case, rather, it is related to the power injected into or extracted from the membrane by the air pressure.

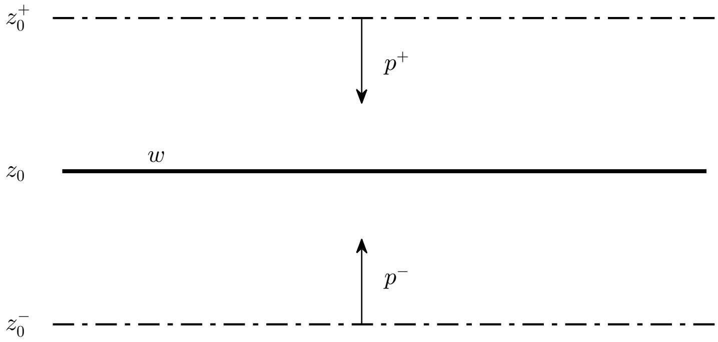


Figure 5.4: Representation of the air coupling geometry. Membrane profile at $z = z_0$, with pressures p^+ and p^- acting at positions z_0^+ and z_0^-, respectively.

Consider now the acoustic field. The 3D space is split in half by the presence of the 2D structure, and this fact must be taken into account in the energy analysis. Following the calculation of (5.12), in fact, one obtains

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{a}}}{\mathrm{d}t} = -\int_{\mathbb{R}^2\times\{z_0^+\}} \rho_a \partial_t\Psi\partial_z\Psi \,\mathrm{d}\sigma + \int_{\mathbb{R}^2\times\{z_0^-\}} \rho_a \partial_t\Psi\partial_z\Psi \,\mathrm{d}\sigma, \tag{5.39}$$

where the two integrals are calculated over the two infinite regions immediately above and below the coordinate $z = z_0$, where the 2D structure is defined.

Considering the air and the 2D structure as a whole system, it is clear that the total energy $\mathfrak{H}_{\mathrm{tot}}$ must be conserved, meaning that

$$\frac{\mathrm{d}\mathfrak{H}_{\mathrm{tot}}}{\mathrm{d}t} = \frac{\mathrm{d}\mathfrak{H}_{\mathrm{sm}}}{\mathrm{d}t} + \frac{\mathrm{d}\mathfrak{H}_{\mathrm{a}}}{\mathrm{d}t} = 0. \tag{5.40}$$

In order for this to happen, the terms on the right hand side of equations (5.38) and (5.39) must sum to zero. In other words,

$$\int_{\mathbb{R}^2} \partial_t w\, p^+ \,\mathrm{d}\sigma - \int_{\mathbb{R}^2\times\{z_0^+\}} \rho_a \partial_t\Psi\partial_z\Psi \,\mathrm{d}\sigma = 0, \tag{5.41a}$$

$$\int_{\mathbb{R}^2} \partial_t w\, p^- \,\mathrm{d}\sigma + \int_{\mathbb{R}^2\times\{z_0^-\}} \rho_a \partial_t\Psi\partial_z\Psi \,\mathrm{d}\sigma = 0. \tag{5.41b}$$

Given the relations (5.10) between velocity, pressure and acoustic field Ψ, it is natural to impose the following conditions on Ψ and w

$$\partial_t w = -\,\partial_z \Psi|_{z=z_0^+} \quad \text{and} \quad p^+ = -\,\rho_a \partial_t \Psi|_{z=z_0^+}\,, \tag{5.42a}$$

$$\partial_t w = -\,\partial_z \Psi|_{z=z_0^-} \quad \text{and} \quad p^- = +\,\rho_a \partial_t \Psi|_{z=z_0^-}\,. \tag{5.42b}$$

Their physical interpretation is straightforward—the first of each pair requires the transverse velocity of the 2D structure to be equal to the velocity of the acoustic field, whereas the second defines the pressure acting on the structure in terms of Ψ, with different signs depending on the direction (p^+ acts from above, p^- from below).

Finite 2D domains and boundary conditions

When the membrane is defined over a finite domain $\mathcal{D}$, the previous argument can be repeated, with coupling conditions appearing only over $\mathcal{D}$. Outside this region, in fact, the integration of Ψ along the z direction finds no barrier. For a fixed membrane, no additional boundary conditions need be applied over $\partial\mathcal{D}$, as no pressure is exerted by the acoustic field there.

5.4.2 Finite difference scheme

The numerical implementation of air coupling follows the reasoning of the continuous case. Here, however, the spatial steps for the 2D and 3D grids, approximating the two regions of interest, are in general different, due to the different stability conditions in place. This problem has been acknowledged before by several authors [117, 2, 52]. The use of the same step for both grids, in fact, is unwise, as it would cause bandlimiting effects on one of the two grids, as discussed on page 46 for the plate equation. Rhaouti *et al.* [146, 52] exploited the particular values of the wave speeds in air and in the membrane in order to construct the membrane mesh as a refinement of the air's, while minimising at the same time the dispersion artefacts. It is clear that such specialised approach could potentially introduce severe bandlimiting effects, if general values of the wave speeds be chosen. One possible alternative solution is the use of interpolation matrices between the two grids. This approach, which has the additional advantage of preserving the energy conservation of the numerical scheme in an elegant way, has been proposed by Bilbao in [20], and will be adopted here, as well.

Returning now to the case of an infinite membrane, let $w^n_{l_x,l_y}$ be the discrete approximation of $w(x,y,t)$, with $(l_x,l_y) \in \mathbb{Z}^2$, over a grid with spatial step h_{sm}. It is convenient to place this 2D grid at vertical coordinate z_0, half way between neighbouring horizontal slices of the 3D field. This can always be done for the single membrane in isolation. The finite difference scheme for (5.37) can be written as

$$\rho H \delta_{tt} w = \mathfrak{O}_m w + p^+ + p^-, \tag{5.43}$$

where $\mathfrak{O}_m$ is the finite difference counterpart of the operator $\mathcal{O}_m$. An energy analysis can be performed according to the particular choice for $\mathfrak{O}_m$, but leads invariably to the energy balance

$$\delta_{t+}\mathfrak{h}_{\text{sm}} = \langle \delta_{t\cdot} w,\, p^+ + p^- \rangle_{\mathbb{Z}^2}, \tag{5.44}$$

where $\mathfrak{h}_{\text{sm}}$ represents the energy for the stiff membrane.

The finite difference scheme (5.27) for the acoustic field remains unaltered everywhere, except for the two horizontal slices above and below the 2D structure, see Figure 5.5. Let l_z^+ and l_z^- be the indices for these slices, respectively. Here, a direct update is prevented by the presence of the membrane, which acts as a partial barrier—the Laplacian stencil requires the value of a ghost point on the other side, which is to be determined via coupling conditions. The energy analysis reveals the presence of additional terms produced by the summation by parts, such that

$$\delta_{t+}\mathfrak{h}_\mathrm{a} = -\rho_a \langle \delta_{t\cdot}\Psi, \delta_{z-}\Psi \rangle_{\mathbb{Z}^2\times\{l_z^+\}} + \rho_a \langle \delta_{t\cdot}\Psi, \delta_{z+}\Psi \rangle_{\mathbb{Z}^2\times\{l_z^-\}}, \tag{5.45}$$

where the two inner products on the right hand side are calculated over the 2D slices of the acoustic field grid at l_z^+ and l_z^-, respectively. Note that the two inner products appearing in (5.44) and (5.45) have slightly different characters. Although being both 2D inner products, they are calculated over the grids for w and Ψ, respectively. Thus, the grid spacings that must be used according to the definition (2.108) are h_sm in the first case, and h_a in the second. As for the continuous case, when one considers the system as a whole, the total energy must be conserved, and the terms on the right hand side of (5.44) and (5.45) must sum to zero, such that

$$\langle \delta_{t\cdot}w,\, p^+ \rangle_{\mathbb{Z}^2} - \rho_a \langle \delta_{t\cdot}\Psi, \delta_{z-}\Psi \rangle_{\mathbb{Z}^2\times\{l_z^+\}} = 0, \tag{5.46a}$$

$$\langle \delta_{t\cdot}w,\, p^- \rangle_{\mathbb{Z}^2} + \rho_a \langle \delta_{t\cdot}\Psi, \delta_{z+}\Psi \rangle_{\mathbb{Z}^2\times\{l_z^-\}} = 0. \tag{5.46b}$$

Here, a problem arises, as anticipated at the beginning of this section. As the grids for w and Ψ do not align due to the different spatial step, interpolation must be used. Let $\mathcal{I}^{\mathrm{3D}\to\mathrm{2D}}$ be the interpolant from (a 2D slice of) the 3D grid to the 2D grid, and $\mathcal{I}^{\mathrm{2D}\to\mathrm{3D}}$ be the interpolant from the 2D grid to (a slice of) the 3D grid. Denoting with Ψ^+ and Ψ^- the slices of the acoustic field directly above and below the 2D structure, respectively, one could write the following coupling conditions

$$\mathcal{I}^{\mathrm{2D}\to\mathrm{3D}}\delta_{t\cdot}w = -\,\delta_{z-}\Psi|_{l_z^+} \quad \text{and} \quad p^+ = -\rho_a \mathcal{I}^{\mathrm{3D}\to\mathrm{2D}}\,\delta_{t\cdot}\Psi|_{l_z^+} \tag{5.47a}$$

$$\mathcal{I}^{\mathrm{2D}\to\mathrm{3D}}\delta_{t\cdot}w = -\,\delta_{z+}\Psi|_{l_z^-} \quad \text{and} \quad p^- = +\rho_a \mathcal{I}^{\mathrm{3D}\to\mathrm{2D}}\,\delta_{t\cdot}\Psi|_{l_z^-}\,, \tag{5.47b}$$

which are analogous to those derived in the continuous case. As these two sets of conditions are independent, different interpolants could in principle be used for the upper and lower slice of the acoustics field. However, in order for the scheme to be stable, a condition must be imposed on these operators, which can be more easily derived when Eqs. (5.47) are cast into vector form.

Take, e.g., the first of (5.46), and let $\mathbf{w}$, $\boldsymbol{\Psi}^+$ and $\mathbf{p}^+$ be the vectorised representations of w, $\Psi|_{l_z^+}$ and p^+, respectively. The coupling condition now becomes

$$h_\mathrm{sm}^2 \delta_{t\cdot}\mathbf{w}^T\mathbf{p}^+ - h_a^2\rho_a\delta_{t\cdot}(\boldsymbol{\Psi}^+)^T\delta_{z-}\boldsymbol{\Psi}^+ = 0. \tag{5.48}$$

Substituting the vector expression for p^+ given in (5.47a) into the previous equation yields

$$h_\mathrm{sm}^2 \delta_{t\cdot}\mathbf{w}^T\left(-\rho_a\boldsymbol{\mathcal{I}}^{\mathrm{3D}\to\mathrm{2D}}\delta_{t\cdot}\boldsymbol{\Psi}^+\right) - h_a^2\rho_a\delta_{t\cdot}(\boldsymbol{\Psi}^+)^T\delta_{z-}\boldsymbol{\Psi}^+ = 0, \tag{5.49}$$

where $\boldsymbol{\mathcal{I}}^{\mathrm{3D}\to\mathrm{2D}}$ is the matrix representation of the interpolant $\mathcal{I}^{\mathrm{3D}\to\mathrm{2D}}$. Gathering together the identical terms leads to

$$-\rho_a \delta_{t\cdot}(\mathbf{\Psi}^+)^T \left(h_{\mathrm{sm}}^2 (\boldsymbol{\mathcal{I}}^{\mathrm{3D}\to\mathrm{2D}})^T \delta_{t\cdot}\mathbf{w} + h_a^2 \delta_{z-}\mathbf{\Psi}^+ \right) = 0, \tag{5.50}$$

where the superscript $(\,\cdot\,)^T$ denotes matrix transposition. In order for this equation to hold, the expression in parentheses must vanish. Given its similarity with the first of (5.47a), one must have

$$\boldsymbol{\mathcal{I}}^{\mathrm{2D}\to\mathrm{3D}} = \frac{h_{\mathrm{sm}}^2}{h_a^2} (\boldsymbol{\mathcal{I}}^{\mathrm{3D}\to\mathrm{2D}})^T, \tag{5.51}$$

for both equalities to hold at the same time. This condition on the interpolants guarantees the energy conservation of the scheme, regardless of the particular choice of interpolation adopted. The same condition may be arrived at for the lower slice of the acoustic field using a similar reasoning.

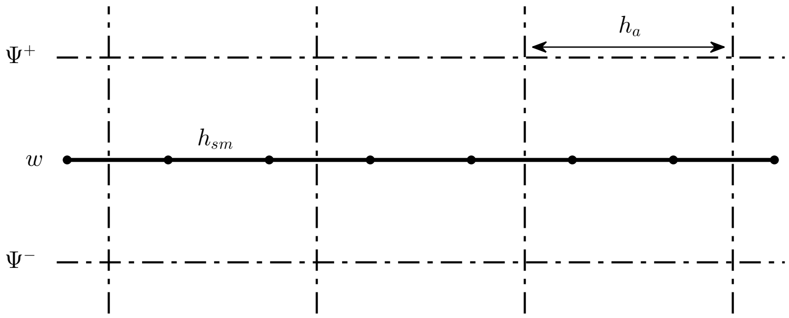


Figure 5.5: Vertical cross section of the acoustic field grid (dashed-dotted lines) and of the stiff membrane grid (solid line). Horizontal slices of the acoustic field above and below the membrane are marked as Ψ^+ and Ψ^-, respectively. The discrete steps h_a and h_{sm} for the acoustic field and membrane are also indicated.

Interpolants

Given a grid function w_{l_x,l_y}, interpolation is the necessary tool to extract the value of w at an output position (x_o, y_o) not belonging to the grid. The simplest (and crudest) way to achieve this is by nearest neighbour interpolation, where the value at (x_o, y_o) is the same as that of the nearest grid point.

Better results can be achieved through bilinear interpolation. Consider Figure 5.6, where the value at position (x_o, y_o) depends on the four neighbouring points A, B, C, and D. Let the integers representing the grid point A be l_{x_A} and l_{y_A}. Then, define the remainders $\alpha_x = x_o/h - l_{x_A}$ and $\alpha_y = y_o/h - l_{y_A}$, normalised measures of the distance between the output location and the grid point ($0 \le \alpha_x < 1$, $0 \le \alpha_y < 1$). Finally, the interpolated value for the function w at (x_o, y_o) can be obtained as a weighted average of the four nearest neighbouring values as

$$\mathcal{I}(x_o, y_o)w = (1-\alpha_x)(1-\alpha_y)w_A + \alpha_x(1-\alpha_y)w_B + (1-\alpha_x)\alpha_y w_C + \alpha_x\alpha_y w_D. \tag{5.52}$$

Interpolation is also necessary when two grids with different spacings are required to communicate between one another, as in the case of air coupling. In this case, interpolation must be performed for every point of the output grid in terms of the input grid. When the output grid function is stored as a vector of size N_o, the interpolant can be expressed as an $N_o \times N_i$ rectangular matrix, where N_i is the size of the input grid function. Given the bilinearity of the operation, interpolants can be constructed by means of Kronecker tensor product between the x- and y-axis interpolants. Typical sparsity patterns for the interpolation matrices are shown in Figure 5.7.

Higher order interpolants are of course possible, like bicubic or sinc interpolation, provided that they satisfy the condition (5.51). These, however, require a larger stencil of points and come at higher computational cost. Thus, it is easier in many applications to adopt bilinear interpolation, in order to keep the computational complexity low.

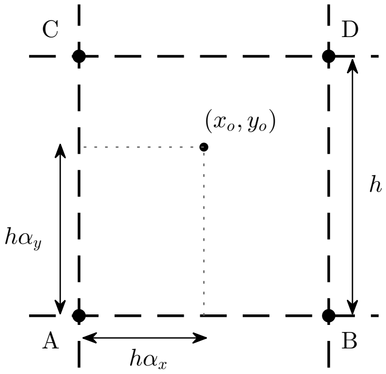
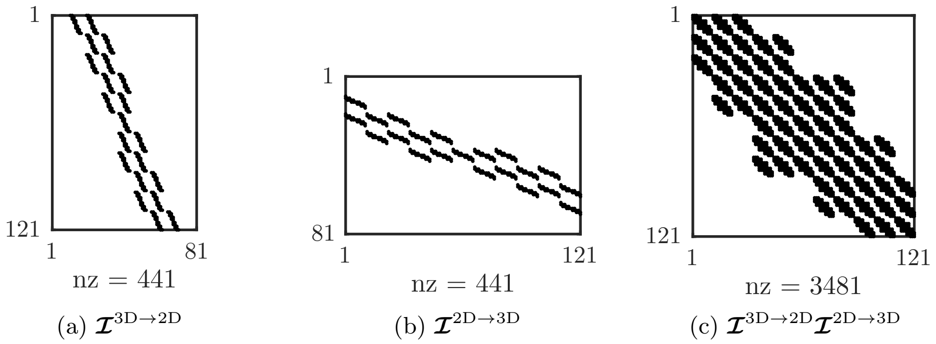


Figure 5.6: Bilinear interpolation at a point (x_o, y_o) over a grid with spacing h. The values at the nearest neighbouring points A, B, C, and D are weighted through the remainders α_x and α_y.



(a) $\mathcal{I}^{3D\rightarrow 2D}$ (b) $\mathcal{I}^{2D\rightarrow 3D}$ (c) $\mathcal{I}^{3D\rightarrow 2D}\mathcal{I}^{2D\rightarrow 3D}$

Figure 5.7: Sparsity patterns for the interpolation matrices introduced in the previous section, together with the number of non-zero entries (nz) in the matrix. Interpolation is performed between the 2D grid for a square stiff membrane (which has $N_x = N_y = 11$ points) and a slice of the 3D acoustic field grid (which has 81 points). Apart from a scaling factor, $\mathcal{I}^{3D\rightarrow 2D}$ and $\mathcal{I}^{2D\rightarrow 3D}$ are transpose matrices. Note that, as some points of the acoustic field lie outside of the region determined by the membrane, they do not intervene in interpolation. The product of the two interpolants is also presented.

5.4.3 Numerical implementation

The numerical implementation of the air coupling scheme described above depends on the nature of the system, and in particular, whether it is linear or nonlinear. The air coupling with a linear membrane has been discussed by Bilbao in [20], while the coupling with a membrane with a Berger nonlinearity has been treated in [30]. The coupling with a von Kármán nonlinearity and with a distributed collision model will be presented here, and can be considered one of the original contributions of this work.

Consider first the case of a stiff membrane in the linear regime. Equation (5.43) becomes

$$\rho H \delta_{tt} w = \mathfrak{O}_{\mathrm{m}} w + p^+ + p^-, \tag{5.53}$$

where $\mathfrak{O}_{\mathrm{m}}$ is the operator defined in (2.165) acting on w^n only, such that the update of the scheme without air would be explicit, and let $\widetilde{w}$ be the set of quantities known from the previous time step (see (2.167)). When air coupling is present, the update becomes

$$\begin{aligned}
w^{n+1} &\overset{(5.43)}{=} \widetilde{w} + \frac{1}{\rho H(1+\sigma_0 k)}\left(p^- + p^+\right) && (5.54\text{a})\\
&\overset{(5.47)}{=} \widetilde{w} + \frac{\rho_a}{\rho H(1+\sigma_0 k)}\mathcal{I}^{3\mathrm{D}\to 2\mathrm{D}}\left(\delta_{t\cdot}\Psi^- - \delta_{t\cdot}\Psi^+\right) && (5.54\text{b})\\
&\overset{(2.107\text{b})}{=} \widetilde{w} + \frac{\rho_a}{\rho H(1+\sigma_0 k)}\mathcal{I}^{3\mathrm{D}\to 2\mathrm{D}}\left[\delta_{t-}\Psi^- - \delta_{t-}\Psi^+ + \frac{k}{2}\left(\delta_{tt}\Psi^- - \delta_{tt}\Psi^+\right)\right] && (5.54\text{c})\\
&\overset{(5.27)}{=} \widetilde{w} + \frac{\rho_a}{\rho H(1+\sigma_0 k)}\mathcal{I}^{3\mathrm{D}\to 2\mathrm{D}}\left[\delta_{t-}\Psi^- - \delta_{t-}\Psi^+ + \frac{c^2 k}{2}\left(\delta_{\Delta}^{(3)}\Psi^- - \delta_{\Delta}^{(3)}\Psi^+\right)\right] && (5.54\text{d})\\
&= \widetilde{w} + \frac{\rho_a}{\rho H(1+\sigma_0 k)}\mathcal{I}^{3\mathrm{D}\to 2\mathrm{D}}\Big[\delta_{t-}\Psi^- - \delta_{t-}\Psi^+ + \\
&\qquad \frac{c^2 k}{2}\left(\delta_{\Delta}^{(2)}\Psi^- - \delta_{\Delta}^{(2)}\Psi^+\right) + \frac{c^2 k}{2h_a}\left(\delta_{z-} - \delta_{z+}\right)\Psi^- - \frac{c^2 k}{2h_a}\left(\delta_{z-} - \delta_{z+}\right)\Psi^+\Big] && (5.54\text{e})\\
&\overset{(5.47)}{=} \widetilde{w} + \frac{\rho_a c^2 k}{2\rho H h_a(1+\sigma_0 k)}\mathcal{I}^{3\mathrm{D}\to 2\mathrm{D}}\left[-2\mathcal{I}^{2\mathrm{D}\to 3\mathrm{D}}\delta_{t\cdot} w\right] + \frac{\rho_a}{\rho H(1+\sigma_0 k)}\mathcal{I}^{3\mathrm{D}\to 2\mathrm{D}}\mathfrak{K} && (5.54\text{f})
\end{aligned}$$

where $\mathfrak{K}$ represents quantities known from previous time steps, and it is defined as

$$\mathfrak{K} = \frac{c^2 k}{2}\left(\delta_{\Delta}^{(2)}\Psi^- - \delta_{\Delta}^{(2)}\Psi^+\right) + \delta_{t-}\Psi^- - \delta_{t-}\Psi^+ + \frac{c^2 k}{2h_a}\delta_{z-}\Psi^- - \frac{c^2 k}{2h_a}\delta_{z+}\Psi^+, \tag{5.55}$$

and $\delta_{\Delta}^{(2)}$ is the 2D discrete Laplacian defined in (2.105). Defining the composition of the interpolants as

$$\mathcal{E} = \mathcal{I}^{3\mathrm{D}\to 2\mathrm{D}}\mathcal{I}^{2\mathrm{D}\to 3\mathrm{D}}, \tag{5.56}$$

introducing the constant η_c which will be defined shortly and bringing the unknown terms on the left hand side, one finally obtains

$$\left(1 + \eta_c\mathcal{E}\right) w^{n+1} = \widetilde{w} - \eta_c\mathcal{E}\, w^{n-1} + \frac{\rho_a}{\rho H(1+\sigma_0 k)}\mathcal{I}^{3\mathrm{D}\to 2\mathrm{D}}\mathfrak{K}, \tag{5.57}$$

which can now be used to calculate w^{n+1}. When written in vector-matrix form, this scheme takes the form

$$\mathbf{A}\mathbf{w}^{n+1} = \mathbf{B}\mathbf{w}^{n} + \mathbf{C}\mathbf{w}^{n-1} + \mathbf{k}^{n}, \tag{5.58}$$

where $\mathbf{k}^n$ is a known vector that depends on Ψ. All the matrices appearing here are constant, with

$$\mathbf{A} = \mathbb{1} + \eta_c \mathbf{E}, \qquad \eta_c = \frac{\rho_a c_a^2}{2\rho H h_a (1 + \sigma_0 k)}, \tag{5.59}$$

and with $\mathbf{E}$ the matrix representation of the operator $\mathcal{E}$. Here, $\mathbf{A}$ is positive definite, given the relation (5.51) between the interpolants and η_c is a coupling constant that depends on the parameters of the system. This scheme is thus implicit, as it requires the solution of a linear system. Once $\mathbf{w}^{n+1}$ is known, the values of the acoustic field at positions l_z^+ and l_z^- can be updated by relating the ghost points of $\boldsymbol{\Psi}$ to $\mathbf{w}^{n+1}$.

Algorithm structure

It is worth now pausing the discussion in order to analyse the details of the numerical implementation outlined above. The update for the scheme can be schematically performed as follows:

1. Update the acoustic field. The update for the points in contact with the membrane is partial, and does not take into account the ghost points on the other side of the membrane. The update for all the other points, instead, will not be modified by the following operations.
2. Calculate $\mathfrak{K}$ according to (5.55).
3. Update the membrane using (5.58). At this stage, a linear system solution is required (see below).
4. Calculate the missing derivatives of the acoustic field through

$$\boldsymbol{\mathcal{I}}^{\mathrm{2D}\to\mathrm{3D}}\delta_{t\cdot}\mathbf{w} \;= -\delta_{z-}\boldsymbol{\Psi}^{+}, \qquad \boldsymbol{\mathcal{I}}^{\mathrm{2D}\to\mathrm{3D}}\delta_{t\cdot}\mathbf{w} \;= -\delta_{z+}\boldsymbol{\Psi}^{-}, \tag{5.60}$$

   which are the matrix-vector form of (5.47a) and (5.47b).
5. Update the acoustic field over the 2D slices in contact with the membrane.

5.4.4 Iterative methods

The finite difference scheme presented above depends on the solution of the linear system (5.58), which has the form

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{5.61}$$

where $\mathbf{x}$ is the unknown $\mathbf{w}^{n+1}$ and $\mathbf{b}$ is the known vector. In a typical simulation, this operation can represent a considerable portion of the computation time. Furthermore, in a parallel implementation, this part of the code can become the bottleneck, as this is a serial operation if standard iterative methods are used. Using a direct linear system solution, like Gaussian elimination, on an $N \times N$ system requires on the order of $N^3/3$ operations [167], and this task becomes rapidly very computationally expensive. Storing the inverse is also not a possibility, given the large number of grid points generally involved, and the fact that such inverse would be a dense matrix.

One possible way to overcome this issue is the use of iterative methods [167, 157]. Additional information and few important results are collected in Appendix C. For the linear system at hand, one simple possibility is the Gauss-Seidel method. This requires the matrix $\mathbf{A}$ to be split into two matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ such that

$$\mathbf{A} = \mathbf{A}_1 - \mathbf{A}_2. \tag{5.62}$$

Starting from a guess solution $\mathbf{x}_0$ to (5.61), the original system can then be transformed into the following recursion:

$$\mathbf{A}_1 \mathbf{x}_{i+1} = \mathbf{A}_2 \mathbf{x}_i + \mathbf{b}. \tag{5.63}$$

It is possible to show that this method converges to the exact solution when every eigenvalue λ of the matrix $\mathbf{A}_1^{-1}\mathbf{A}_2$ satisfies $|\lambda| < 1$ [167] (see also Appendix C).

In the present case, the natural choice for such splitting is probably the one derived from the definition of $\mathbf{A}$ (5.59), that is

$$\mathbf{A}_1 = \mathbb{1}, \qquad \mathbf{A}_2 = -\eta_c \mathbf{E}. \tag{5.64}$$

The key point is that the matrix $\mathbf{E}$ representing the product of the interpolants in (5.56), has maximum eigenvalues near unity, which can be easily verified with a numerical check. Thus, one obtains with a very good approximation that the spectral radius r of the matrix $\mathbf{A}_1^{-1}\mathbf{A}_2$ is

$$r(\mathbf{A}_1^{-1}\mathbf{A}_2) = \eta_c, \tag{5.65}$$

and this parameter governs the rate of convergence of the algorithm. For typical simulations, the coupling coefficient η_c is quite small. Substituting the stability condition (5.33) into (5.59), in fact, one obtains

$$\eta_c = \frac{\rho_a c_a}{2\sqrt{3}} \cdot \frac{k}{\rho H}. \tag{5.66}$$

Considering fixed the parameters for the air, η_c depends only on the sampling rate and on the thickness and density. For a 1 mm thick metal plate at sampling rate of 44.1 kHz, the value of the coupling coefficient is $\approx 3.4 \times 10^{-4}$. It is found that this value leads to convergence of the algorithm in 5-6 iterations.

5.4.5 Energy conservation

Consider a square metal plate with $L_x = L_y = 0.3$ m and thickness 1 mm coupled with the acoustic field under lossless conditions and with reflective boundaries on the 3D box. Figure 5.8 shows the results in terms of energy for a simulation performed at sampling rate of 44.1 kHz.

The plate is initialised with a raised cosine deformation of amplitude 3 mm and width 3 cm. It is possible to see the conservation of the total energy, while exchanges of energy take place between the plate and the air. A dynamic equilibrium is soon attained between the two quantities, which oscillate around constant medium values. This seems to suggest that there is no favoured direction for the energy to flow. Energy variations are on the order of machine precision.

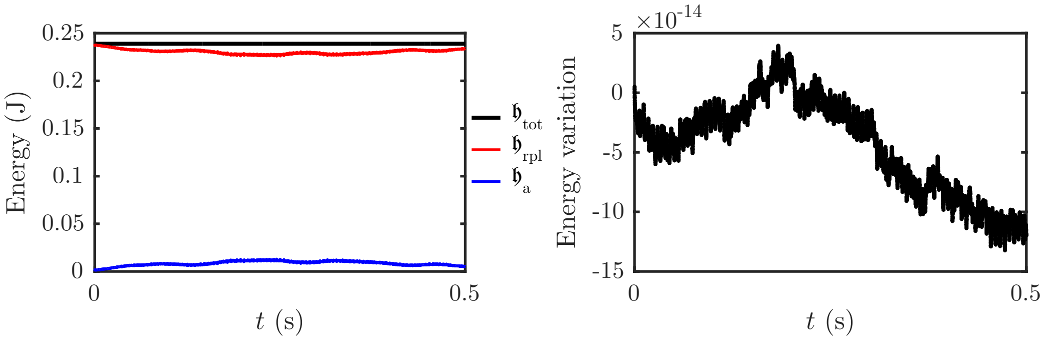


Figure 5.8: Energy conservation for the linear plate coupled with the acoustic field under lossless conditions and with reflective boundaries over the 3D box. Left: Contributions to the total energy from the plate and the air. Their sum, i.e., the total energy of the system, is conserved. Right: Normalised energy variations for the system. The sampling rate is 44.1 kHz.

5.4.6 Air loading effects

When a 2D plate or membrane is coupled with the acoustic field, the frequencies of the modes for the object in isolation are shifted downwards, as the presence of air effectively acts as an added mass [84]. This phenomenon is generally referred to as air loading, and is important in determining the modal frequencies of drums [60].

Consider, for example, a simply supported steel plate, with $L_x = 0.4$ m and $L_y = 0.3$ m, thickness $H = 0.5$ mm and $\kappa = 0.77$, coupled with the acoustic field. The frequencies for the plate in isolation are known theoretically, and are given in (2.85). Figure 5.9 shows the spectrum of the output taken over the plate coupled with the air up to 200 Hz. It is possible to see a downward shift of all the modes with respect to the theoretical values, as expected. The values of these peaks, obtained by zero-padding the signal and interpolating the spectrum, are reported in Table 5.1, together with the relative cent deviations from the theoretical values. A similar study has been performed by the Author in a previous work [181], with analogous results.

Without air	**With air**	**Cent dev.**
44.52 Hz	43.73 Hz	-30.97
86.28 Hz	85.17 Hz	-22.38
136.33 Hz	134.62 Hz	-21.80
155.87 Hz	154.11 Hz	-19.67
178.08 Hz	176.18 Hz	-18.59

Table 5.1: Frequencies for the first five modes of the simply supported plate with parameters as in the caption of Figure 5.9. The theoretical values for the isolated plate are reported together with the results of a simulation with air coupling included and the relative cent deviations.

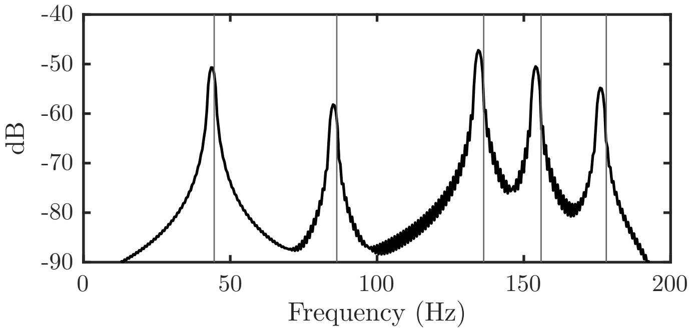


Figure 5.9: Spectrum of the output for a simply supported steel plate, with $L_x = 0.4$ m and $L_y = 0.3$ m, thickness $H = 0.5$ mm and $\kappa = 0.77$ coupled with the acoustic field. The peaks are shifted downwards with respect to the theoretical values for a plate in isolation (marked with vertical lines), indicating the presence of air loading effects. The simulation sampling rate is 65536 Hz.

5.4.7 Air coupling with Berger nonlinearity

The coupling between the air and a lossy plate with a Berger type nonlinearity will be discussed in this section. The continuous equation for this system can be written as

$$\rho H \partial_t^2 w = -D\Delta^2 w - 2\sigma_0 \partial_t w + \mathfrak{T}\Delta w + p^+ + p^-, \tag{5.67}$$

where the first part represents the equation for an isolated plate with Berger nonlinearity, as in (3.1), while the last two terms represent the pressure of the acoustic field acting above and below the plate, respectively. A finite difference implementation for this scheme will combine that of the simple plate with the one described in the previous section. This approach has been adopted by Bilbao and Webb in [30], where a nonlinear membrane of a timpani drum embedded in a 3D space has been modelled. The finite difference scheme for the present case can be written as

$$\rho H \delta_{tt} w^n = -D\delta_{\Delta,\Delta} w^n - 2\sigma_0 \delta_{t\cdot} w^n + \mathfrak{t}^n\, \delta_\Delta w^n + p^+ + p^-, \tag{5.68}$$

where all the quantities have already been defined in Section 3.1. It is easy to show that, ultimately, this scheme can be written in vector-matrix form as

$$\mathbf{A}^n \mathbf{w}^{n+1} = \mathbf{B}\mathbf{w}^n + \mathbf{C}^n \mathbf{w}^{n-1} + \mathbf{k}^n. \tag{5.69}$$

with $\mathbf{k}^n$ a vector that depends on known values of Ψ. Here, the matrix $\mathbf{A}^n$ is defined as

$$\mathbf{A}^n = (1+\sigma_0 k)\mathbb{1} + \mathbf{q}^n \mathbf{q}^{n,T} + \eta_c \boldsymbol{\mathcal{I}}^{3\text{D}\to 2\text{D}} \boldsymbol{\mathcal{I}}^{2\text{D}\to 3\text{D}} \tag{5.70}$$

with $\mathbf{q}^n$ as in (3.16) and depends on the time step n. Given the small value of η_c, iterative methods are again an attractive way of solving the linear system, as suggested in [30].

Energy conservation

The system presented has an associated energy which is conserved in the lossless case and with reflective boundaries. The total energy is simply the sum of the contributions from the plate, the Berger nonlinearity and the air:

$$\mathfrak{h}_{\mathrm{tot}} = \mathfrak{h}_{\mathrm{rpl}} + \mathfrak{u}_{\mathrm{Berger}} + \mathfrak{h}_{\mathrm{a}}, \tag{5.71}$$

where the notation has been introduced in Section 3.1.1. Figure 5.10 shows the energy conservation of the scheme for a simulation with identical parameters and initial conditions as Figure 5.8. Normalised energy variations for the scheme are on the order of machine precision.

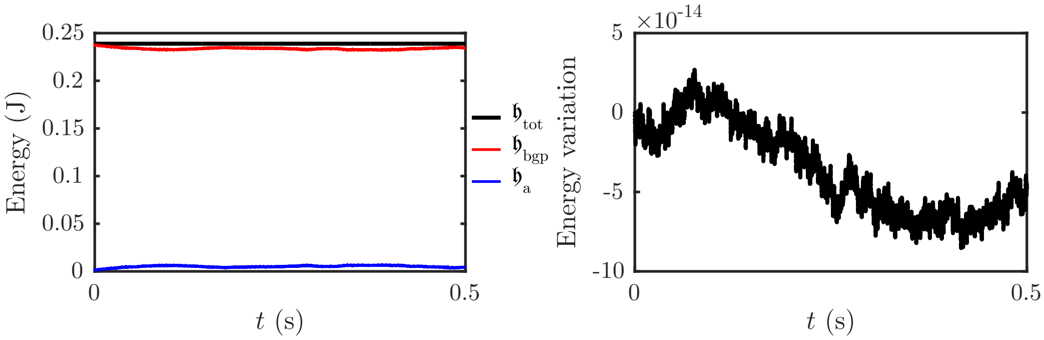


Figure 5.10: Energy for the Berger plate coupled with the air under lossless conditions and with reflective boundary conditions over the 3D box. Left: Contribution to the total energy from the plate ($\mathfrak{h}_{\mathrm{bgp}} = \mathfrak{h}_{\mathrm{rpl}} + \mathfrak{u}_{\mathrm{Berger}}$) and the air. Right: Normalised variations for the total energy. The sampling rate is 44.1 kHz.

5.4.8 Air coupling with von Kármán nonlinearity

Air coupling for a von Kármán nonlinear plate described by the system of equations (3.28) follows the same reasoning as the previous cases, and is an original contribution of this work. This constitutes the basis for the multiple plate virtual environment that will be presented in Section 6.2. Equation (3.28a) must be augmented with the air pressure terms, giving

$$\rho H \partial_t^2 w \quad = -D\Delta^2 w + \mathcal{L}(w, \Phi) + p^+ + p^-, \tag{5.72}$$

while (3.28b) remains unaltered.

A finite difference scheme is readily obtained as

$$\rho H \delta_{tt} w = -D\delta_{\Delta,\Delta} w + \mathfrak{l}(\mu_{t\cdot}\Phi, w) + p^+ + p^- \tag{5.73a}$$

$$\mu_{t+}\delta_{\Delta,\Delta}\Phi = -\frac{EH}{2}\mathfrak{l}(w, \mathrm{s}_{t+}w) \tag{5.73b}$$

The implementation requires this scheme to be cast into matrix-vector form, as it relies on the solution of a linear system, as discussed in Section 3.2.7. Such a system is similar to (3.62), and can be written as

$$\mathbf{A}_a^n \mathbf{x}^{n+1} = \mathbf{b}_a^n \tag{5.74}$$

The unknown $\mathbf{x}$ is still the concatenated vector of $\mathbf{w}$ and $\mathbf{\Phi}$, see (3.61). The matrix $\mathbf{A}$ and vector $\mathbf{b}$, however, must be modified in order to include air coupling effects as

$$\mathbf{A}_a^n = \left[\begin{array}{c|c} \mathbb{1} + \eta_c \mathbf{E} & -\boldsymbol{\Lambda}_{\boldsymbol{\Phi}}^n \\ \hline \boldsymbol{\Lambda}_{\mathbf{w}}^n & \mathbf{D}_{\Delta,\Delta} \end{array}\right], \quad \mathbf{b}_a^n = \left[\begin{array}{c} \mathbf{B}\mathbf{w}^n - \mathbf{C}\mathbf{w}^{n-1} + \boldsymbol{\Lambda}_{\boldsymbol{\Phi}}\boldsymbol{\Phi}^{n-1} + \mathbf{k}^n \\ -\mathbf{D}_{\Delta,\Delta}\boldsymbol{\Phi}^n \end{array}\right], \tag{5.75}$$

where $\mathbf{E}$ is the air coupling matrix and $\mathbf{k}^n$ is a vector of known terms which depend on the acoustic field Ψ. Notice now that the upper left part of the matrix $\mathbf{A}_a^n$ is not the identity matrix as in (3.63), thus another strategy must be adopted to solve the system. One possibility is to solve the system as it appears. Another possibility is the use of iterative methods, exploiting once again the fact that the coupling constant η_c is small [183].

Energy conservation

The energy conservation for the coupled von Kármán/air scheme is presented in Figure 5.11. The simulation is run with the same parameters as the two simulations presented before (linear and Berger plate), under lossless conditions and with reflective conditions over the boundaries of the 3D box. Normalised energy variations are on the order of machine precision.

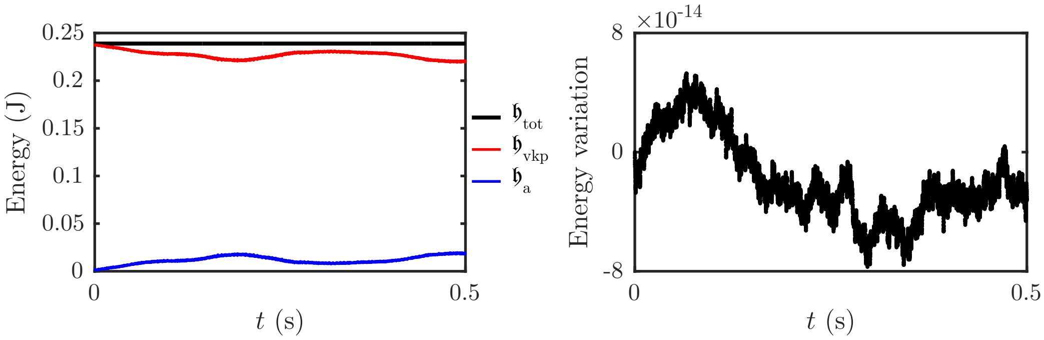


Figure 5.11: Energy of the von Kármán plate coupled with the air, under lossless conditions and with reflective boundaries over the 3D box. Left: Contribution to the total energy from the plate and the air. Right: Normalised variations for the total energy of the system. The sampling rate is 44.1 kHz.

5.4.9 Air coupling with a linear membrane and a mallet

The triple coupling between the air, a linear stiff membrane and a mallet striking from above will be discussed here. The equation for the membrane now reads

$$\rho H \partial_{tt} w = \mathcal{O}_{\mathrm{m}} w - g f_c + p^+ + p^-, \tag{5.76}$$

which must be coupled with the equation for the mallet, see (4.24), and with that for the acoustic field, Eq. (5.11). As before, two extra terms representing the action of the acoustic field are included in the membrane equation. The collision force f_c can be expressed in terms of a potential, see Eqs. (4.27) and (4.28). No direct interaction is considered between the mallet and the air, given the lumped nature of the object.

A finite difference discretisation for the membrane can be written as

$$\rho H \delta_{tt} w^n = \mathfrak{D}_{\mathrm{m}} w^n - g f_c^n + p^+ + p^- \tag{5.77}$$

where, again, all the symbols have been introduced in the previous chapter (see also the List of symbols on page viii). The numerical implementation resembles that for the membrane/air coupling in Section 5.4.3. Casting the various terms into matrix-vector form yields the following update

$$\mathbf{A}\mathbf{w}^{n+1} = \widetilde{\mathbf{w}} - \frac{k^2}{\rho H(1+\sigma_0 k)}\mathbf{g}f_c^n, \qquad \widetilde{\mathbf{w}} = \mathbf{B}\mathbf{w}^n + \mathbf{C}\mathbf{w}^{n-1} + \mathbf{k}^n \tag{5.78}$$

where all the terms are the same as Eq. (5.58) and are known from the previous time step, except for the last one. The vector $\mathbf{g}$ represents the discrete distribution of the impact point, while f_c^n is the collision force. Thus, apart from the presence of the matrix $\mathbf{A}$ defined in (5.70), the implementation can proceed as for the mallet and membrane in isolation. Using the notation of Section 4.2.1, the implementation requires the solution of the following nonlinear equation

$$\Gamma(r) = r + m\frac{\Pi(r+a) - \Pi(a)}{r} + b = 0, \tag{5.79}$$

in the unknown $r = \zeta^{n+1} - \zeta^{n-1}$, which is formally identical to (4.37) but for the definition of b and m, that now become

$$b = \zeta^{n-1} + \widetilde{z} - \frac{k^2}{\rho H(1+\sigma_0 k)}h_m^2\mathbf{g}^T\mathbf{A}^{-1}\widetilde{\mathbf{w}}, \quad m = \frac{k^2}{\rho H(1+\sigma_0 k)}h_m^2\mathbf{g}^T\mathbf{A}^{-1}\mathbf{g} + \frac{k^2}{M}. \tag{5.80}$$

Here, the variable $\zeta^n = h_m^2\mathbf{g}^T\mathbf{w}^n - z^n$ as defined in (4.33) represents the interpenetration of the mallet, and $\widetilde{z}$ groups together the known terms in the update for z^{n+1}. Notice now the presence of the inverse of the matrix $\mathbf{A}$ in the definitions. The scalar m is a constant, and can be easily computed before the loop and stored, while the quantity $\mathbf{A}^{-1}\widetilde{\mathbf{w}}$ can be computed using iterative methods (see Section 5.4.4).

Energy conservation

Consider a Mylar membrane under linear conditions with thickness 0.1 mm and wave speed 107 m/s, excited with a lumped mass of 25 g travelling with an initial speed of 5 m/s. The collision parameters chosen are $\alpha = 2.5$ and $K = 10^9$. The system is coupled with the acoustic field. In order to introduce the case of a drum head which will be presented in the next chapter, a circular domain is chosen for the membrane. Figure 5.12 shows the energy conservation results for this particular simulation at sampling rate of 44.1 kHz. The initial energy of the mallet is exchanged with the membrane and the air, with a peak at about 5 ms. The final energy of the mallet is lower than the initial one, as some is transferred to the membrane/air system. Energy variations are on the order of machine accuracy during the impact. A few discrete jumps can be seen when the mallet moves freely, and can be attributed to the loss of accuracy in the calculation of the mallet position—see Section 4.1.6 and Appendix B.4.

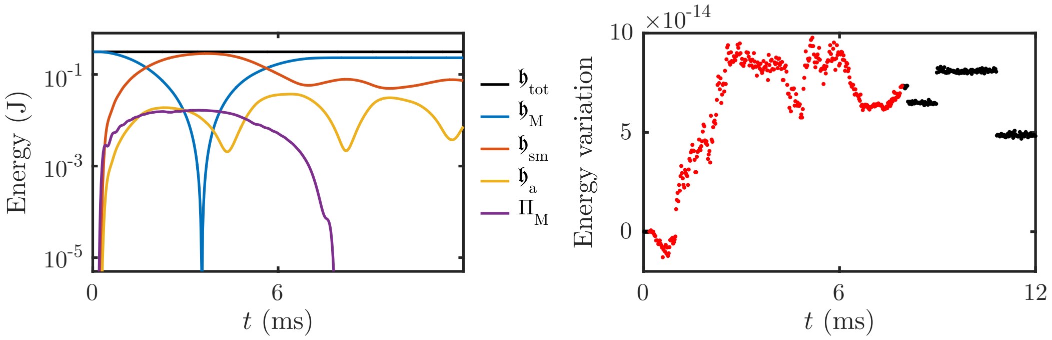


Figure 5.12: Energy conservation for the membrane/mallet system coupled with the 3D acoustic field. Left: Contributions to the total energy from the various components ($\mathfrak{h}_{\text{tot}} = \mathfrak{h}_{\text{M}} + \mathfrak{h}_{\text{sm}} + \mathfrak{h}_{\text{a}} + \Pi_{\text{M}}$). Right: Normalised variations of the total energy. Contact interaction is marked in red. The sampling rate is 44.1 kHz.

5.4.10 Distributed air coupling with a linear membrane and a string

The interaction between a string and a membrane has been discussed in Section 4.3.2. In this section, the coupling between the string/membrane system and the air will be presented. The equation for a membrane attached to a colliding string (see Eq. (4.72)), can be extended to include air coupling by adding the pressure terms:

$$\rho_m H \partial_t^2 w = \mathcal{O}_m w + \int_{\mathcal{D}_s} g(\chi, \mathbf{x}) \mathcal{F} \, \mathrm{d}\chi + g_0 f_0 + g_{L_s} f_{L_s} + p^+ + p^-, \tag{5.81}$$

while the equation for the string given in (4.48) remains unaltered. This must be accompanied by suitable boundary conditions, as explained in Section 4.3.6.

The numerical implementation of the system follows that of the non-coupled equations (see Sections 4.3.4 and 4.3.6), but the presence of air coupling requires (4.78) to be modified as

$$\mathbf{A}\mathbf{w}^{n+1} = \widetilde{\mathbf{w}} + \eta_m \mathbf{I}_{sm} \mathbf{f}^n + \eta_m \mathbf{g}_0 f_0^n + \eta_m \mathbf{g}_{L_s} f_{L_s}^n, \tag{5.82}$$

where all quantities have been defined before. Notice now the presence of the matrix $\mathbf{A}$ defined in (5.70). Inverting $\mathbf{A}$ in the previous equation leads to an expression that lends itself to be treated in a similar way to (4.78). First, write the pointwise forces f_0^n and $f_{L_s}^n$ in terms of $\mathbf{f}^n$, and then substitute them back into (5.82). At this point, following the procedure indicated in Section 4.3.6 leads to a nonlinear vector equation formally identical to $\mathbf{\Gamma}^{\dagger} = \mathbf{0}$ in (4.83). The difference here resides in the presence of $\mathbf{A}^{-1}$ in the calculation.

Simulation results

Consider a circular membrane with the same parameters as in Section 5.4.9, in contact with a metal string with tension $T_s = 3$ N. The string is fixed at the boundaries of the membrane and stretched along its main diameter. An initial raised cosine deformation on the membrane, with amplitude 1.2 mm and width 2 cm is adopted. Figure 5.13 shows the energy conservation of the scheme under lossless conditions and with reflective conditions over the boundaries of the 3D box at 44.1 kHz sampling rate. The initial energy of the membrane is gradually transferred to the air and, in minor portion, to the string and the collision potential (summed together here).

Normalised energy variations are on the order of machine accuracy. Larger variations can be seen at the beginning when the string is pushed downwards along its entire length.

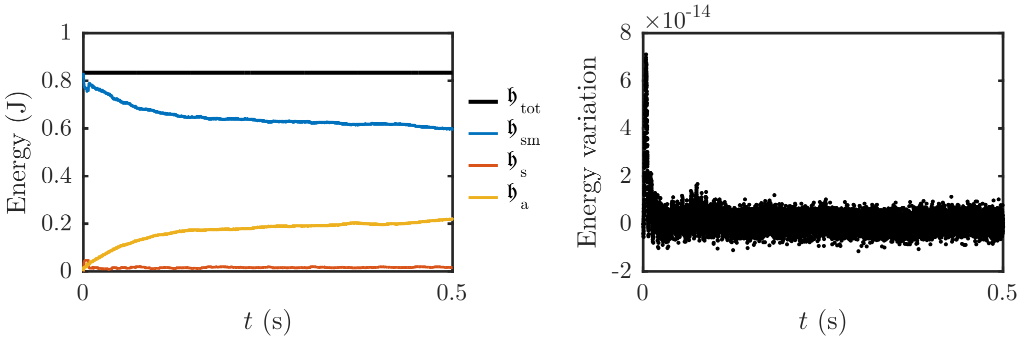


Figure 5.13: Energy conservation for the string/membrane system coupled with the 3D acoustic field, under lossless conditions and with reflective conditions over the boundaries of the 3D box. Left: Contribution to the total energy from the membrane, string and air ($\mathfrak{h}_{\mathrm{tot}} = \mathfrak{h}_{\mathrm{sm}} + \mathfrak{h}_{\mathrm{s}} + \mathfrak{h}_{\mathrm{a}}$). The string collision potential is summed to the string energy. Right: Normalised variations of the total energy. The simulation sampling rate is 44.1 kHz.

Chapter 6

Virtual instruments

The ultimate goal of physics-based sound synthesis is to build virtual instruments that musicians and composers can use to create music. These instruments need to be flexible enough to produce the large variety of sounds generally looked for by the musicians, while at the same time, as numerical algorithms, must be robust and stable over a wide range of parameters. When losses are not predominant, dissipation cannot be of great help in designing stable algorithms, and hence different approaches must be employed. The energy techniques presented in this work can thus be a valid candidate to ensure the stability of complex algorithms.

Frameworks for the construction of virtual instruments have been in existence for some time, starting from the well known CORDIS-ANIMA [42] and MOSAIC/Modalys [133, 76] systems to the more recent BlockCompiler [111]. As far as percussion instruments are concerned, digital waveguides have been successfully applied by van Duyne [193], Laird [117], Aird [2] and Gärder [87]. Modular sound synthesis environments have been proposed by Bilbao [18]. Modal methods have been applied to the sound synthesis of gongs by Ducceschi [74]. A hybrid approach based on finite elements and far-field acoustic transfer functions has been adopted by Chadwick *et al.* for the simulation of nonlinear thin shells [47]. A snare drum model has been recently proposed by Avanzini [6]. Finite difference methods have been adopted for the simulation of timpani drums in [146, 30] and snare drums [20], but also for xylophones [51, 72] and plates in the linear regime [53, 119].

The Author's contribution to this field is given by a novel modular sound synthesis environment composed of several plates vibrating under nonlinear conditions and coupled with the acoustic field [182], a novel model for the simulation of nonlinear double membrane drums [183] and a snare drum model [184]. These will be described in this chapter.

The instruments presented here are all based on the simple physical and numerical models developed in the previous chapters, which can be combined together in order to create complex systems. Since these technical aspects have already been discussed in detail, they are not repeated here. The emphasis is instead given to some musical and artistic aspects, with general remarks about the structure of a sound synthesis code and the control strategy for the various instruments.

6.1 Structure of a sound synthesis code

The sound synthesis environments that are presented in this chapter share a common structure. The core of each program is a numerical algorithm that solves the underlying equations for the system with a finite difference approach. The structure of this algorithm remains unchanged between different runs, and the user has no ability to modify it. This can be thought of as a "black box" that can be fed with input parameters and that produces sounds as output. Many similarities between the different sound synthesis environments can be found also at the level of numerical algorithm. Every code, in fact, can be divided into two parts, the precomputation phase and the main loop. The former is the setup stage of the code, whose main goal is to compute the finite difference operators necessary for the calculation and to initialise the algorithm, while the latter is the core of the code, where the equations describing the system are solved numerically and the sound is generated. More details regarding the various parts of the codes can be found in the next sections.

6.1.1 Input files

Each sound synthesis code requires the user to provide an instrument and a score file as inputs. The instrument file specifies the various parameters of the simulation. These include the sampling rate and all the physical quantities that describe the system, such as the size of the plates or membranes, their thickness, and their density. The read-out positions for the sound outputs are also specified here.

The score file specifies how the instrument is to be played. The simulation duration is indicated, and each strike is defined by the time of impact, the duration, the maximum force and the coordinates of the impact point on the 2D structure. Additional comments on scores and controls will be given in Section 6.5.

6.1.2 Precomputation stage

The precomputation stage can be further divided into several parts, which are represented in Figure 6.1.

System parameters generation

The initial part of the code is devoted to the generation of the parameters that describe the system. The physical parameters provided by the user are first converted into derived parameters (e.g., for a membrane, the tension, the density and the thickness are combined into the wave speed). These are then used to determine the grid spacings based on the stability conditions of the schemes and the number of grid points.

Geometry

The following step consists in defining the geometry of the system. First, the grids in Cartesian coordinates for the various components are created. These can be used to determine the location of the various grid points, and as a reference between different grids. Interpolation matrices are built according to the relative geometries of the grids, and different lists of points that satisfy certain conditions are also created at this stage. The latter can be used to identify which points of the acoustic field interact with the plate/membrane, or which grid points lie on the boundaries.

Finite difference matrix generation

The construction of finite difference matrices is at the heart of the precomputation stage. Here, the differential operators used in the simulation are approximated, taking into account boundary conditions. For example, the construction of the biharmonic operator in matrix form for the plate equation is given in (2.140). Additional matrices that combine together several operators are also built, in order to simplify the numerical implementation. These are generally indicated with **A**, **B** and **C** in the previous chapters.

Input/output

At this point, the program can compute the input/output vectors of the system. First, the raised cosine strikes that excite the instrument are created as external force vectors based on the parameters supplied by the user in the score file. The vectors representing the strike locations are also built. Then, vectors representing read-out positions are created, using interpolation, based on the parameters given in the instrument file. These are used to select from the entire state of the system the grid points from which the output waveform will be read and saved.

Initialisation

Finally, memory is allocated for the vectors representing the finite difference variables, and these are created according to the initial conditions of the problem specified by the user. At this point the precomputation stage ends, and the program enters the main loop.

6.1.3 Main loop

The main loop is where the PDEs describing the system are actually solved by updating the variables using the recursion determined by the finite difference schemes. Several different implementation strategies have been presented in the previous chapters which can be readily translated into numerical schemes.

In order to obtain the update, a loop over the time step is used. This encloses the entire calculation, which consists of several sequential operations involving matrices and vectors. The same operations are performed at every time step. At the end of each iteration, the variables of interest (like the displacement of a membrane, or the pressure of the acoustic field) are stored in a pre-allocated vector and the variables are reinitialised for the next time step.

6.1.4 Output sounds

At the end of the calculation, the user obtains the output files, computed at the specified positions. As the entire acoustic field is computed at every time step, the output sounds can be read simultaneously at several different locations, at no additional computational cost. Another possibility is to read the vibration of the plate/membrane directly over the 2D surface.

The outputs produced are arrays of numbers representing, e.g., the time history of the velocity of a membrane or the pressure of the acoustic field. These can be readily converted into sounds.

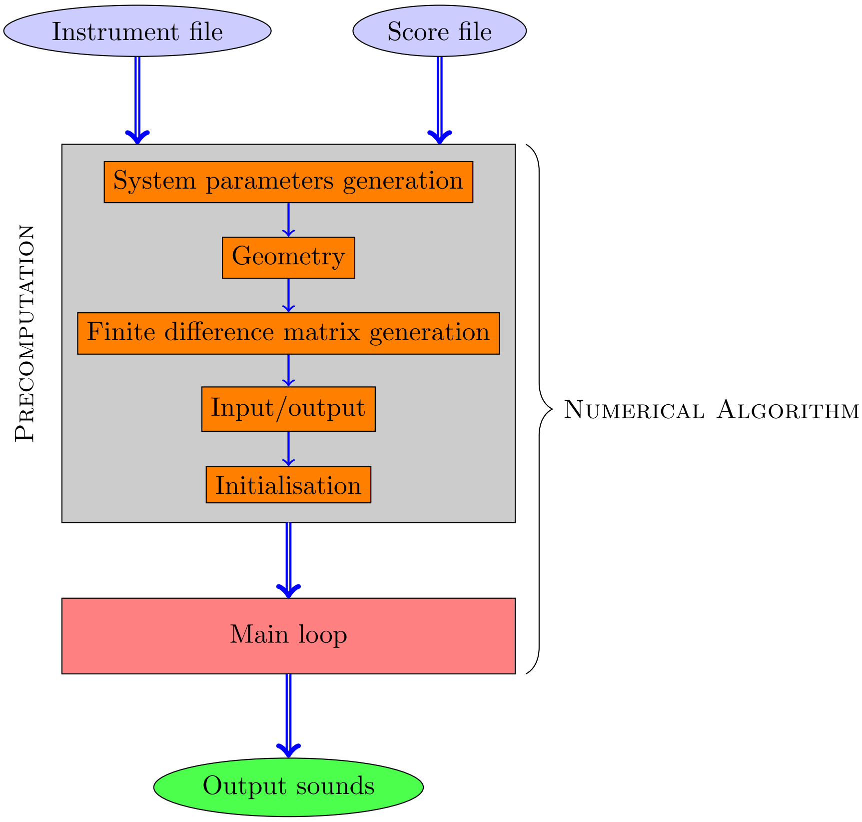


Figure 6.1: Diagram of a typical sound synthesis code. Two files created by the user, the instrument and score (in blue), are fed into the numerical algorithm (represented as a gray box). The various stages of the precomputation phase are marked with orange boxes, while the main loop is represented in red. At the end of the calculation, sound outputs are produced in the form of arrays of numbers (green ellipse).

6.2 Multiple plate environment in 3D

The multiple plate environment in 3D (MP3D) [182] is a virtual modular instrument composed of several rectangular plates vibrating under nonlinear conditions and embedded in a 3D box of air, with which they are coupled. The plates can be placed parallel to one another at any position within the box, and they can be thought of as "floating" in the air. Inputs are given by raised cosine strikes that can be injected at any position of the plates. The output sounds can either be taken directly from the velocity of the plates, or from the pressure of the acoustic field at many simultaneous locations within the box.

6.2.1 Description of the system

Consider a 3D box of parallelepipedal shape of dimensions $L_{\mathrm{box}} \times L_{\mathrm{box}} \times H_{\mathrm{box}}$ with Cartesian coordinates (x, y, z). This contains the main components of the MP3D, which are N_{pl} plates, each of which is defined over a rectangular region $\mathcal{P}_i$, with $i = 1, \ldots, N_{\mathrm{pl}}$, parallel to the xy plane. The position of the centre of each region is specified by a vector $\mathbf{x}_c^{(i)} = (x^{(i)}, y^{(i)}, z^{(i)})$. The physical variable of interest is the transverse displacement $w_i(x, y, t)$ of each plate. These are governed by the equations of motion for a von Kármán plate with losses and air coupling described in Section 5.4.8. The air is described by a 3D wave equation in the velocity potential

Ψ, see (5.11). At the walls of the air box, the absorbing boundary conditions described in Section 5.2.2 are employed. Figure 6.2 is a schematic representation of the geometry of a typical MP3D system. Three plates with different physical parameters are embedded in a finite box of air. Their position is defined by the location of their centres. Several different outputs can be specified within the box, at no additional computational cost.

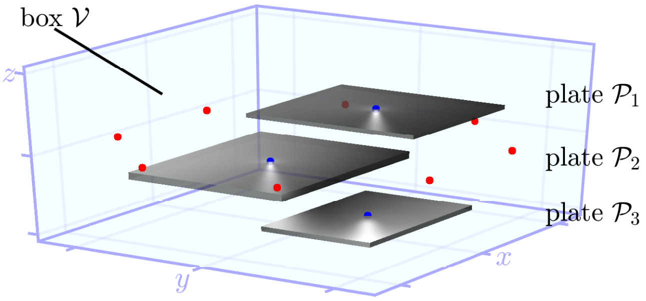


Figure 6.2: Geometry of a typical multiple plate environment in 3D. Three plates with different thickness are embedded in a box of air, with centres marked in blue. Several output locations (in red) can be taken simultaneously.

6.2.2 Numerical implementation

The numerical implementation for a single plate embedded in a box of air has been discussed in Section 5.4.8. The extension to N_{pl} plates in the same box can be obtained in a straightforward way by looping the algorithm over the different plates. One thing to notice is that different plates communicate with different slices of the 3D field. This issue is dealt with in the "geometry" phase of the precomputation stage of the code (see Section 6.1.2), where specific matrices are created in order to select the portion of the acoustic field that is in contact with each plate. One possible problem is that this approach holds as long as the plates are independent from one another. In other words, each plate needs to be coupled to different points of the acoustic field, such that no direct interaction occurs.

6.2.3 Output

In order to obtain sounds from this environment, the user must specify "virtual microphones" where the vibrations are to be recorded. In this code, there are two possibilities. The first is to choose some locations over the surface of the plate and to record the instantaneous velocity, as one would do by attaching a pick-up on a real world instrument. The second possibility is to save the pressure of the acoustic field at any point within the computational box. As the entire state of the system is computed at every time step, the output writing is not computationally expensive, and several "virtual microphones" can be used at the same time.

6.2.4 Computational cost

In a simulation, the main computational cost lies in the main loop. The precomputation stage, in fact, is performed only once and its cost is negligible with respect to the rest of the code.

The computational cost of the MP3D code and some possible ways to speed it up by means of parallel implementation has been the object of a recent work [142]. In this paper, the prototype code written in MATLAB by the Author has been ported to C and then CUDA by J. Perry, who also benchmarked the new versions against the original. Some of the results obtained in [142] are summarised here.

In the main loop of the MP3D code, the run time can be roughly divided into three parts: the air field update, the linear system solution for the plates and the remainder of the code. When the air box is small, the run time for these three parts is comparable, and none of them dominates the code. In order to obtain the maximum possible speed up, a hybrid approach involving CPUs and GPUs simultaneously was adopted. The air box, given the simple structure of the update, was a perfect candidate for parallelisation, as discussed in detail in the work of Webb [201]. The plate updates, instead, were calculated in parallel on different CPUs. The solution of the linear system (3.65b), in fact, being a serial operation, was not suited for parallel implementation. Finally, the remainder of the code, which consists mainly of matrices construction, could be sped up by adopting an alternative matrix format to MATLAB's `sparse`. The final optimised version of the code was between 60 and 80 times faster than the original version. Additional comments can be found in [142].

6.2.5 Energy

The energy analysis techniques presented in the previous chapters can be applied in order to ensure the stability of the numerical scheme. The worst case scenario, that of a system where no losses are present, must be checked first. This is to ascertain that no anomalous growth or leakage of energy be present in the code, which could potentially indicate the instability of the numerical simulation.

The total energy $\mathfrak{h}_{\mathrm{MP3D}}$ of the multiple plate system is given by the sum of the energy of the acoustic field and the energy of the various plates,

$$\mathfrak{h}_{\mathrm{MP3D}} = \mathfrak{h}_{\mathrm{a}} + \sum_{j=1}^{N_{pl}} \mathfrak{h}_{\mathrm{vkp}}^{(j)}, \tag{6.1}$$

where $\mathfrak{h}_{\mathrm{a}}$ has been defined in (5.30) and the energy for a single plate with von Kármán nonlinearity $\mathfrak{h}_{\mathrm{vkp}}$ has been given in (3.57).

Figure 6.3 shows the energy for a system of three plates, similar to that in Figure 6.2, under lossless conditions and with reflective conditions over the walls of the 3D box. The upper plate is excited with a raised cosine initial deformation, while the other two plates are set into motion through the wave propagation in air. The initial energy of the upper plate is transferred to the surrounding air and to the other plates, while the total energy of the system remains constant to machine precision.

6.2.6 Simulation results

Consider a system of three interacting plates within a box of air. Figure 6.4 shows the configuration of the system at successive time steps for a typical simulation. A strike on the upper plate generates a pressure wave that propagates through the box and reaches the other plates. The middle and lower plate are then excited and they start to vibrate, although the vibration

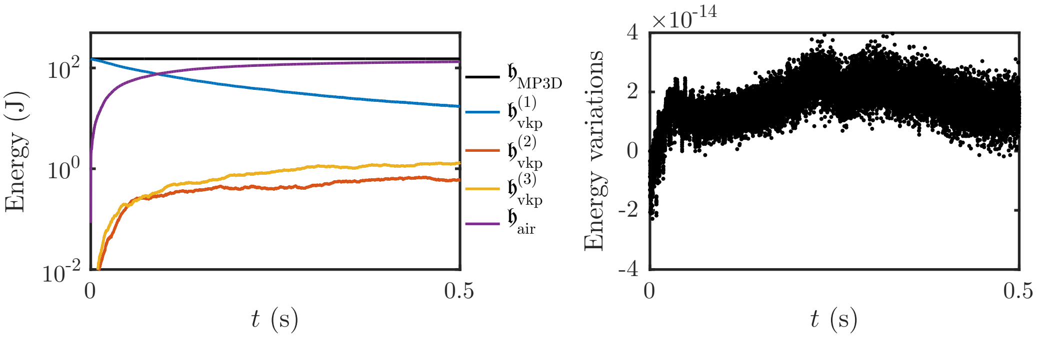


Figure 6.3: Energy for a typical simulation with the MP3D system. Left: Contributions to the total energy from the various components of the system in the lossless case and under reflective conditions over the boundaries of the 3D box. Right: Normalised variations of the total energy. The sampling rate of the simulation is 44.1 kHz.

amplitude is orders of magnitude smaller than the upper plate one. Absorbing conditions have been applied at the boundaries of the box.

6.3 Nonlinear double membrane drums

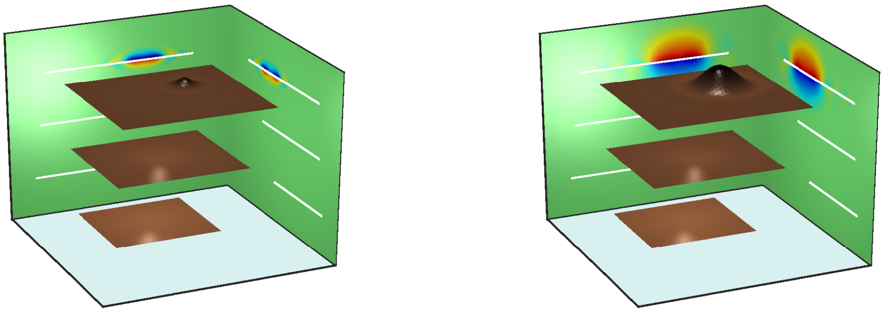
Double membrane drums like the tom-tom and the bass drum are an important component of Western music. Drums with two heads are also found in the Eastern tradition, like the Indian mridangam and the Japanese taiko. A physical model of such instruments can then be a very versatile tool in the hands of a composer.

Previous attempts to create virtual models of drums have already been mentioned in the introduction to this chapter, and include timpani drums [146, 117, 2, 30] and snare drums [20, 6]. To the knowledge of the Author, physical models of drums with nonlinearities have been presented only in [6, 29, 30] so far.

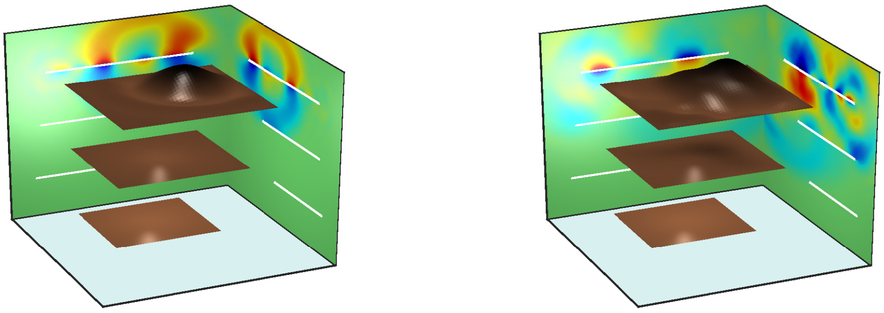
Nonlinearities in drums have a less dramatic importance than in gongs (see Section 3.2). Still, they can add a perceptually salient character to the simulated sound. Some remarks in this direction have been made by Fletcher *et al.* in [83], where pitch glides for the bass drum are reported at high striking amplitudes. An increase in the high frequency content of the spectrum of a tom-tom has been reported in [66]. More recently, a preliminary experimental study on nonlinearities in tom-toms has been performed by the Author in collaboration with M. Newton [186]. More on this topic will be said in Appendix A.

From a sound synthesis perspective, the type of nonlinearity included in the model can have a major influence on the quality of sound that is generated. The two nonlinear models described in Chapter 3, those of Berger and von Kármán, exhibit different behaviours at high striking amplitude. The Berger model, in particular, is not able to produce the bright attacks typical of real drums. For this reason, the nonlinearity included in the present model is of von Kármán type. In the engineering literature, there is ample consensus on the use of von Kármán equations for a membrane, and a complete review can be found in [109, 108]. Additional remarks on the dynamic von Kármán model applied to membranes can be found in [136]. To the knowledge of the Author, however, this physical description has never been applied to drum heads before in the context of musical acoustics.

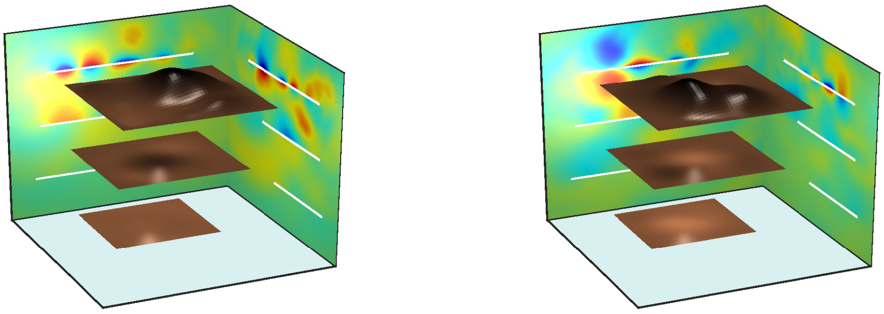
The present model for a double membrane drum consists of two von Kármán membranes



(a) $t = 0.45$ ms (b) $t = 1.00$ ms



(c) $t = 1.50$ ms (d) $t = 2.60$ ms



(e) $t = 4.00$ ms (f) $t = 6.00$ ms

Figure 6.4: Snapshots of the configurations of the multiple plate system in 3D after a raised cosine strike on the upper plate, at times as indicated. The vertical slices along the xz and yz planes at the centre of the box are displayed on the walls of the box. Notice how the pressure wave generated by the blow propagates inside the box and reaches the middle plate first, and then the lower one. Both plates are then set into motion, although the vibration amplitude is smaller than that of the upper plate. The scale of the plate displacements is altered by a factor 300 for illustration purposes. The sampling rate is 44.1 kHz.

coupled by a rigid shell and immersed in air. The implementation shares some similarities with the MP3D code described before (Section 6.2).

6.3.1 Description of the system

The physical model of a drum requires several interacting components: two membranes, a rigid cavity, and a finite enclosure of air. A schematic representation of the model is given in Figure 6.5. The two membranes are obviously the main components of the drum. The upper membrane, called batter head, is struck with a stick or a mallet, while the second membrane, called resonant or carry head, is set into motion by the pressure inside the cavity generated by the blow. The drumheads are defined over two 2D circular regions $\mathcal{M}_b$ and $\mathcal{M}_c$ with radius R, placed at vertical coordinates z_b and z_c, respectively. The centres of the two regions are aligned at $(x, y) = (0, 0)$. The transverse displacement $w_i(x, y, t)$ for each membrane $(i = b, c)$ is governed by the 2D wave equation for a stiff nonlinear membrane described in Section 3.3, with air coupling defined in Section 5.4.8. A rigid shell $\mathcal{S}$ with height $H_{\text{shell}} = z_b - z_c$ encloses the cylindrical region between the two membranes.

The acoustic field is described by the velocity potential Ψ and is governed by the 3D wave equation (5.11). At the walls of the computational box $\mathcal{V}$, absorbing boundary conditions are applied, see Section 5.2.2. Coupling conditions between the membranes and the air have been discussed in Section 5.4.

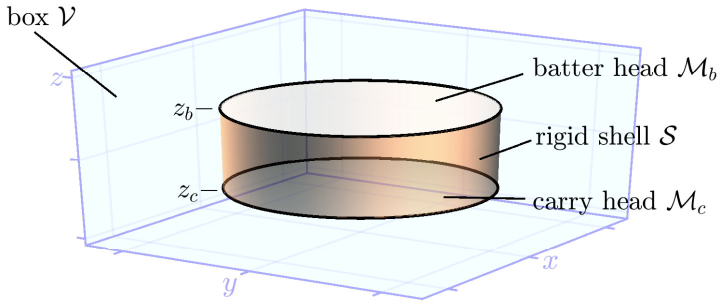


Figure 6.5: Geometry for the double membrane drum model. Two membranes $\mathcal{M}_b$ and $\mathcal{M}_c$ are positioned at vertical coordinates z_b and z_c, respectively, with axes aligned. A rigid shell connects the two heads, delimiting the air cavity between them. The system is embedded in a 3D box of air, with which it is coupled.

6.3.2 Numerical implementation

The numerical scheme for a nonlinear stiff membrane coupled with the air has been discussed in Section 3.3. Here, however, the difference lies in the circular geometry of the drum heads. A simple solution is the use of a staircase approximation for the circular boundary, as that presented in Section 2.5.2 for the plates. Clamped conditions are applied at the boundaries of the two membranes. A similar procedure must be applied to the rigid boundary of the shell, where Neumann conditions are applied. This has been discussed in Section 5.3.4.

The stability conditions determine the grid spacings for the two membranes and for the air. These are, in general, different, therefore interpolation becomes necessary when coupling conditions are to be implemented. This issue has already been dealt with in Section 5.4.2.

6.3.3 Energy conservation

The total energy $\mathfrak{h}_{\text{DMD}}$ of the double membrane drum is given by the sum of the contributions $\mathfrak{h}_{\text{a}}$ from the acoustic field, and $\mathfrak{h}_{\text{b}}$ and $\mathfrak{h}_{\text{c}}$ from the two membranes, such that

$$\mathfrak{h}_{\text{DMD}} = \mathfrak{h}_{\text{a}} + \mathfrak{h}_{\text{b}} + \mathfrak{h}_{\text{c}}. \tag{6.2}$$

The discrete energy for the acoustic field has been given in (5.30), while the energy for a membrane with von Kármán nonlinearity has been given in (3.73). Figure 6.6 shows the results in terms of energy conservation for a tom tom drum with radius $R = 10$ cm, height 12 cm and Mylar membranes with tensions $T_b = 2000$ N/m and $T_c = 1500$ N/m, excited with an initial raised cosine with diameter 1 cm and strike position $(x_0 = 5, y_0 = 0)$ cm. It is possible to see how the initial energy of the membrane is gradually transferred to the acoustic field. Normalised energy variations are on the order of machine precision.

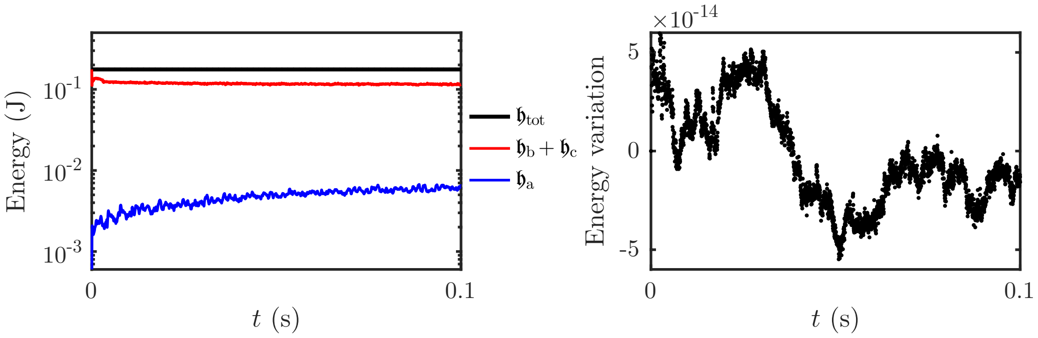


Figure 6.6: Energy for the double membrane drum. Left: Contributions to the total energy from the membranes and the acoustic field. Right: Normalised variations for the total energy of the system. The simulation sampling rate is 44.1 kHz.

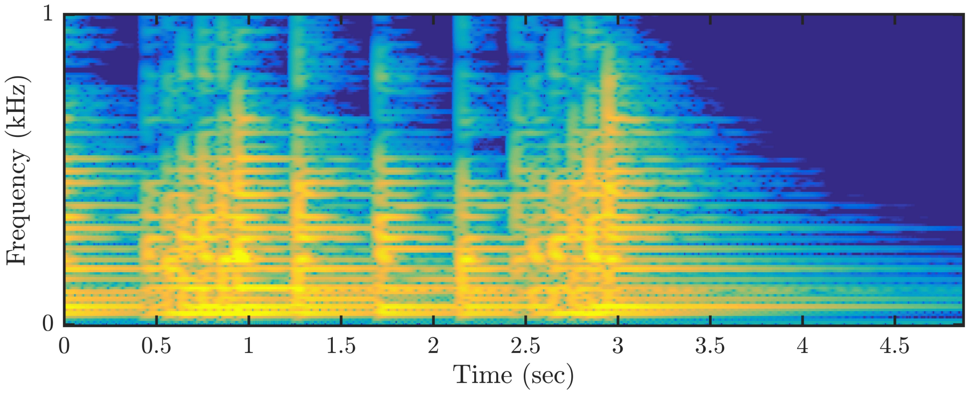
6.3.4 Computational cost

The main challenge of this model resides in the solution of the linear system presented in Section 5.4.8 for both membranes. Here, the same considerations broached in Section 6.2.4 for the MP3D system hold. Given the smaller grid size of a membrane with respect to that of a plate, however, the state size in this case is generally much larger than in the previous system. As discussed in [183], the solution of the linear system for the bass drum can take a considerable portion of the run time. Possible ways to increase the performance of the code have been presented in Section 5.4.8, and these apply also in the present case. The approach adopted by J. Perry for the porting of the MP3D system described in Section 6.2.4 can be extended to the present case, with the same benefits in terms of computational time. In fact, the two systems have been recently combined into a unique framework.

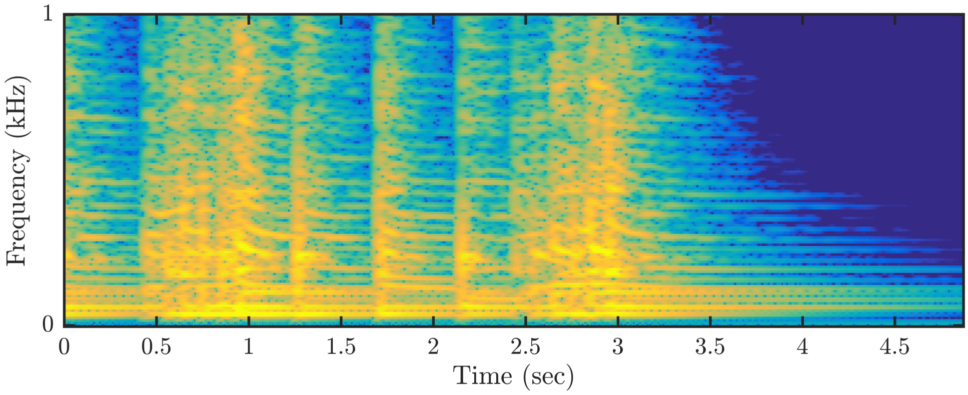
6.3.5 The bass drum

The bass drum is probably the biggest percussion instrument commonly used in the Western orchestra—the drumheads can have a radius of more than half a meter! Experimental investigations on this drum have been performed by Fletcher and Bassett [83], and have demonstrated the presence of strong nonlinear effects in the drumheads.

From a sound synthesis point of view, a linear simulation cannot reproduce the most salient features of the sound of this instrument, and a nonlinear model is therefore necessary. Figure 6.7 shows a comparison between the linear and the von Kármán model, for a typical rolling gesture performed on a bass drum with radius $R = 0.27$ cm and cavity height of 30 cm. Pitch glide effects and sharper attacks can be noticed in the spectrogram of the nonlinear simulation. These determine more dramatic and ultimately more realistic attacks for the high amplitude strikes. More on the influence of different nonlinear models on the quality of the simulated drum sounds will be said in Appendix A.



(a) Linear model.



(b) Nonlinear model.

Figure 6.7: Comparison between a linear and a nonlinear model for the membrane equations in the case of a bass drum with radius $R = 27$ cm and shell height of 30 cm. The von Kármán nonlinear model is able to produce pitch glide effects that are perceptually important in characterising the sound of a low pitched drum.

6.4 Snare drum

The snare drum is distinguished from other double membrane drums by the presence of a set of thin wires that are in direct contact with the lower membrane, which lend it its characteristic rattling sound. The first and probably still most important study on the acoustical properties of the snare drum is that of Rossing *et al.* [153]. Recent studies include [161, 204]. The Author has contributed to this topic with a preliminary experimental study on a simplified system done in collaboration with M. Newton [185] aimed at investigating the snare/membrane collision mechanism.

From a simulation point of view, the snare drum is a very challenging target, as components of all possible dimensions are present at the same time: the membranes are defined over 2D regions, the snares are described by 1D equations, the mallet is a lumped or "zero-dimensional" object, and of course the acoustic field is defined in 3D.

The main challenge here is to model the collisions between the snares and the lower membrane. This issue has been addressed in [184], where a novel energy conserving scheme for the simulation of distributed collisions has been proposed.

6.4.1 Description of the system

The description of the system is similar to that of the double membrane drum presented earlier (see Section 6.3.1), and the same notation can be adopted. Here, however, the nonlinearity in the membranes is not present. On the other hand, a set of snares is placed in contact with the lower membrane (see Figure 6.8). In a real snare drum, the snares end at a plastic or metal bridge which is connected to the rim of the membrane, and are kept in contact with the drum head by a tensioning mechanism (see Figure 6.8.) In order to simplify an already complicated system, in this model both ends of the snares are attached directly to the membrane. The N_s snares are modelled as a set of 1D objects, defined over regions $\mathcal{D}_s^{(i)}$, for $i = 1, \dots, N_s$, each described by the equation for a stiff string discussed in Section 4.3.1. The triple interaction between the snares, a membrane and the air has been analysed in Section 5.4.10, and the same considerations hold here.

6.4.2 Numerical implementation

The numerical implementation for the model follows that of the double membrane drum presented before. The complication here resides in the simulation of the collisions between the mallet and the batter head, and between the snares and the carry head. These have been dealt with in detail in Sections 5.4.9 and 5.4.10, respectively, to which the interested reader is referred. See also [184].

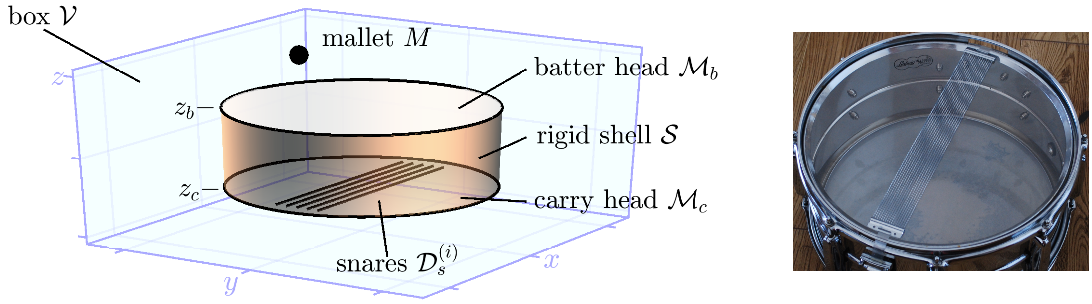


Figure 6.8: Left: Geometry of the snare drum. Two stiff membranes $\mathcal{M}_b$ and $\mathcal{M}_c$ are positioned at vertical coordinates z_b and z_c, respectively, with axes aligned. A rigid shell connects the two heads, delimiting the air cavity between them. A set of snares $D_s^{(i)}$ is placed in contact with $\mathcal{M}_c$. A lumped mallet M strikes $\mathcal{M}_b$. The system is embedded in a 3D box of air. Right: Picture of a real snare drum. Notice the metal bridges connecting the snares and the membrane.

6.4.3 Energy conservation

The total energy of the system is given by the contribution from the different components. In particular,

$$\mathfrak{h}_{\mathrm{SD}} = \mathfrak{h}_{\mathrm{a}} + \mathfrak{h}_{\mathrm{b}} + \mathfrak{h}_{\mathrm{c}} + \mathfrak{h}_{\mathrm{s}} + \mathfrak{h}_{\mathrm{M}} + \Pi_{\mathrm{s}} + \Pi_{\mathrm{M}}. \tag{6.3}$$

where the various terms on the right hand side are the energy of the acoustic field, the batter membrane, the carry membrane, the snares, the mallet, and the collision potentials, respectively. The energy $\mathfrak{h}_{\mathrm{a}}$ for the acoustic field has been given in (5.30), while the energy for a stiff membrane can be found in (2.164). The energy $\mathfrak{h}_{\mathrm{s}}$ for a stiff string has been given in (4.45), while the energy for the mallet is the kinetic energy of a mass. Finally, Π_{s} and Π_{M} represent the collision potentials for the string and the mallet, respectively.

By combining the energy conserving schemes discussed in Sections 5.4.9 and 5.4.10, it is possible to obtain an energy balance for the full drum that can be written as

$$\delta_{t-}\mathfrak{h}_{\mathrm{SD}} = 0, \tag{6.4}$$

in the lossless case. Figure 6.9 shows an energy plot for a typical simulation. The snare drum under consideration has two Mylar membranes with radius 15 cm, thickness 0.1 mm and tensions $T_b = 2000$ N/m and $T_c = 1410$ N/m, respectively. The air cavity is 15 cm in height. The mallet has mass $M = 15$ g and initial velocity 10 m/s. Five steel snares with $L_s = 24$ cm are in contact with the lower membrane. It is possible to see how the initial energy of the mallet is transferred to the other components, while the total energy of the system remains constant. Normalised energy variations are on the order of machine precision.

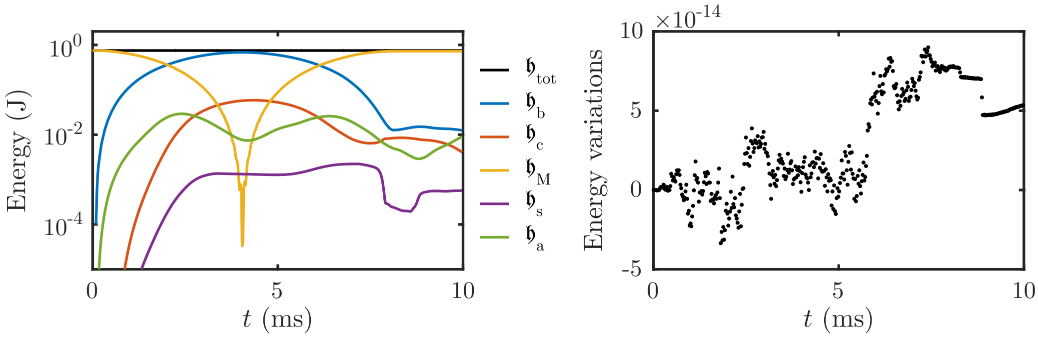
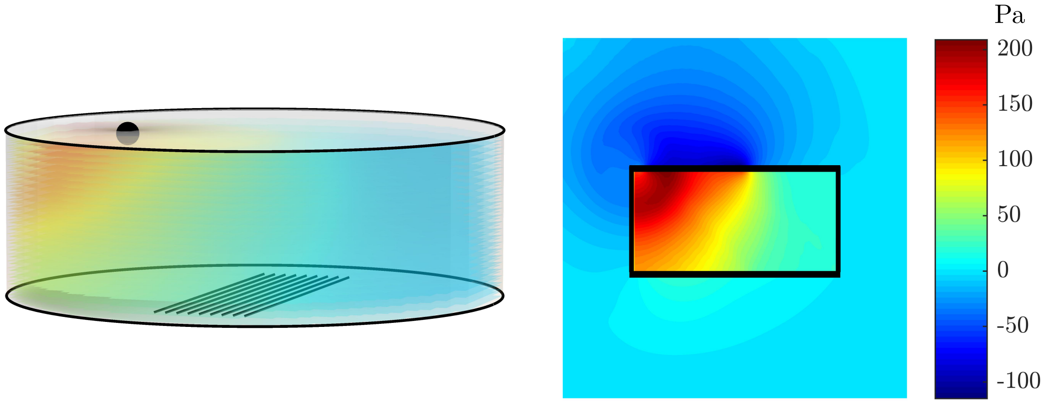


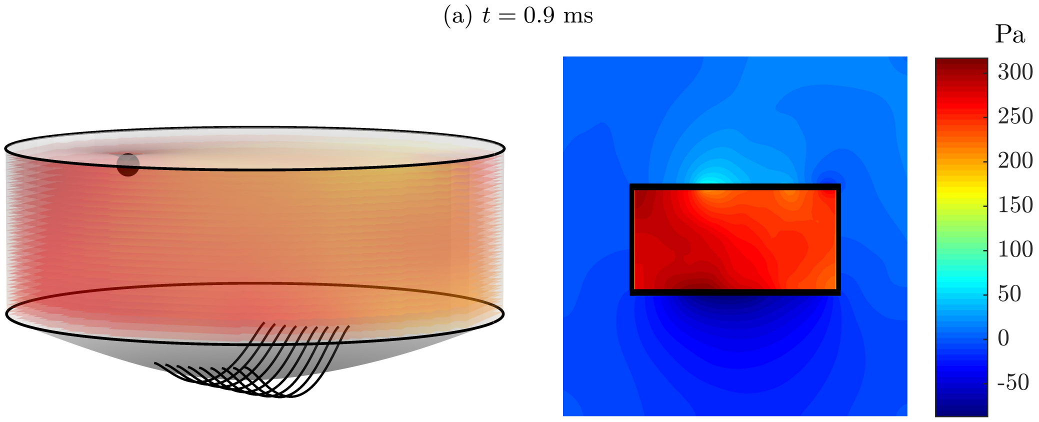
Figure 6.9: Evolution of the energy quantities for a snare drum simulation. Left: Contributions to the total energy (plotted in black) from the various components of the snare drum. The collision potentials are summed to the mallet and string energies, respectively. Right: Normalised variations of the total energy.

6.4.4 Evolution of the system

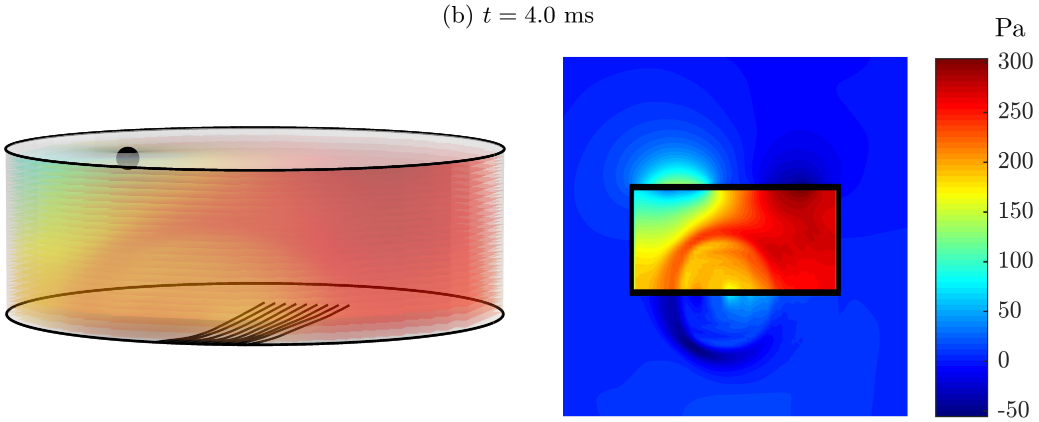
Consider now a snare drum in the lossy case, with absorbing conditions over the boundaries of the 3D box. The two membranes are now tensioned with $T_b = 3000$ N/m and $T_c = 2000$ N/m, and nine snares are in contact with the lower membrane. Figure 6.10 shows the configuration of the system at times as indicated. Also plotted is the pressure of the acoustic field over a vertical central slice along the y-axis, thus perpendicular to the direction in which the snares are aligned. When the mallet hits the membrane and starts to push it downwards, a region of positive pressure forms inside the drum. The build up of pressure inside the cavity causes the carry membrane to move downwards, pushing and launching the snares in the same direction. At $t = 7.3$ ms, the snares bounce against the lower membrane for the first time, creating a wave front of positive pressure that propagates inside the drum. Notice how an opposite wave front is generated on the other side of the snare membrane, as well. At $t = 9$ ms the mallet detaches from the batter membrane, and at $t = 11.6$ ms a second collision takes place between the snares and the membrane. Notice that up to this point the movement of the various snares is fairly coherent. Finally, at $t = 14$ ms the snares are launched again, and their motion starts to be randomised.



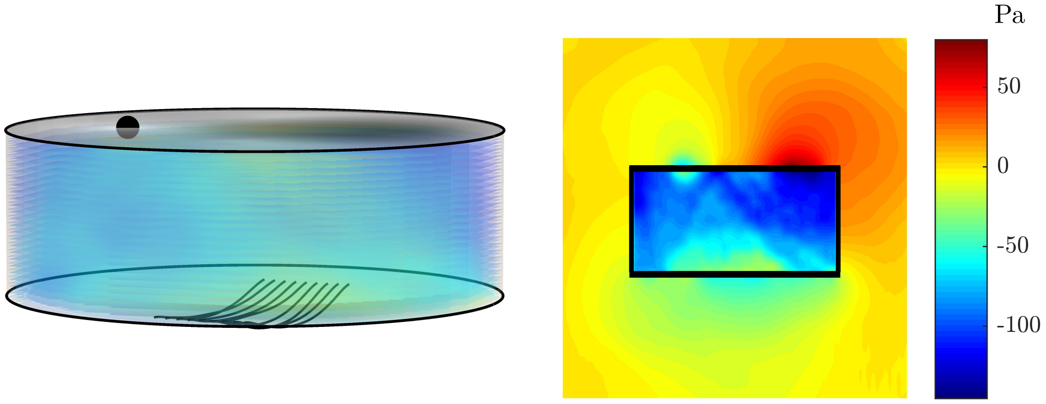
(a) $t = 0.9$ ms



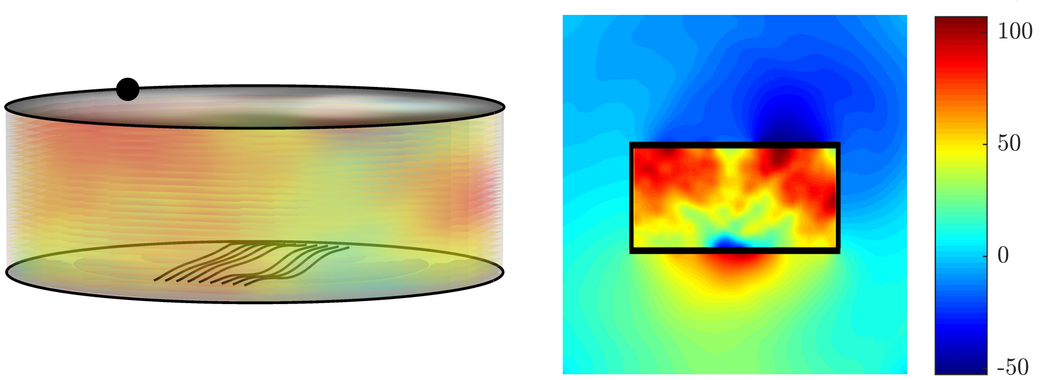
(b) $t = 4.0$ ms



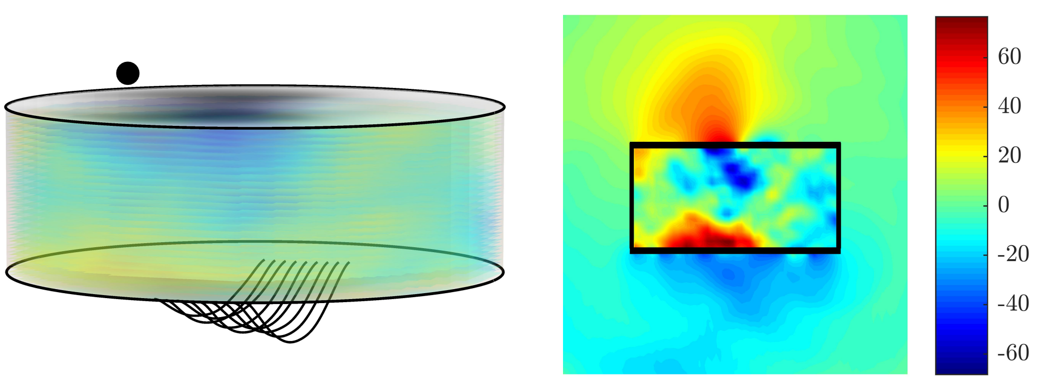
(c) $t = 7.3$ ms



(d) $t = 9.3$ ms



(e) $t = 11.6$ ms



(f) $t = 14.0$ ms

Figure 6.10: Evolution of the snare drum simulation at times as indicated. For each set of figures, a snapshot of the snare drum configuration (including the lumped mallet) on the left side, is accompanied with the correspondent state of the pressure variation along a vertical cross section of the box. A color bar indicates the pressure scale in Pa.

6.5 Control and scores files

So far, only the structure of the models and their numerical implementation has been discussed, but little has been said about what the user/composer can do in order to create sounds. In this section, some aspects about control and score generation are discussed.

6.5.1 Raised cosine strikes and score files

The first and perhaps the simplest possibility to control the instruments presented above is the use of raised cosine strikes. The user has the ability to inject strikes into the plates/membranes, which are modelled as raised cosine excitations in time, and are specified by two parameters, the maximum force $F_{\max}$ and the duration τ. Regardless of the particular sound synthesis environment, the strikes are fed into the equations of motion as an external force f_{exc}, whose mathematical definition is given by

$$f_{\text{exc}}(t) = \begin{cases} \frac{F_{\max}}{2}(1-\cos(2\pi t/\tau)) & \text{for } 0 \le t \le \tau \\ 0 & \text{otherwise.} \end{cases} \tag{6.5}$$

It is assumed that this force starts at $t = 0$ s, but with obvious modifications this expression can be translated at any time $t \in \mathbb{R}$. This lumped force can act at any point over the membrane or over a spatial distribution g, which is typically chosen as a spatial raised cosine as defined in (2.47). The equation for a stiff membrane with the external forcing term acting over a distribution g can thus be written as

$$\rho H \partial_t^2 w = \mathcal{O}_m w + g\, f_{\text{exc}}. \tag{6.6}$$

By varying the amplitude and the duration of the strikes, the composer is able to change the sounds created by the plates or drums. As a rule of thumb, shorter impulses create a brighter sound richer in high frequencies. Figure 6.11 shows two examples of raised cosine shapes depending on $F_{\max}$ and τ.

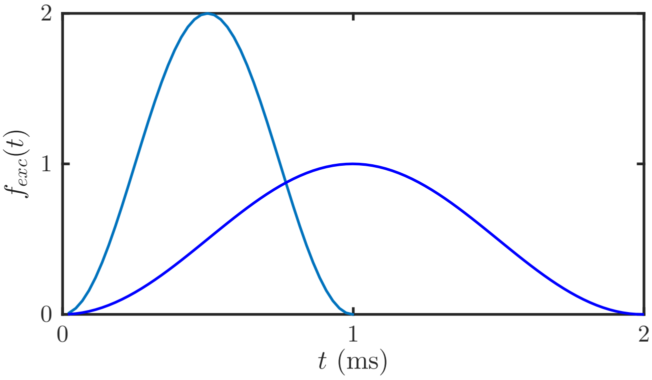


Figure 6.11: Different raised cosine strikes created with different values for $F_{\max}$ and τ. Shorter strikes determine a brighter output sound.

It is straightforward to extend the implementation of a single excitation to many consecutive strikes. Each of these is characterised by a maximum force $F_{\max}$ and a duration τ, together with the striking location (x_s, y_s) and the time of impact. All these parameters constitute a score file that is fed into the algorithm in order to create complex gestures, as anticipated in Section 6.1. An example of this is discussed in the next section.

6.5.2 Graphic interface for gesture generation

From a composer's point of view, the task of specifying a large number of parameters for a typical gesture can be daunting. To this end, the use of breakpoint functions has been explored. A simple graphic interface has been designed by the Author, together with S. Bilbao and P. Graham, and has then been implemented by P. Graham. Starting from a few numbers, the user can specify the profiles for the various parameters, thus obtaining a score which can be fine-tuned in a second stage.

The user has to insert the initial parameters:

- the sampling rate of the simulation (default value is 44.1 kHz);
- the start time of the gesture (default value is 0 s);
- the duration of the gesture (default value is 4 s);
- the granularity of the graph (default value is 2 points/s);
- the maximum number of strikes per second (default value is 20 BPS).

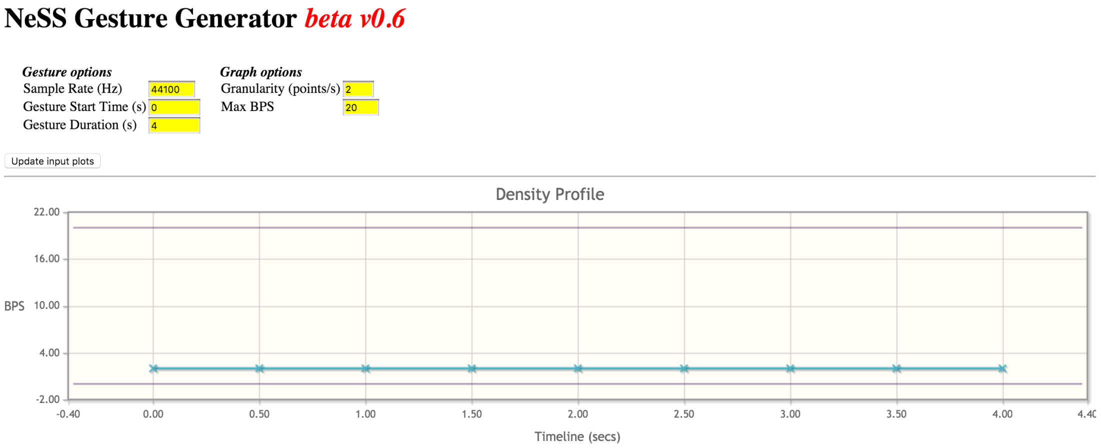
The value for the granularity determines the "degrees of freedom" of the gesture, by specifying how many equally spaced points can be found for each second of the gesture profile.

At this point, several graphs are generated, one for each of the following variables:

- density of strikes;
- randomness factor for the density;
- amplitude of the strikes;
- duration of the strikes;
- strike location along the x-axis;
- strike location along the y-axis;
- randomness factor for the strike location.

Each of the initially flat profiles can be modified by dragging the points of the breakpoint function up and down with the mouse, thus increasing or decreasing the value of the associated variable (see Figure 6.12). The inclusion of two randomness profiles introduces more flexibility in the gesture design.

Finally, the interface plots the graphs of the various parameters of the generated strikes, and the user can visually check the results for the final gesture. In particular, the system displays the values of the various strikes for each of the four parameters: amplitude, duration and x and y position. Figure 6.13 shows the results of the amplitude profile of the gesture for a typical case. Furthermore, a low sampling rate simulation is run on a linear steel plate, in order to allow for an auditory check of the gesture. A score file similar to that in Figure 6.14 is generated, and can be fed to the codes as explained in Section 6.1 in order to synthesise the sounds.

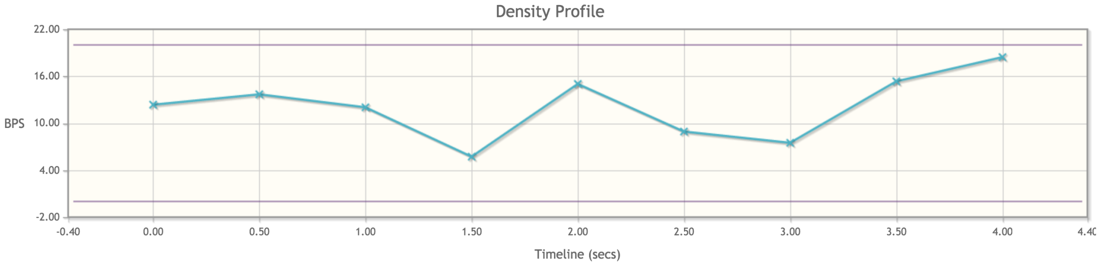


NeSS Gesture Generator *beta v0.6*

Gesture options
Sample Rate (Hz) 44100
Gesture Start Time (s) 0
Gesture Duration (s) 4

Graph options
Granularity (points/s) 2
Max BPS 20

(a) Initial flat profile



(b) User modified profile

Figure 6.12: (a) Portion of the gesture generator web page. The user is invited to select a few basic parameters that are then used by the system to generate a flat breakpoint function. (b) The user can drag the position of the points in order to specify the desired profile for the variable (the density of strikes, in this case). Similar graphs appear for all the other strike parameters.

6.5.3 Mallet excitation

Percussion instrument players are used to a wide variety of striking mechanisms to generate sounds with their instruments, such as drumsticks, brushes and soft mallets. A detailed simulation of such delicate interactions is of course desirable to achieve a high level of accuracy in the simulation. One first step in this direction is available for the snare drum, and it is the use of a mallet excitation. Although more physically accurate, however, this method presents some challenges from the control point of view.

It is easy to specify initial velocity and position for the mallet in order to generate a single strike, but it is more challenging to create complex gestures with it. One obvious way to proceed, in fact, is to drive the mallet with an external force. This, however, must be supplied both during the hitting phase, as well as in the release phase, when the mallet bounces off the instrument away from it. The force must, therefore, both accelerate and decelerate the mallet, in order to prevent the mallet from going too far from the surface. This challenge requires further investigation and a close work with the composer, in order to find the best implementation strategy. For these reasons, the mallet excitation does not currently fit into the score file framework that has been presented above.

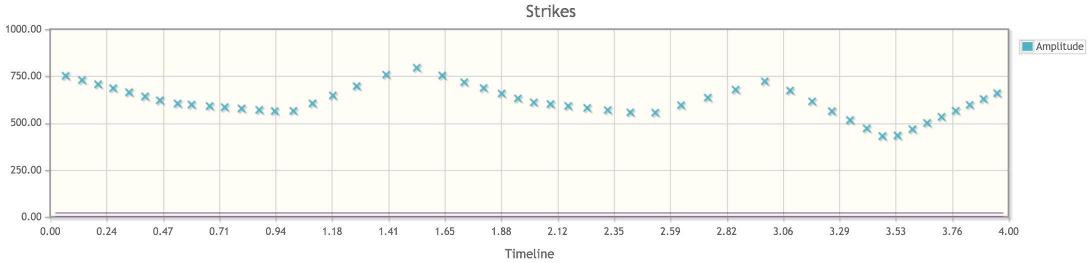


Figure 6.13: Amplitude profile for a typical gesture. Each point in the graph represents a strike, with parameters specified in the score file.

```
duration 4.000
strike 0.065 plate1 0.670 0.551 0.0011 751.591
strike 0.139 plate1 0.673 0.563 0.0012 726.996
strike 0.202 plate1 0.598 0.547 0.0015 705.989
strike 0.271 plate1 0.589 0.590 0.0019 683.260
strike 0.338 plate1 0.552 0.612 0.0022 660.809
strike 0.394 plate1 0.541 0.630 0.0025 642.286
strike 0.469 plate1 0.513 0.652 0.0029 617.427
strike 0.534 plate1 0.495 0.633 0.0029 603.725
strike 0.600 plate1 0.477 0.622 0.0026 597.195
strike 0.657 plate1 0.511 0.546 0.0023 591.555
...
```

Figure 6.14: Portion of the score produced with the gesture generator web page and represented in Figure 6.13. The first line determines the duration of the simulation. After this, each line starts with the word "strike", followed by the impact time, the label of the plate where the strike acts (when multiple plates are present in the virtual environment), the x and y normalised positions of the strike, the duration and finally the amplitude.

Chapter 7

Summary and conclusions

This chapter contains the summary of the results obtained in this thesis, as well as some concluding remarks and the directions of future work.

7.1 Summary

The papers cited in this section refer to the List of publications on page iv.

Chapter 2 - Linear plate vibration

In this chapter, the linear vibration of thin plates is presented. The governing equations are discussed in the continuous case first, and it is shown how boundary conditions can be derived by requiring the total energy of the system to be conserved. Rectangular and circular plates are considered. Then, an approach for solving numerically the equations of motion with the finite difference method is discussed. It is shown how energy techniques can be used in order to design a stable scheme, and how to derive consistent boundary conditions. Simulation results highlighting energy conservation and the behaviour in the frequency domain are presented.

An original contribution of this chapter is the analysis of the numerical energy variations patterns in terms of random walks. Additionally, it is shown how to apply energy analysis when dissipation is present, by adding back to the total energy of the system the accumulated dissipated power. Although similar ideas have been suggested before, it is shown how a careful implementation is required in this case to minimise round off errors.

Chapter 3 - Nonlinear plate vibration

The nonlinear vibration of thin plates is discussed in this chapter. Two well-known physical models are presented, those of Berger and von Kármán. The Berger model is described first in the continuous case, and then one possible finite difference discretisation of the equations of motion is presented. Some details about an optimised implementation are given.

More space is devoted to the von Kármán model. The equations of motion in the continuous case are presented in detail, with remarks about energy conservation and boundary conditions. One possible finite difference scheme is then analysed. Different implementation strategies are compared in terms of energy conservation and computation speed. One contribution of this work is the idea to use iterative methods to solve the linear system that is required in the scheme.

Although this approach produces less accurate results, the gain in terms of computation speed can be justified in sound synthesis applications.

Finally, an extension of these two nonlinear models to the case of membranes is discussed. The use of the Berger model in conjunction with membranes is common in the literature, but the von Kármán model has never been used in this context in the field of musical acoustics. This is another contribution of this work, which has been broached in Refs. A and E.

Chapter 4 - Collisions

In this chapter, the simulation of collisions is discussed. Penalty-based methods are applied to derive a collision force that depends on the mutual interpenetration of the colliding objects. A recently developed approach is explored in the case of lumped/lumped, lumped/distributed and distributed/distributed collisions. It is shown how energy techniques can be applied in these cases to design stable numerical schemes.

The basic ideas are broached first for the very simple case of a mass colliding against a rigid barrier. Then, the same approach is adapted to the simulation of the interaction between a lumped mallet and a membrane.

The original contribution of this work is in the design and implementation of a stable numerical scheme for the interaction between a distributed string and a stiff membrane (Refs. D and G). Particular attention is devoted to the boundary conditions applied at both ends of the string. It is explained how to derive a stable scheme when these are attached directly onto the membrane. Simulation results showing the conservation of the numerical energy in the three cases are presented.

Chapter 5 - Coupling with the acoustic field in 3D

The first part of this chapter constitutes a primer on 3D acoustics simulation. The basic equations for 3D wave propagation are discussed in terms of the acoustic velocity potential. Energy considerations are used to derive stable boundary conditions. A finite difference scheme is presented, together with the associated discrete energy.

A more extended section is then devoted to the coupling of 2D structures, like plates and membranes, with the acoustic field. Coupling conditions are derived from energy considerations both in the continuous and discrete case. In the particular form adopted here, this approach was developed by S. Bilbao in the case of linear and simple Berger-type nonlinear 2D structures. It is the first time, however, that such analysis is presented with full details.

The main original contribution of this work is the extension of air coupling to the plates in the presence of von Kármán nonlinearities (Refs. A and C) and with mallet and string collisions (Refs. D and G). Energy techniques must be carefully employed in order to obtain a stable scheme in these cases.

Chapter 6 - Virtual instruments

This chapter focusses on the construction of virtual instruments embedded in 3D by combining the energy conserving finite difference schemes presented in the previous chapters. Before discussing the three original virtual instruments which constitute the main contribution of this work, some general remarks about the structure of a typical sound synthesis code are given.

Then, an original sound synthesis environment is presented which is based on a set of thin plates under nonlinear conditions immersed in a 3D acoustic field with which they are coupled (Ref. C). Results showing the energy conservation of the scheme and the evolution of the model in a typical simulation are presented. Here, the design and numerical implementation in MATLAB of the prototype code has been performed solely by the Author. All the results presented in this chapter are obtained with this code. The porting of this program to CPU and GPU mentioned in this section has been done by J. Perry (Ref. L).

The second original instrument is a model for a double membrane drum with von Kármán nonlinearities included in the drumhead equations (Ref. A). The continuous and discrete models are discussed, together with the results showing the energy conservation of the schemes. As for the previous model, the design and numerical implementation in MATLAB of the prototype code has been performed solely by the Author. All the results presented in this chapter are obtained with this code. The porting of this program to CPU and GPU mentioned in this section has been done by J. Perry (Ref. L).

Finally, the third virtual instrument is an energy conserving model for a snare drum excited with a mallet (Ref. G). Results showing the energy conservation of the scheme and the evolution of the system during a typical simulation are presented.

Additional remarks regarding the control of these instruments conclude this chapter. In this section, a graphic interface for the generation of score files is referred to. This has been designed jointly by the Author, S. Bilbao and P. Graham. Its implementation as a web-based interface is due to P. Graham.

Appendix A - Experimental investigation of nonlinearities in drum membranes

In this appendix, a preliminary study on nonlinear effects in drum membranes is presented. This has been conducted in collaboration with M. Newton at the University of Edinburgh and has been the topic of a previously published paper. The data analysis performed in Ref. E is revisited here, with an additional set of data at very high striking excitations not previously considered. Pitch glide effects can clearly be seen in the experimental data.

An improved numerical simulation is also presented, which compares the models of Berger and von Kármán for nonlinear membrane vibration. Pitch glide effects are produced at high excitation amplitude, as expected, but the two models present slightly different features. The Berger model, in particular, produces pitch glides that are much wider than the von Kármán case. It is argued that the von Kármán model might give results in better agreement with experiments, although further investigation is needed for a conclusive answer. In terms of sound synthesis, however, the outputs generated with the von Kármán model are more realistic that those with the Berger model.

Appendix B - Finite precision effects in FDTD schemes

This appendix offers novel insights on some round off effects that appear in finite difference schemes. First, it is clarified what is meant by conservation of energy "to machine precision", and how energy variations should be calculated in order to highlight the correct quantisation. This is an original contribution of this work.

Then, the implementation of a numerical scheme for the simple harmonic oscillator is analysed, both in the lossless and lossy case. A second order implementation is compared with the corresponding first order system. It is found that the first order system behaves better than the second order implementation from an energy conservation point of view, and that some numerical errors can be avoided with a different implementation.

The case of a free particle is analysed after. The second order implementation and the first order system are compared numerically, and an explanation for the quantised jumps that can be seen in the second order energy plot is proposed.

Finally, round off errors are analysed in the case of the 3D wave equation. It is demonstrated how a naïve implementation can lead to a linear drift in the energy calculation, and how it is possible to cure this artefact.

7.2 Concluding remarks

In this work, it is demonstrated how physics-based finite difference simulations can be used to produce provably stable algorithms for sound synthesis. The techniques presented here rely heavily on a discrete energy quantity that is conserved in the lossless case, and that is non-negative when the stability conditions for the scheme are satisfied. This approach is obviously not new, but is based on a vast literature of previous works, which have been acknowledged throughout this work.

Why finite differences?

After the extensive treatment of finite difference schemes given in this work, it is perhaps worth trying to weigh the advantages and disadvantages of this method, specifically when applied to the simulation of percussion instruments. Different approaches are in fact available, and have been successfully adopted in the past. Digital waveguides, for example, have been chosen by several researchers to simulate embedded drums [2, 117] and plates [193]. Although some attempts at incorporating nonlinear effects have been made, it is still unclear how to simulate strong physics-based nonlinearities as those of the von Kármán model in this context.

For some problems, modal methods are an attractive option, particularly when the system to be simulated is linear. Recent advances in the field by Ducceschi and Touzé [73, 74], however, have produced modal simulations even for the von Kármán plate equation, both for scientific and sound synthesis purposes. In this case, the particular structure of the problem has allowed for an efficient numerical implementation, that is just a few times beyond real-time for typical plates [75]. Another advantage of this approach is the versatility in the implementation of losses in the system—a different coefficient can, in fact, be chosen for each individual mode. Here, the difficulty resides in the calculation of the nonlinear coupling coefficients that govern the evolution of the system. These must be recomputed every time the physical parameters of the plate are changed, and require a large amount of computer memory for storage. A more general difficulty with modal methods is the lack of flexibility to handle non-standard geometries and coupling with different components.

A well-known alternative approach to numerical simulation is the finite element method [210]. This has been adopted in several occasions in musical acoustics and sound synthesis applications (see, e.g., [146, 37, 68, 46]). The starting point of this approach is the use of shape functions that approximate locally the solution to the problem in the spatial domain, by considering the initial problem under a variational formulation. Time-integration is then performed using standard techniques, such as Runge-Kutta methods, which can become very expensive in terms of memory storage when many past steps are considered for accuracy. Finite element methods are very attractive whenever the problem at hand is defined over an irregular domain. The downside, however, is the lengthy process that is required to set up the discrete mesh, which is generally carried out using external software packages over which the user has little or no control. Given the delicate problems generally found in sound synthesis applications, such as numerical bandwidth and dispersion, this may not always be a wise choice. It is not uncommon, in fact, for these software packages to produce grids with large variations in the elements' size and shape. This can severely harm the quality of the synthesised sound, as the discussion on polar grids in Section 2.5.1 should clearly illustrate.

Finite difference methods have several advantages, which should be clear from the models presented in this work. First of all, they are fairly simple to program. The various schemes can generally be written as simple matrix-vector multiplications. An additional strength is the possibility to simulate interacting elements of various nature. In this sense, the snare drum model presented here is an example of how different components, ranging from the lumped mallet to the 3D acoustic field, can be combined together in the presence of collision nonlinearities. The precomputation stage of a typical simulation, as described in Section 6.1, is also relatively inexpensive in terms of memory storage, given that the finite difference operators in matrix form have generally a very sparse structure. This, however, only holds in the case of very simple geometries, such as those presented here.

One problem that is often encountered in practical applications is the simulation of geometries that are not well described by Cartesian coordinates. As discussed before, circular domains are very common in drums and gongs, but their simulation within the finite difference framework can be done only at the price of some approximations. A new direction that is emerging is that of finite volume methods. Although not new (see, e.g., the classic book by LeVeque [127]), these techniques appear to be the natural extension of finite difference methods, without being restricted to regular grids. Unstructured meshes, for example, can successfully be applied near curved boundaries to better model the domain under investigation. Applications of this approach to room acoustics have been reported recently by Bilbao [21] and Hamilton [96, 97], while preliminary studies on plates [98] have also been performed. One challenge that remains open to this date is the application of the finite volume method to the case of circular plates in the presence of free and simply supported boundary conditions. Preliminary attempts performed jointly by Bilbao and by the Author have so far proven inconclusive. It is beyond doubt, however, that a successful implementation of this method, incorporating also a von Kármán type nonlinearity, would greatly improve the accuracy of the models presented in this work.

Computational cost

One issue that still hampers the full usability of the virtual instruments presented in this work is their high computational cost. At audio rate, in fact, the numerical codes can take hours to generate a second of output. A successful case study has been presented, that of the multiple plate environment in 3D and the double membrane drum, where a huge increase in computation speed has been obtained with a hybrid approach using a combined CPU and GPU implementation. A speed-up of 60-80 times against the original prototype code has been obtained for typical simulations. In this form, the code has been used by composers G. Delap and C. Chafe for their works.

The development of new techniques similar to those presented here is obviously desirable to increase the computation speed of these algorithms, with the ultimate goal being their use in real time. Although still a long way ahead, at least for the models presented in this work, recent advances in finite difference simulation of complex systems by Webb and Bilbao [203] give hope that this goal may become a reality.

Energy conservation and round off errors

Several different PDEs have been analysed in this work, and the numerical behaviours of the finite difference schemes that discretise them have been discussed. These can be combined together in order to create complex systems with several interacting components, with coupling conditions determined again by energy considerations. One question that remains open is how to extract more information from the energy variation patterns that are produced. In many occasions throughout this work, it has been pointed out how round off errors can play a considerable part in the energy conservation of the scheme.

An interesting perspective that is followed in this work is that proposed by Henrici in his seminal work [101], and pursued recently by Hairer and other researchers [94, 197]. Within this framework, different energy variation patterns are caused by the propagation of random round off errors in the scheme. As long as the energy patterns behave like random walks, the scheme can be considered well designed. This approach requires that a statistical analysis over a large number of simulations be performed, in order to ascertain the average behaviour of the scheme. This is obviously a lengthy process, as it requires hundreds or thousands of simulations to be run. Alternative approaches, based ideally on a single run of the code, and currently (to the knowledge of the Author) not available, are then highly desirable.

Appendix A

Experimental investigation of nonlinearities in drum membranes

The experimental investigation of percussion instruments has a long history, which dates back at least to the works of Lord Rayleigh [169]. The most systematic and extensive contribution in this field, however, has been given by Rossing and Fletcher, who performed several studies on various types of percussion. Their results are collected in the classic textbooks [84, 154, 152]. Many experimental studies on drums are available—target systems include bass drums [83], tom-toms [66, 149], timpani drums [151, 60] and snare drums [153]. Recent works by Worland *et al.* [205] have analysed the vibrations of a circular membrane and compared them with the theoretical predictions.

Nonlinearities play a major role in gongs and cymbals, and an intensive investigation has been conducted, both experimentally and theoretically, by Touzé, Thomas and Chaigne in recent years [187, 174, 54]. One interesting question is if such nonlinearities are present in drums, and what importance they have in determining the perceived sound. Nonlinearities in drum membranes, in fact, have not received much attention, apart from sporadic comments in the works of Fletcher and Rossing (see, e.g., [83] and [84, p. 608]). The Author has performed a preliminary experimental investigation on a floor tom drum [186] in collaboration with M. Newton at the University of Edinburgh. The aim of the experiment was to obtain quantitative measurements from the real drum to compare with the results of numerical simulations. This study and its results are described in the remainder of this chapter.

A.1 Preliminaries

A.1.1 Experimental setup

The tom-tom used for the experimental investigation has a radius $R = 20$ cm and height 42 cm. The measured thicknesses of the two membranes were 0.175 mm for the batter head, and 0.19 mm for the carry head. The drum was then positioned over a flat surface, with drumheads aligned vertically and perpendicular to the surface itself. In order to reduce rigid body movements during the strikes, rubber strips were positioned at both sides of the drum. This provided a stable seat for the instrument, while at the same time increasing the friction against slipping. The batter head was oriented towards the edge of the support surface, and a small patch of reflecting material was attached at the centre of it. On the opposite side, a Polytec OVF-5000 laser Doppler vibrometer (LDV) was pointed at the reflecting patch, such as to measure the displacement of the drum head. Sounds were recorded with a Bruel & Kjaer Type 4134 near field microphone, positioned at half a radius of distance from the centre of the membrane, at about 5 cm from the surface. The data acquisition was achieved with a Bruel & Kjaer PULSE system connected to a PC. This setup allowed for the synchronous recording of the various streams of data, such that the microphone signal could be exactly aligned with LDV signal. A picture and a schematic representation of the experimental setup are shown in Figure A.1.

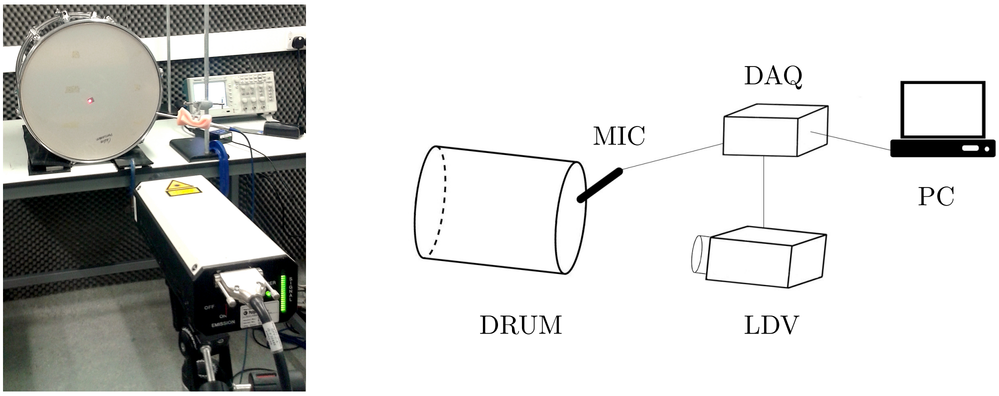


Figure A.1: Left: Picture of the experimental setup, showing the drum, the LDV and the microphone. Right: Scheme of the equipment used. The microphone (MIC) and the LDV are connected to the digital acquisition system (DAQ) and then to the PC.

A.1.2 Striking mechanism

Different mechanisms for striking the drum in a consistent way were tested. A pendulum-like device was initially adopted, consisting of a thin wire with attached a heavy spherical weight. This was hanging from a fulcrum that was suspended by a heavy support, while being able to rotate freely along a predefined direction. By lifting the weight from its equilibrium position and letting it go under gravity, the device could be used to strike the drum at a given location. Preliminary data were collected, but the force generated by the pendulum was found to be insufficient to trigger the nonlinear effects to a considerable level. For this reason, this strategy was discarded, and it was decided to strike the drum with a drum stick. Several different force levels were recorded and studied, from the linear to the strongly nonlinear regime, with repeated strikes for each level.

A.1.3 LDV signals

As no calibrated striking device was available, a quantifiable relation between the striking force and the displacement of the membrane cannot be given. The recorded signals, however, could be grouped together *a posteriori* based on the similarities that they display, hinting at compatible initial conditions. From the original data, collected with the LDV and the microphone, four different sets of signals can be identified, each corresponding to different striking amplitudes: low, medium, high, and very high. For each of these sets, three signals are taken into consideration, and can be considered three distinct repetitions at the same striking amplitude. All measurements were recorded independently one from the other.

The first challenge in the data analysis is that strikes with a similar amplitude need to be temporally aligned in order for all three to have the same starting point. Finding the exact initial sample of the strike within the signal is almost impossible, especially at low amplitudes, when the initial slope of the curve is almost zero. The strategy adopted, therefore, is to find the sample corresponding to the first maximum for each signal, and to align the vibrometer signals based on that. The same trimming can be applied to the microphone signals, as well, since these data are synchronised with the vibrometer.

The data used for the analysis are plotted in Figure A.3. Here, the first 1.5 s of the signals are shown, together with a zoomed portion over the first 0.1 s. The different amplitudes of the signals are apparent from the plots on the left hand side, corresponding to the four different striking amplitudes adopted. The compatibility of the initial conditions used between different strikes at similar amplitude can be checked by noticing the similarity between the different signals on the right hand side of Figure A.3.

A quantitative evaluation of the intensity of the strikes can be obtained by analysing the displacement of the membrane recorded with the LDV. Figure A.2 shows the initial part of the vibrometer signal for four different striking forces. As in the previous case, the various signals need to be aligned along the temporal axis. This time, the minima have been aligned at $t = 0$ s, as they correspond to the maximum displacements of the membranes. As expected, larger vibration amplitudes are produced by stronger blows. At low striking force, the maximum displacement recorded was of about 2 mm, at medium force of just below 4 mm, at high force 5 mm, and about 6 mm at very high striking force. Another interesting fact that can be noticed in Figure A.2 is the shorter wavelength of high amplitude signals. This hints at a change of frequency in the initial portion of the drum sound depending on the excitation amplitude. This is typical of pitch glide effects.

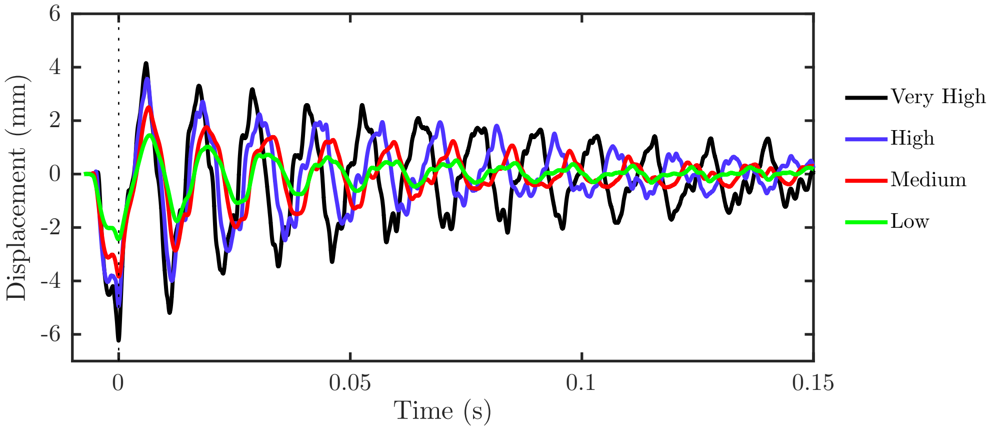
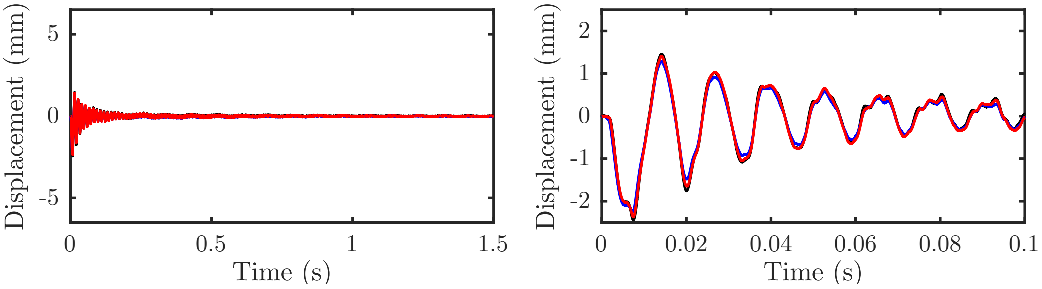
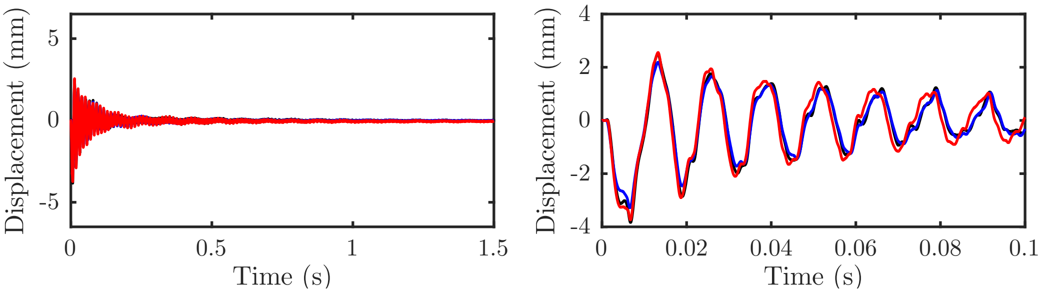


Figure A.2: Comparison of the vibrometer signals for different striking amplitudes. As expected, strong blows cause the membrane to vibrate at larger amplitude than low ones. Nonlinear effects depend on the ratio between the displacement and the thickness of the membrane.

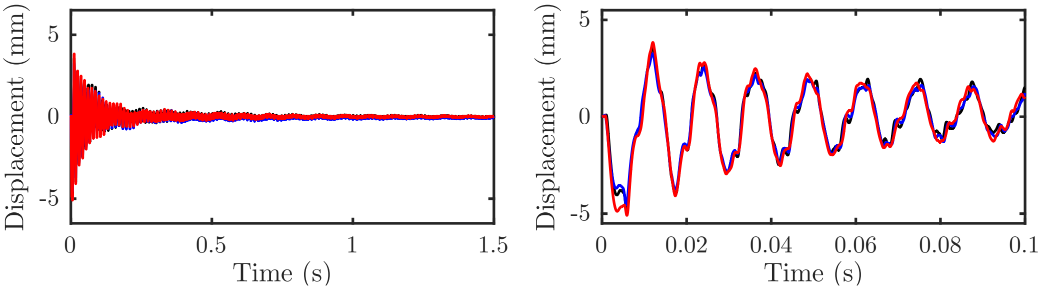
(a) Low amplitude



(b) Medium amplitude



(c) High amplitude



(d) Very high amplitude

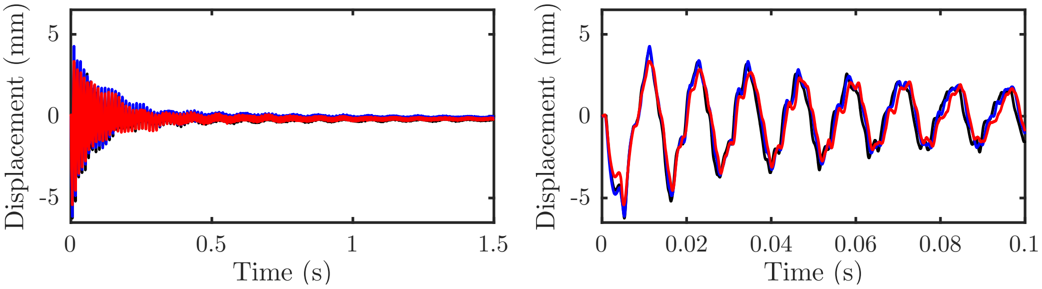


Figure A.3: Data from the vibrometer signals at different striking amplitudes. Three repetitions are shown in black, blue and red for each amplitude. Left: First 1.5 s of the signals. The same scale has been used in the vertical axis, in order to render apparent the different amplitude of the signals. Right: Zoomed portion of the first 0.1 s of the signals. Here, the similarities between the different repetitions are visible, meaning that comparable strikes have been injected.

A.2 Data analysis

A.2.1 Linear modes

A first step in the present study is the determination of the frequencies at which the upper membrane vibrates in the linear regime. These can be obtained by plotting a Fourier spectrum of the entire vibrometer signal at low amplitude excitation. In this case, as will shortly be seen, the peaks in the spectrum do not shift position over time, thus signifying that the vibration is in the linear regime. The results of this analysis are plotted in Figure A.4.

Peak frequency determination

Given the finite precision of the Fourier transform and the limited number of data points available, interpolation techniques must be used in order to extrapolate the frequency of the various peaks. A very simple approach consists in zero-padding the original signal [165]. Zero-padding can be thought of as a cheap way of interpolating a signal as it introduces more resolution in the Fourier transform without changing the information that it carries. This produces smoothed out peaks, that near the maximum can be approximated by a parabolic function. Depending on the level of zero-padding used, the frequency of the maximum can be assumed to be the frequency of the peak.

Better results, however, can be achieved through quadratic interpolation near the maximum. It has been noticed, in fact, that near the maximum the peak has a parabolic shape. Thus, by fitting a parabola through the maximum and its two neighbouring points, it is possible to achieve results with a considerable degree of accuracy. This is the approach adopted in the remainder of this chapter.

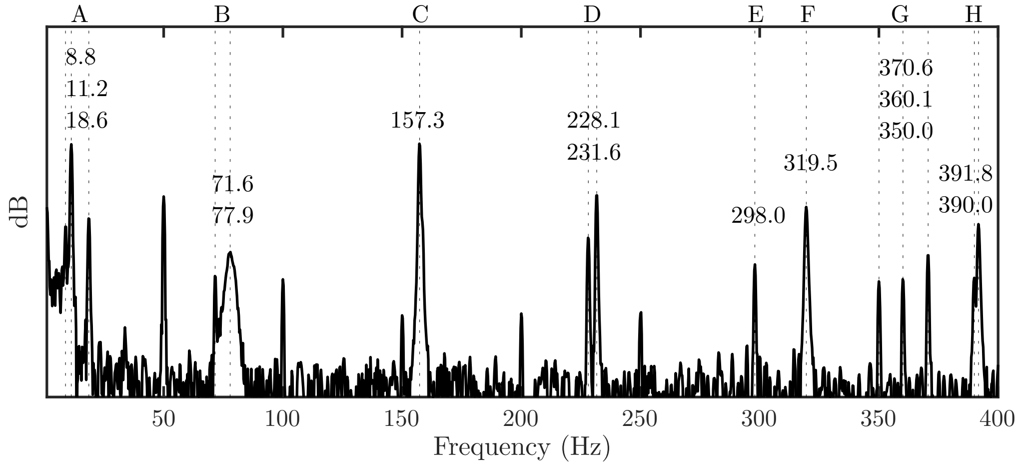


Figure A.4: Spectrum of the upper membrane, obtained from a Fourier analysis of the averaged vibrometer signals at low amplitudes. The frequencies of the peaks up to 400 Hz are reported in the Figure. The peaks at multiples of 50 Hz up to 250 Hz have been ignored as they are related to background noise.

Results

Three first peaks appear in the linear spectrum of the low amplitude signal at frequencies of 8.8 Hz, 11.2 Hz and 18.6 Hz, and might be due to the interaction between the drum and the supporting table. Then, a doublet of peaks appears with very close frequencies of 71.6 Hz and 77.9 Hz. The most prominent peak of the spectrum is located at 157.3 Hz. Up to 400 Hz, several other peaks appear at frequencies indicated in Figure A.4. Sharp spikes can also be noticed at multiple frequencies of 50 Hz, up to 250 Hz. These represent background noise coming from interferences with the electrical equipment, and can be disregarded from the analysis.

It is convenient, for the following analysis, to label the peaks or groups of peaks with letters[1]. As some of them are very close in frequency, it is sometimes easier to refer to pairs or triplets of peaks with the same letter, differentiating them by using a subscript number. These labels, together with the frequencies of the various peaks, are indicated in Figure A.4.

A.2.2 Pitch glide effects

In order to estimate pitch glide effects, one possibility is to analyse an initial portion of the signal and to compare it with a later portion, and to repeat the process for all the different striking amplitudes. The strategy adopted consists in averaging the vibrometer signals for every striking amplitudes, thus obtaining four signals from the initial twelve. Then, for each signal, a portion of 0.25 s (16384 samples at the acquisition rate of 65536 Hz) at $t = 0.02$ s is selected, and compared with one of equal length at $t = 1$ s. The choice of ignoring the first 0.02 s of the signal is to prevent the analysis from being altered by the presence of the mallet, which at that point has already detached from the drumhead. The results of this analysis are plotted in Figure A.5.

Low amplitude

At low amplitudes, the peaks simply decay in intensity, without changing position. This is typical of linear systems, whose behaviour can be well described by a superposition of modes. Here, excitation amplitudes were not enough to trigger nonlinear effects. An interesting phenomenon involves the peak B. The initial frequency is 77.9 Hz, while the final one is 71.6 Hz. That this is not a pitch glide is confirmed by plotting a Fourier spectrum over a longer window. Both peaks can be seen simultaneously in this case, with B_1 gradually prevailing over B_2.

Medium amplitude

In the medium amplitude case, a slight shift can be noticed in all of the peaks above C. The deviations observed are between 6 and 11.2 cents. The analysis on peak B shows an initial frequency of 76.5 Hz, which is 1.4 Hz lower than the linear value. In this case it is not clear what caused the discrepancy, and additional investigations would be required.

The values of the initial frequencies for the various peaks together with the relative variations in cents are reported in Table A.1.

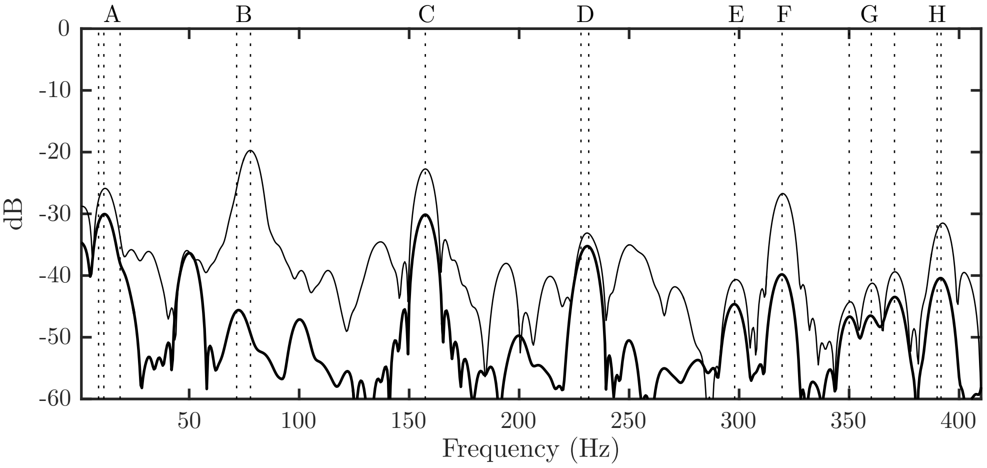
---

[1] There is obviously no connection between the frequency of the peaks and the musical notes!

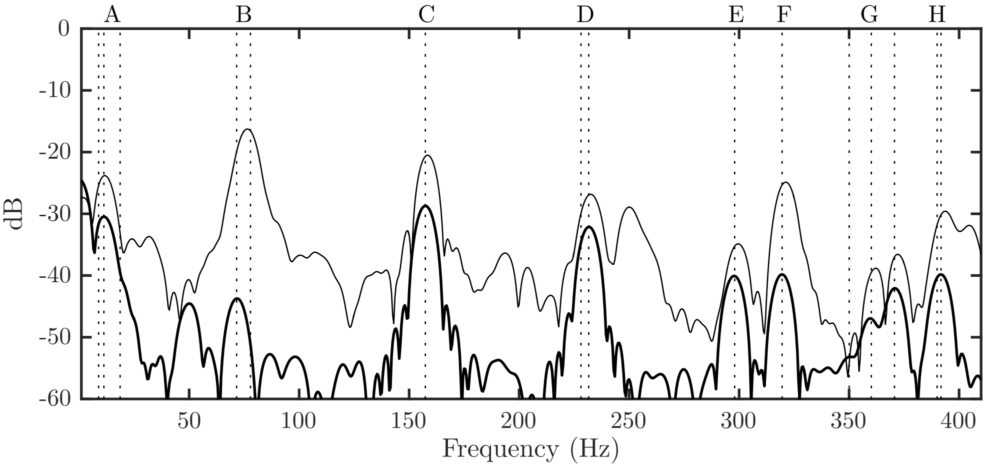
Peak	**Linear Frequency (Hz)**	**Initial Frequency (Hz)**	**Cent deviation**
C	157.3	158.4	11.2
D_2	231.6	232.5	6.5
E	298.0	299.5	9.1
F	319.5	321.2	9.0
G_2	360.1	362.1	9.6
G_3	370.6	372.1	6.8
H_2	391.8	393.7	8.4

Table A.1: Frequencies for the shifted peaks at medium excitation amplitude, together with the variations in cents from the linear values.

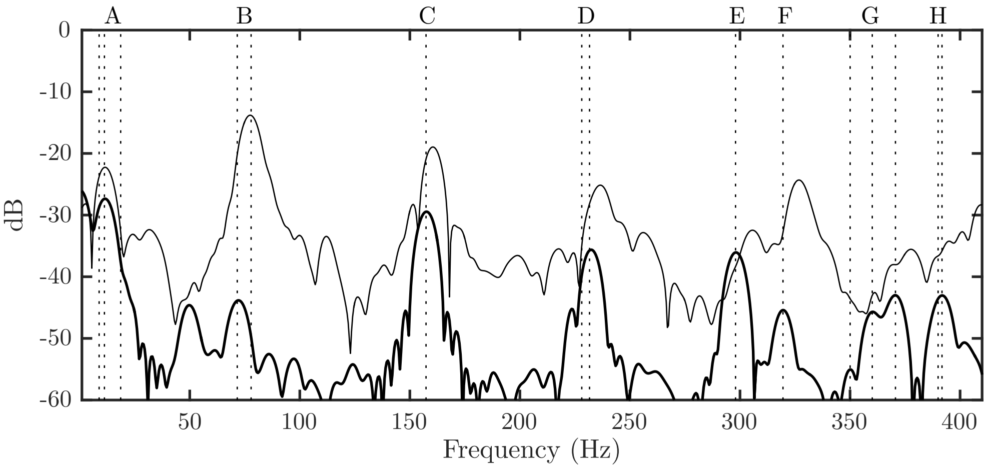
(a) Low amplitude



(b) Medium amplitude



(c) High amplitude



(d) Very high amplitude

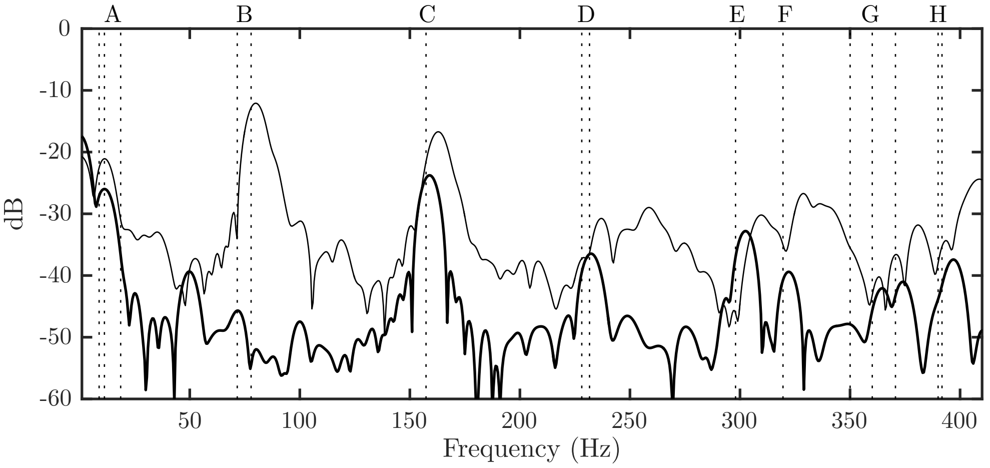


Figure A.5: Spectra for the averaged vibrometer signals at striking amplitudes as indicated. For each plot, the spectrum of a portion of 0.25 s of the signal at time $t = 0.02$ s is plotted with a thin line, while the spectrum for a later portion with identical duration at $t = 1$ s is plotted with a thick line. The vertical dotted lines represent the frequencies of the peaks in the linear regime. The shift of some peaks in the initial spectra at increasing excitation amplitude indicates the presence of pitch glide effects.

High amplitude

At high amplitude, a change in frequency becomes noticeable for most of the peaks. All the groups from C to H, in fact, exhibit a glide, even though for the doublet D it is difficult to resolve between the two frequencies, and a single maximum appears. Only the upper peak in the triplet G is clearly present. Pitch glides in this case range from 32.6 to 44.1 cent deviations with respect to the linear peaks. The behaviour of the peak B is again puzzling. The initial frequency appears to be at 77.5 Hz, therefore still below the frequency of B_2 measured in the linear case. No clear explanation to this phenomenon can be suggested.

The values of the initial frequencies of the peaks which can be most clearly identified are reported in Table A.2, together with their relative difference from the linear values.

Peak	**Linear Frequency (Hz)**	**Initial Frequency (Hz)**	**Cent deviation**
C	157.3	160.5	33.9
D_2	231.6	236.5	36.2
E	298.0	305.7	44.1
F	319.5	326.8	38.8
G_3	370.6	377.7	32.6

Table A.2: Frequencies for the shifted peaks at high excitation amplitude, together with the variations in cents from the linear values.

Very high amplitude

At very high excitation amplitude, the frequency shift becomes even more apparent. Here, the doublet B presents a single peak appearing at 80 Hz, with a deviation of 46.1 cents with respect to the frequency of B_2. Frequency shifts of up to 66.6 cent deviation can be observed. Notice that at this stage, hysteresis effects seem to be present. As a matter of fact, some peaks (C, E and F) set onto a final frequency that is shifted from the value in the linear regime.

The values of the frequencies for the initial spectrum of the signal are reported in the following table, together with the cent deviations with respect to the linear values.

Peak	**Linear Frequency (Hz)**	**Initial Frequency (Hz)**	**Cent deviation**
B_2	77.9	80.0	46.1
C	157.3	162.9	60.0
D_2	231.6	237.0	39.7
E	298.0	309.7	66.6
F	319.5	328.8	49.5
G_1	350.0	363.2	63.6
G_2	360.1	371.0	51.4
G_3	370.6	380.9	47.3

Table A.3: Frequencies for the shifted peaks at very high excitation amplitude, together with the variations in cents from the linear values.

A.2.3 Comments

The analysis in the previous sections has highlighted the presence of pitch glide effects in drum membranes. These appear to be present at high striking amplitudes, when the corresponding vibrations of the drumhead are large. This is in line with what has been reported by previous researchers (see, e.g., [83] and [84, p. 608]). The data presented before show that pitch glide effects start to appear even at moderate amplitudes for high frequency peaks. Shifts of up to 11.2 cents have been observed in this case (see Table A.1). Wider glides were reported at high and very high frequencies, with a maximum values of 66.6 cents for the peak E at very high striking amplitudes (see Table A.3).

A.3 Numerical simulation

The experimental results discussed in the previous sections can be compared with numerical simulations performed using the finite difference model for nonlinear double membrane drums presented in Section 6.3. A similar study has been done in [186], with the aim of comparing two different nonlinear models for membrane vibration, those of Berger and von Kármán.

As some of the parameters of the experimental system could not be measured, namely the density and the tension of the two membranes, an exact comparison between experiment and simulation cannot be made. Besides, the striking mechanism used in the experiment is fundamentally different from the raised cosine excitation injected in the numerical model. Therefore, the following analysis is to be considered only as a starting point for future, more elaborate investigations. Nonetheless, it is possible to tune the virtual drum such that it behaves in a similar way to the tom tom used in the experiment.

Given these limitations, the density of Mylar has been chosen according to values found in the literature. The missing values for the tensions have then been chosen in order to match the recorded sounds of the upper and lower membrane. This strategy, although very crude, can still provide interesting information about the physical behaviour of the system.

A.3.1 Physical model

The aim of the present analysis is to compare the different nonlinear models presented in Section 3.3, in the light of the data gathered with the experiment. Consider, therefore, a double membrane drum excited with a long raised cosine strike as described in Section 6.5. The physical model for this system has been discussed in detail in Section 6.3. Here, however, two different types of nonlinearities in the membrane equations are considered: the Berger model and the von Kármán model. The equations of motion in both cases are discussed in Section 3.3 for the isolated membranes, and in Sections 5.4.7 and 5.4.8 for the complete system.

The parameters used for the simulation can be found in Table A.4. Some of them, like the dimensions of the drum and the thickness of the membranes, have been measured from the real instrument. Others, like the density and Young's modulus for Mylar membranes, have been chosen according to values found in the literature [84]. Finally, the values for the tension have been deduced from the recordings. The simulations have been run at a sampling rate of 44.1 kHz, producing 1.5 s of output.

Parameter	**Simulation value**
sampling rate	44100 Hz
radius R	0.20 m
shell height	0.42 m
density of Mylar	1400 kg/m^3
Young's modulus of Mylar	3.5 GPa
batter membrane thickness	$H_b = 0.175$ mm
carry membrane thickness	$H_c = 0.19$ mm
batter membrane tension	$T_b = 1140$ N/m
carry membrane tension	$T_c = 954$ N/m

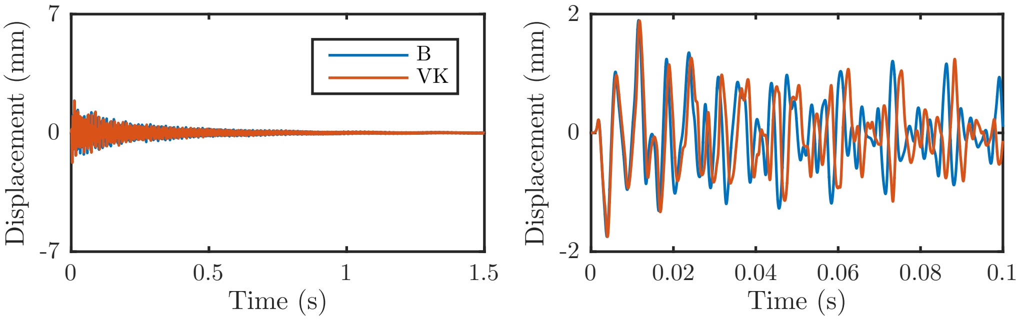
Table A.4: Parameters used in the numerical simulation of the virtual drums.

A.3.2 Simulation data

The data obtained from the numerical simulation using the Berger and von Kármán models are plotted in Figure A.6. Two excitation amplitudes are compared, which generate displacements that are comparable to the low and very high amplitude cases for the experimental data. As has been done for the experimental data in Figure A.3, a portion of 1.5 s of both signals has been plotted, together with a zoomed version of the first 0.1 s.

At low amplitude, the maximum displacement of the membrane in both cases is less than 2 mm. At high excitation amplitudes, the same striking force produces different maximum displacements for the two models—the first negative peak has a maximum displacement of 5.1 mm and 6.7 mm for the Berger and von Kármán case, respectively.

(a) Low amplitude



(b) Very high amplitude

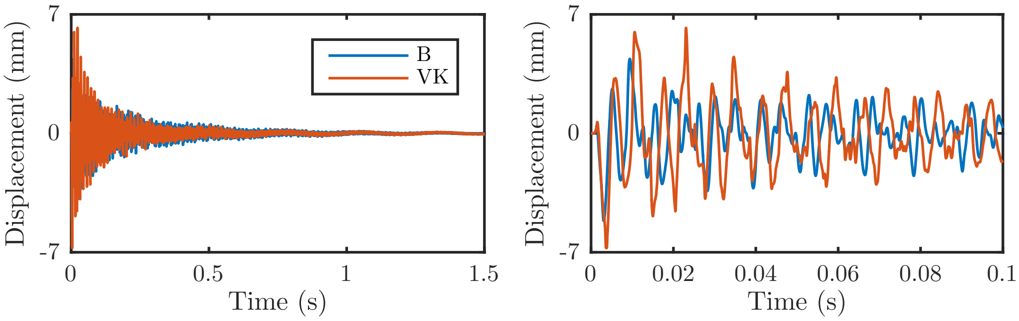


Figure A.6: Output signals obtained with a numerical simulation of a double membrane drum with Berger and von Kármán nonlinearities. The displacements of the upper drumhead at a point near the centre are plotted at striking amplitudes as indicated. Left: Plot of the entire signal. Right: Zoomed portion of 0.1 s of the signals. Notice the change of scale in the upper right plot. The Berger nonlinearity is plotted in blue, while the von Kármán nonlinearity is plotted in red.

A.3.3 Linear spectrum

In order to analyse the frequencies in the nonlinear case, it is first useful to plot a spectrum for the membrane under linear conditions. In this case, it is very simple to run a simulation with the same excitation parameters used before, but without the presence of the nonlinearities. The results of this analysis are plotted in Figure A.7. Although there are some clear differences with the experimental results of Figure A.4, the parameters chosen are probably not far from those of the real instrument. The various peaks or groups of peaks are labelled with letters from A to H.

A first peak, marked with A, appears at a frequency of 2.5 Hz. Then, a prominent peak appears at 77.3 Hz, which is labelled with B. Notice that, contrary to the experimental data, only a single peak appears here. This could be due to a not perfect tuning of the membrane in the experimental drum, leading to the mode splitting in the measurements. The peak C has a frequency of 157.6 Hz, which is close to the 157.2 Hz found in the experimental data. The next two peaks in the spectrum appear at frequencies of 231.1 Hz and 262.6 Hz, and are labelled with D. Notice that D_1 falls in the range of the doublet D which appears in the experimental data, while no corresponding peak to D_2 can be found in the experimental measurements. Seven other peaks appear below 450 Hz—a single peak E, a single peak F, a doublet G and a triplet labelled H—whose frequencies are indicated in Figure A.7.

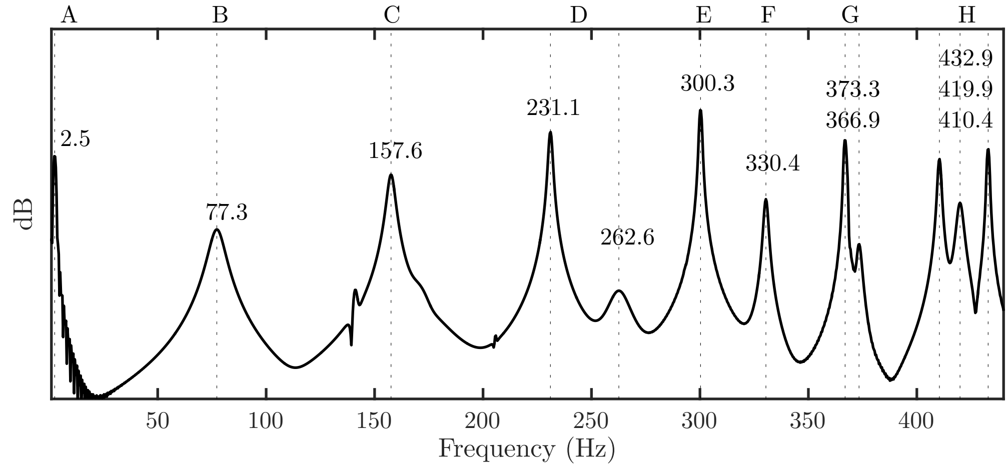


Figure A.7: Spectrum for the upper membrane under linear conditions. The peaks are marked with a vertical dotted line. The various peaks are marked with a label, and the values of their frequency are reported in Hz.

A.3.4 Data analysis

The signals in Figure A.6 have been analysed in a similar way to the experimental case. An initial portion of the signals at $t = 0.005$ s, with length 8192 steps (corresponding to a duration of approximately 0.19 s), has been compared in the frequency domain with a later portion of identical length at $t = 1$ s. Zero-padding and interpolation have been used in order to obtain the frequencies of the peaks.

Low amplitude

Figure A.8 shows the comparison between the initial spectra and those for a later portion of the signals at low amplitudes for both the Berger and the von Kármán models. For the von Kármán case, the deviations from the linear peaks range from 5.9 to 9.2 cents, with the only exception of the peak B that is shifted only by 1.6 cents. The Berger case, instead, produces pitch glides that are a third of a semitone wide for all peaks except for the peak B, which is shifted by 17.6 cents. Notice that no pitch glide was reported in the experimental measurements for the low amplitude case. Table A.5 summarises the results at medium excitation amplitudes.

Peak	**Linear freq. (Hz)**	**Berger freq. (Hz)**	**Cent dev.**	**von Kármán freq. (Hz)**	**Cent dev.**
B	77.3	78.1	17.6	77.3	1.6
C	157.6	160.6	33.2	158.1	5.9
D_1	231.1	235.7	34.7	231.9	6.6
E	300.3	305.9	32.4	301.5	7.1
F	330.4	336.1	29.7	331.9	7.9
G_1	366.9	373.1	28.7	368.4	6.8
H_1	410.4	417.7	30.4	412.6	9.2

Table A.5: Results from the analysis of the data at low excitation amplitude. The initial frequencies for the Berger and von Kármán cases are reported, together with the linear frequency of the peaks. Cent deviations from the linear frequencies are calculated before rounding the values to one decimal figure.

Very high amplitude

At very high excitation amplitudes, only four peaks are clearly distinguishable in both simulations. As is apparent, however, the two models exhibit quite different features. The peaks in the Berger case have a much higher frequency than the corresponding peaks in von Kármán. Frequency shifts of up to 136.4 cents can be observed for the peak C. This fact causes wider and faster pitch glides in the Berger case, that are not in line with the experimental results. For the real tom tom, the maximum variation observed was 66.6 cents for the peak E at very high amplitudes.

For the von Kármán model, the frequency shift for the peak B is only 7.4 cents, which is also considerably different from the experiment. The other peaks, however, show a behaviour that is compatible with that of the real drum. For the peak C, e.g., the deviation here is of 39.5 cents, while for the real drum values of 33.9 and 60.0 cents were observed at high and very high amplitudes, respectively.

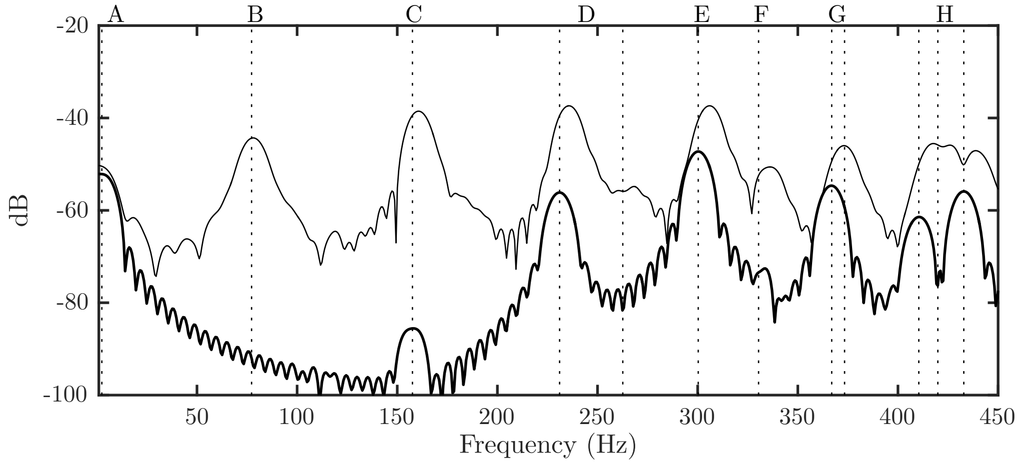
Peak	**Linear freq. (Hz)**	**Berger freq. (Hz)**	**Cent dev.**	**von Kármán freq. (Hz)**	**Cent dev.**
B	77.3	80.8	77.8	77.6	7.4
C	157.6	170.5	136.4	161.2	39.5
D_1	231.1	249.5	132.8	238.2	52.6
E	300.3	320.5	112.7	307.5	41.0

Table A.6: Results from the analysis of the data at very high excitation amplitude. The initial frequencies for the Berger and von Kármán cases are reported, together with the linear frequency of the peaks. Cent deviations from the linear frequencies are also calculated.

A.4 Final remarks

In this chapter, an experimental investigation on a tom tom performed in collaboration with M. Newton is compared with numerical simulations. The experiment highlights the presence of pitch glide effects in drum membranes at high striking amplitudes. This is not a novel

(a) Berger model



(b) von Kármán model

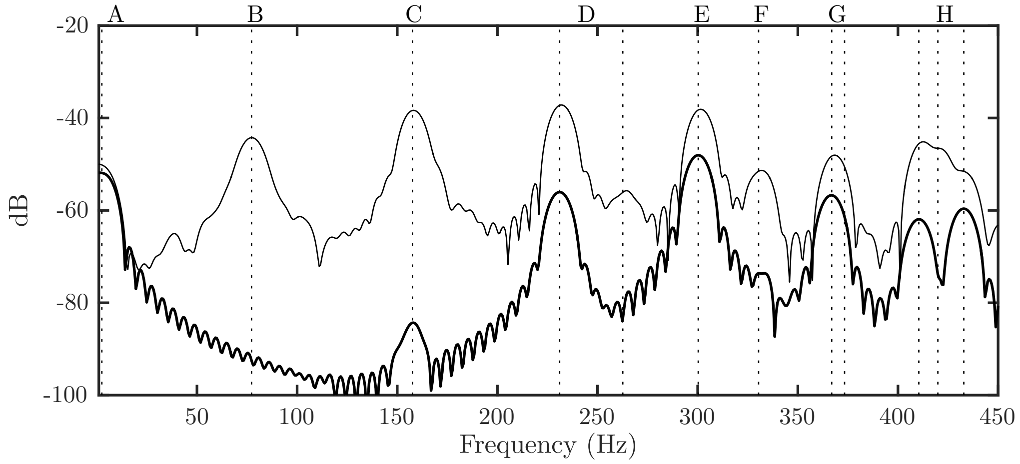
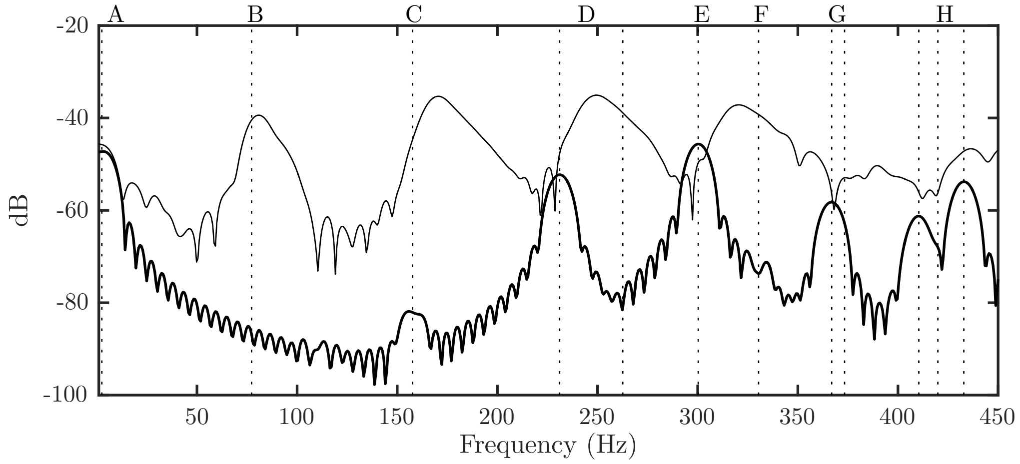


Figure A.8: Spectra for the simulated signals at low striking amplitudes. Top: Berger model. Bottom: von Kármán model. The spectrum for the initial portion of the signal at $t = 5$ ms is compared with a later portion at $t = 1$ s. The Fourier transform is computed over 8192 time steps, corresponding to approximately 0.19 s at the simulation sampling rate of 44.1 kHz.

observation, but it is the first time to the Author's knowledge that such an investigation is performed over several peaks in the spectrum. The experimental data show in fact that many peaks exhibit pitch glide effects, with relative variations of up to a semitone for the lowest modes.

Numerical simulations are also presented in this chapter with the aim of comparing the models at hand with the experiment. This analysis has several limitations that have already been discussed. The most important is the fact that the membrane tensions of the real drum could not be measured, and their values could not be used in the simulation. Some *ad hoc* approximations based on the sound recordings of the strikes have been used to reconstruct the original values, but this somehow limits the validity of the results. Another possible source of discrepancy between the experimental and simulated data is the position of the real drum during the experiment. The tom tom was positioned over a table, lying on a side, and held still in

(a) Berger model



(b) von Kármán model

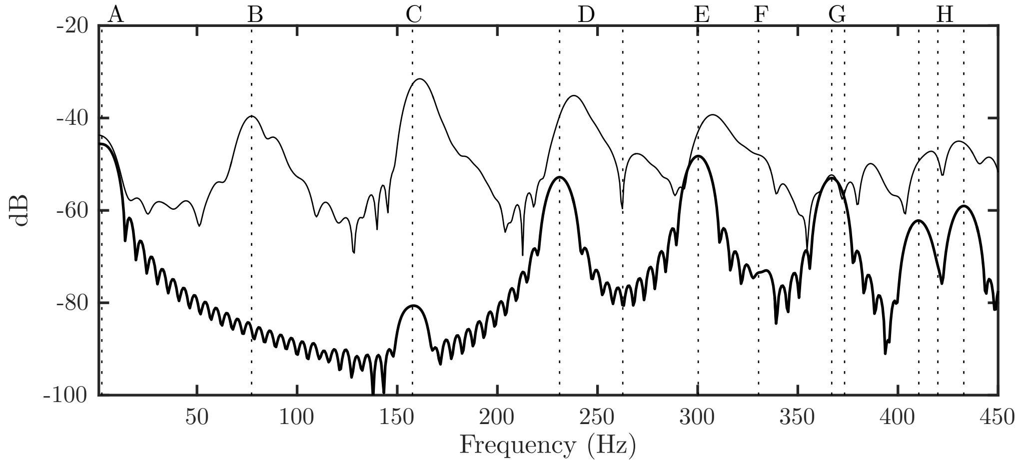


Figure A.9: Spectra for the simulated signals at very high striking amplitudes. Top: Berger model. Bottom: von Kármán model. The spectrum for the initial portion of the signal at $t = 5$ ms is compared with a later portion at $t = 1$ s. The Fourier transform is computed over 8192 time steps, corresponding to approximately 0.19 s at the simulation sampling rate of 44.1 kHz.

order to prevent it from moving. This might have inevitably interfered with the vibrations of the drum in isolation, which is what the numerical model simulates. Furthermore, no information about the striking parameters of the drumstick were recorded, which makes the comparison with a simulated raised cosine strike even more questionable. Finally, an additional source of error in the numerical model could be the choice of the dissipation coefficients, which should also be determined experimentally.

Despite these limitations, however, some useful insights can be drawn from the simulation regarding the difference between the Berger and the von Kármán model. Both models are able to produce pitch glide effects at high striking amplitudes. Under identical initial conditions and with the same physical parameters, however, the frequency shifts produced by the Berger model are wider and faster than those of von Kármán. In terms of sound synthesis applications, this determines outputs that are perceptually not accurate. In this respect, the von Kármán model offers a valid alternative. The pitch glides produced by this model are in fact more compatible with those from the experiment. These findings confirm those reported in [186]. It is clear, however, that further investigation is required in order to improve this analysis.

Appendix B

Finite precision effects in FDTD schemes

Numerical errors are an unavoidable effect of working in finite precision arithmetic. The study of these artefacts and their minimisation is the subject of numerical analysis [103, 138]. Some incidents involving numerical errors are recorded in the history annals.[1] One of the most dramatic is probably the Patriot Missile failure, which occured in 1991, when an American missile battery in Saudi Arabia failed to intercept an incoming Iraqi missile, causing the death of 28 American soldiers and the wounding of around 100 other people. The failure was caused by an inaccurate calculation of time due to the propagation of round-off errors.[2]

Even if numerical errors can hardly produce such dire consequences in the context of musical acoustics, nonetheless they can potentially harm the stability of a simulation. Their study seems therefore very important, but to the Author's knowledge, no previous analysis has been performed yet in this field. In this appendix, a brief account of some of the errors that can be encountered in a finite difference code is given, with an emphasis on simple systems such as the harmonic oscillator. This study can form a basis for the application of these methods to more complex models. A few basic ideas about floating point calculations and energy variations are discussed first.

B.1 Floating point calculations and energy variations

In this work, it has been stated several times that the discrete energy balances presented hold to machine precision of the computer, which has been denoted by ϵ. In this section, this notion will be clarified. First, however, a few basic facts about floating point computation are needed.

B.1.1 Floating point numbers

As is well known, modern computers use a binary floating point system to represent numbers. A fixed number of bits, depending on the precision adopted, are used to represent a signed mantissa and an exponent that specify the value of the number. A short primer on this topic is

---

[1] See, e.g., `http://ta.twi.tudelft.nl/users/vuik/wi211/disasters.html`

[2] For the full details, the interested reader is referred to `http://www.ima.umn.edu/~arnold/disasters/patriot.html`

given by Goldberg in [90]. What is important to note here is that, given the finite bits available, only a finite number of values can actually be represented in a computer.

The term *machine precision* indicates that quantity ϵ such that $1+\epsilon$ is the smallest number bigger than 1 that can be represented. Any real number $r \in \mathbb{R}$ such that $1 < r < 1+\epsilon$ cannot in general be exactly represented as a floating point number, and must then be rounded to either 1 or $1+\epsilon$ according to a rule. In any case, however, its floating point representation $fl(r)$ satisfies the following relation [103]

$$fl(r) = r(1+\delta), \qquad \text{with } |\delta| \leq \frac{1}{2}\epsilon. \tag{B.1}$$

In other words, the relative error between r and its representation $fl(r)$ is bounded by $1/2\epsilon$, or

$$\left|\frac{fl(r)-r}{r}\right| \leq \frac{1}{2}\epsilon. \tag{B.2}$$

The IEEE standard defines ϵ for single precision, $\epsilon = 2^{-23} \approx 1.19 \times 10^{-7}$, and double precision, $\epsilon = 2^{-52} \approx 2.22 \times 10^{-16}$ [103]. Double precision is considered in the remainder of this chapter.

Consider now a positive floating point number a, with $2^e \leq a < 2^{e+1}$ for some integer exponent e. The notation $\lfloor a \rfloor_2 \equiv 2^e$ will be used to indicate the nearest power of two to a, rounding towards zero. With a slight abuse of notation, it is possible to introduce for any floating point number a the function $\epsilon(a)$ that returns the spacing between a and the smallest floating point number bigger than a, that is $a+\epsilon(a)$. Of course, $\epsilon(1) = \epsilon$, the machine precision. The gap $\epsilon(a)$ is given by the formula [140]

$$\epsilon(a) = \lfloor a \rfloor_2 \epsilon = 2^e \epsilon. \tag{B.3}$$

B.1.2 Discrete energy variations

In this work, energy techniques have been applied to the numerical schemes in order to evaluate their behaviour. In practice, a discrete quantity is defined and calculated, which plays the role of the total energy of the system. One then wishes to monitor the variations of this numerical energy with respect to the initial value, in order to ascertain that no artificial energy is added into the system, nor that any unexpected leakage is present.

Consider a numerical scheme with total energy $\mathfrak{h}^n$ at time step n, and let the energy at $n = 0$ be $\mathfrak{h}^0 > 0$. The normalised energy variation

$$\mathfrak{e} = \frac{\mathfrak{h}^n - \mathfrak{h}^0}{\mathfrak{h}^0} \tag{B.4}$$

is often used in the literature [19] to rescale the energy differences down to machine precision. This operation, however, does not lead to the results that one wishes. The difference in the numerator, in fact, is generally proportional to $\epsilon(\mathfrak{h}^0) \leq \epsilon \cdot \mathfrak{h}^0$. Dividing by $\mathfrak{h}^0$ in the previous definition of $\mathfrak{e}$ causes the machine precision quantisation to be "shrunk". The above definition for the normalised energy variation can be modified to the following:

$$\mathfrak{e} = \frac{\mathfrak{h}^n - \mathfrak{h}^0}{\lfloor \mathfrak{h}^0 \rfloor_2}, \tag{B.5}$$

where the notation $\lfloor\cdot\rfloor_2$ has been defined in the previous section, in order to obtain the correct quantisation. This is the definition used throughout this work. Figure B.1 shows this phenomenon for a typical simulation. As is apparent, calculating the difference with the first expression causes the quantisation to be rescaled, whereas with (B.5) the machine precision jumps are clearly visible.

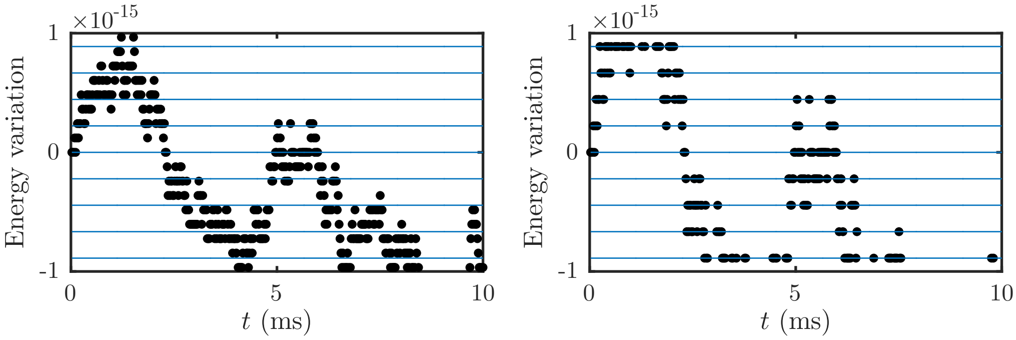


Figure B.1: Difference between the energy variations calculated using the definitions (B.4) and (B.5) for a typical simulation. Left: graph of $\mathfrak{e} = \frac{\mathfrak{h}^n - \mathfrak{h}^0}{\mathfrak{h}^0}$. Right: graph of $\mathfrak{e} = \frac{\mathfrak{h}^n - \mathfrak{h}^0}{\lfloor \mathfrak{h}^0 \rfloor_2}$. As is apparent, the first expression fails to reveal the machine precision quantisation in the energy variation. Horizontal lines represent integer multiples of $\epsilon = 2^{-52}$.

Alternative definition

An alternative definition of normalised energy variations could be the following:

$$\mathfrak{e}^* = \left(\frac{\mathfrak{h}^n}{\mathfrak{h}^0} \right) - 1, \tag{B.6}$$

where the parentheses indicate the operation order. A similar definition has been used, e.g., in [70]. Although mathematically equivalent to (B.4), numerically this formula produces quantised jumps in multiples of ϵ, when $\mathfrak{e}^* > 0$. Jumps that are multiples of $\epsilon/2$ appear for negative values of $\mathfrak{e}^*$, i.e., when $\mathfrak{h}^n < \mathfrak{h}^0$.

Figure B.2 compares $\mathfrak{e}^*$ with the definition of $\mathfrak{e}$ given in (B.5). The results given by the two definitions appear to be different, although an explanation for this behaviour is not readily available.

B.2 Simple harmonic oscillator

Consider now the simple harmonic oscillator (SHO) with mass M and stiffness constant K defined by the ordinary differential equation

$$M \frac{\mathrm{d}^2 u}{\mathrm{d}t^2} = -Ku, \tag{B.7}$$

where $u(t)$ is the position of the mass at time $t \in \mathbb{R}^+$. This equation must be accompanied by two initial conditions, namely $u_0 = u(0)$ and $v_0 = \mathrm{d}u/\,\mathrm{d}t|_{t=0}$, the initial position and velocity of the mass. As is well known, the most general solution for (B.7) when $\omega_0 > 0$ can be written

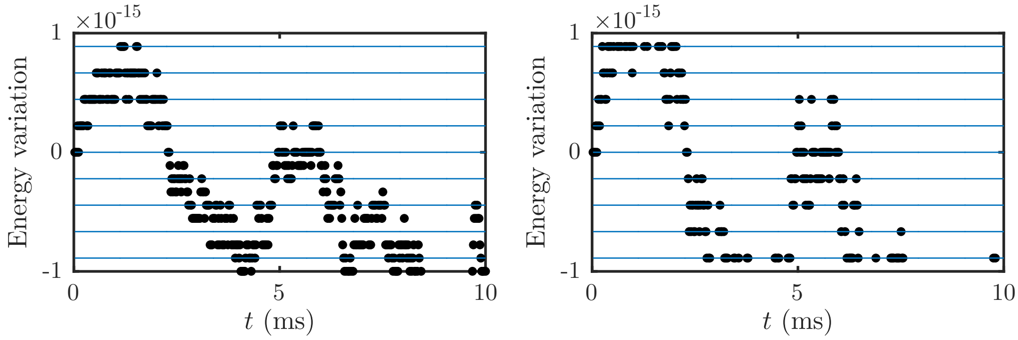


Figure B.2: Difference between the energy variations calculated using the definitions (B.6) and (B.5) for the same simulation as Figure B.1. Left: graph of $\mathfrak{e}^* = \left(\frac{\mathfrak{h}^n}{\mathfrak{h}^0}\right) - 1$. Right: graph of $\mathfrak{e} = \frac{\mathfrak{h}^n - \mathfrak{h}^0}{\lfloor \mathfrak{h}^0 \rfloor_2}$. The first expression gives energy quantisation jumps on the order of machine ϵ when $\mathfrak{e}^* > 0$. Half-ϵ steps, however, appear when $\mathfrak{e}^*$ is negative. Horizontal lines represent integer multiples of $\epsilon = 2^{-52}$.

as

$$u(t) = u_0 \sin(\omega_0 t) + \frac{v_0}{\omega_0} \cos(\omega_0 t), \tag{B.8}$$

where $\omega_0 = \sqrt{K/M}$ is the natural frequency of the system. Using basic algebraic manipulations, the solution $u(t)$ can be rewritten as

$$u(t) = A \cos(\omega_0 t - \phi), \tag{B.9}$$

with $A = \sqrt{u_0^2 + \frac{v_0^2}{\omega_0^2}}$ and $\tan(\phi) = -\frac{v_0}{\omega_0 u_0}$.

A perhaps more fundamental way to write the equation is in terms of a first order system. Introducing the velocity $v(t) = \mathrm{d}u/\,\mathrm{d}t$, it is possible to write (B.7) as

$$M \frac{\mathrm{d}v}{\mathrm{d}t} = -Ku \tag{B.10a}$$

$$\frac{\mathrm{d}u}{\mathrm{d}t} = v. \tag{B.10b}$$

It is easy to be convinced that (B.7) and (B.10) represent indeed the same physical system. In both cases, the energy for the oscillator can be obtained by multiplying the equation by the velocity v. The energy balance that is obtained is

$$\frac{\mathrm{d}\mathfrak{H}_{\text{sho}}}{\mathrm{d}t} = 0, \qquad \mathfrak{H}_{\text{sho}} = \frac{M}{2}v^2 + \frac{K}{2}u^2. \tag{B.11}$$

B.2.1 Numerical simulation

The simulation of an SHO can provide useful insights on the behaviour of numerical schemes, and can serve as a guide in the implementation of more complicated models, as those presented in this work.

The simplest discretisation of (B.7) can be obtained by introducing a time step k and the variable $u^n \approx u(nk)$ approximating the position of the mass at integer multiples of the time step k. The simplest finite difference scheme can be written as

$$M\delta_{tt}u^n = -Ku^n, \tag{B.12}$$

where δ_{tt} is the second difference operator (2.97). The update is given by

$$u^{n+1} = (2 - k^2\omega_0^2)u^n - u^{n-1}. \tag{B.13}$$

A detailed analysis of this scheme is given in [19], so only few details of interest will be treated here. An energy balance for (B.12) can be obtained by multiplying the equation by $\delta_{t\cdot}u$, thus arriving at the following balance

$$\delta_{t-}\mathfrak{h}_{\text{sho}}^{n+1/2} = 0, \qquad \mathfrak{h}_{\text{sho}}^{n+1/2} = \left(\frac{M}{2}(\delta_{t+}u^n)^2 + \frac{K}{2}u^n\,u^{n+1}\right). \tag{B.14}$$

The first order formulation has an interleaved character similar to Yee's FDTD algorithm [208, 172]. It is formally equivalent to the previous second order equation, and can be written as

$$M\delta_{t-}v^{n+1/2} = -Ku^n \tag{B.15a}$$
$$\delta_{t-}u^{n+1} = v^{n+1/2}, \tag{B.15b}$$

where $v^{n+1/2}$ is the discrete approximation to $v(t)$. The update for this scheme is

$$v^{n+1/2} = -\omega_0^2 k u^n + v^{n-1/2} \tag{B.16a}$$
$$u^{n+1} = kv^{n+1/2} + u^n. \tag{B.16b}$$

Notice here that velocity and position are interleaved, meaning that they are calculated at integer and half integer time steps, respectively. In this way, the resulting scheme is explicit. The energy expression (B.14) obtained from the second order scheme remains valid for (B.15) as well, provided that $\delta_{t+}u^n$ be substituted by $v^{n+1/2}$.

Through the determination of non-negativity conditions for $\mathfrak{h}_{\text{sho}}^{n+1/2}$, it is easy to show that both schemes are stable when

$$\frac{\omega_0}{2} \leq F_s, \tag{B.17}$$

where F_s is the sampling rate of the scheme. The discretisation causes the scheme to oscillate at a frequency f_{num} different from f_0, and in particular [19]

$$f_{\text{num}} = \frac{1}{\pi k}\sin^{-1}(k\pi f_0). \tag{B.18}$$

B.2.2 Oscillations in the numerical energy

One interesting question that arises regarding the simple harmonic oscillator presented above is if the first order scheme (B.15) gives the same numerical results as the second order one (B.12). As a matter of fact, their mathematical equivalence in infinite precision arithmetic does not prevent numerical errors from appearing in floating point calculations. Similar studies have been performed by Botts [35] for the wave equation, with the conclusion that the two schemes are numerically equivalent, although no comparison is presented on the actual numbers produced by the two schemes. Figure B.3 shows the outputs and the energy calculations for an oscillator with $M = 0.1$ kg and $f_0 = \omega_0/2\pi = 2$ Hz, with a sampling rate of 44.1 kHz. At the plot scale, the two schemes are indistinguishable, with potential and kinetic energy summing up to a constant total energy, and with the mass vibrating at 2 Hz as expected (the effect of dispersion is negligible at such low f_0, see (B.18).), although different numbers are produced for the outputs. An analysis of the energy variations in the two cases, in fact, is revealing, see Figure B.4. While the normalised energy variations calculated with the first order implementation (B.15) present the usual machine precision quantisation, the second order implementation (B.12) exhibits oscillations well above machine ϵ. These appear to be at twice the numerical frequency f_{num} (B.18) of the scheme. A slight change in the implementation of (B.13) reveals that rounding errors are (at least partially) to blame for this behaviour. The term $k^2\omega_0^2$ which appears in the update can be be considerably smaller than 2, given the presence of the factor k^2. Thus, when the quantity $2-k^2\omega_0^2$ is computed, several least significant digits can be lost due to floating point representation, thus degrading the overall accuracy of the scheme. A simple alternative approach is to perform the update in this form:

$$u^{n+1} = 2u^n - k^2\omega_0^2 u^n - u^{n-1}, \tag{B.19}$$

where the multiplications are performed before the subtraction. This has the effect of breaking the oscillations, as can be seen in Figure B.4c. The resulting energy presents variations which are more than an order of magnitude smaller than implementation (B.12). These are, however, still three orders of magnitude bigger than those of the system implementation (B.15), thus hinting at the fact that the first order implementation might behave better than the second order one.

The problem of accumulation of errors in numerical integration is an old one [39], and is particularly relevant when the long-term behaviour of lossless systems is of interest (for example, in astronomical simulation). Recent works by Hairer *et al.* [94, 62] have concentrated on the stability of numerical integration, and some recipes have been proposed to reduce the harm of round-off errors in the simulations [197, 63].

B.3 Lossy oscillator

The equation of motion for a lossy oscillator can be written as

$$M\,\frac{\mathrm{d}^2 u}{\mathrm{d}t^2} = -Ku - 2\sigma_0 M\,\frac{\mathrm{d}u}{\mathrm{d}t} \tag{B.20}$$

where $\sigma_0 \geq 0$ is a loss coefficient. Obviously, when $\sigma_0 = 0$ the equation reduces to the lossless case (B.7). This equation can be equivalently expressed in a first order formulation as

$$M\,\frac{\mathrm{d}v}{\mathrm{d}t} = -Ku - 2\sigma_0 M v \tag{B.21a}$$

$$\frac{\mathrm{d}u}{\mathrm{d}t} = v. \tag{B.21b}$$

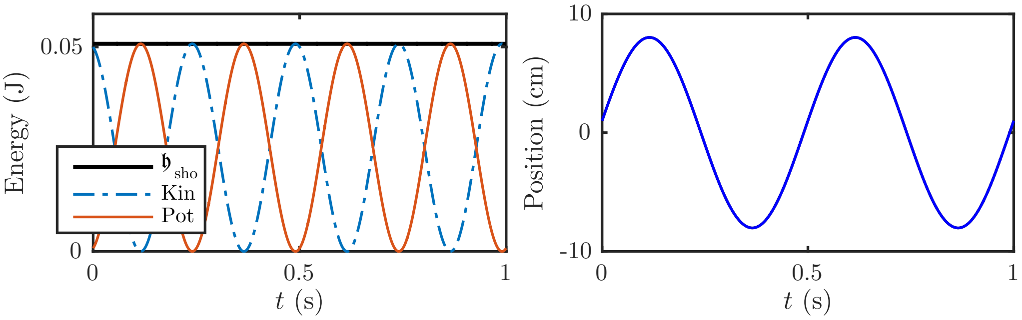


Figure B.3: Comparison between the first order system and the second order equation implementation of the SHO scheme. Left: total energy of the system, with kinetic and potential energy plots. Right: comparison between the outputs (black: first order system, blue: second order equation). The results look identical at this scale. The oscillator has a mass $M = 0.1$ kg, $f_0 = 2$ Hz, and has been initialised with $x_0 = 0.01$ m and $v_0 = 1$ m/s. The sample rate is 44.1 kHz.

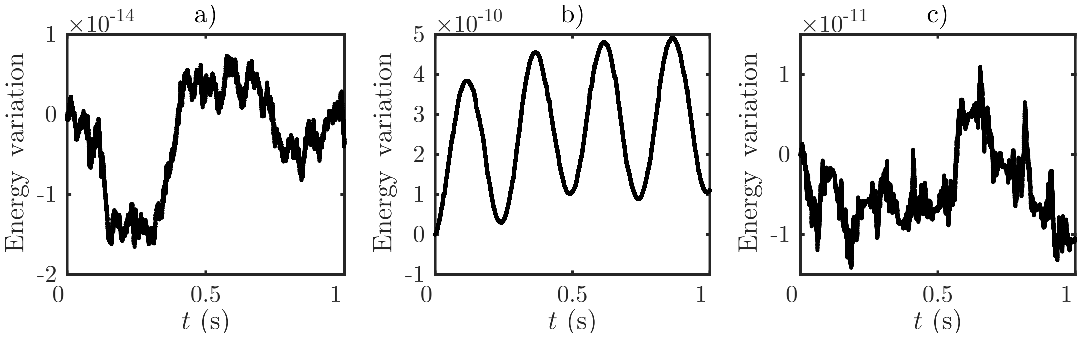


Figure B.4: Comparison between the normalised energy variations obtained with the first order scheme (B.15) and with the second order equations (B.12). While the first order system presents energy quantisation errors (Figure a), the second order equation implemented as (B.12) exhibits oscillations at twice the numerical frequency of the scheme (Figure b). Notice the different scale in the two plots. With a slight change in the implementation, using (B.19), it is possible to reduce the energy variations of the scheme (Figure c). The parameters for the system and for the simulation are as in Figure B.3.

This time, a dissipation term appears in the energy balance for the system, which becomes:

$$\frac{\mathrm{d}\mathfrak{H}_{\text{sho}}}{\mathrm{d}t} = -\mathfrak{Q}, \qquad \mathfrak{Q} = 2\sigma M \left(\frac{\mathrm{d}u}{\mathrm{d}t}\right)^2 \geq 0. \tag{B.22}$$

As expected, the total energy of the system diminishes in time. Integrating this equation between $t = 0$ and $t = T$ gives the following balance:

$$\mathfrak{H}_{\text{sho}}(T) + \int_{t=0}^{T} \mathfrak{Q}(t)\,\mathrm{d}t = \mathfrak{H}_{\text{sho}}(0). \tag{B.23}$$

In other words, adding back to the energy at time T what has been dissipated by loss, one can recover the initial energy of the system. This can be exploited in a simulation to ensure the conservation of the total energy even in the lossy case.

B.3.1 Numerical simulation

A finite difference scheme for (B.20) can be easily written as

$$M\delta_{tt}u^n = -Ku^n - 2\sigma_0 M\delta_{t\cdot}u^n. \tag{B.24}$$

Taking into account (B.19) and the previous discussion on numerical errors, the update becomes

$$(1+\sigma_0 k)u^{n+1} = 2u^n - k^2\omega_0^2 u^n - (1-\sigma_0 k)u^{n-1}. \tag{B.25}$$

The energy balance in this case is

$$\delta_{t+}\mathfrak{h}_{\text{sho}}^{n+1/2} = -\mathfrak{q}^n, \qquad \mathfrak{q}^n = 2\sigma_0 M(\delta_{t\cdot}u^n)^2, \tag{B.26}$$

where $\mathfrak{h}_{\text{sho}}^{n+1/2}$ is as in (B.14). At every time step, then, the energy variation should match the losses of the right hand side. A discrete analogue of (B.22) at time step $n = N$ can be derived as

$$\mathfrak{h}_{\text{sho}}^{N+1/2} + \sum_{j=1}^{N} k\mathfrak{q}^j = \mathfrak{h}_{\text{sho}}^{1/2}, \tag{B.27}$$

where $\mathfrak{h}_{\text{sho}}^{1/2}$ is the initial energy of the system, calculated using the values for the initialisation. The quantity on the left hand side will be referred to as the accumulated energy of the oscillator. Finally, the first order system is

$$M\delta_{t-}v^{n+1/2} = -Ku^n - 2\sigma_0 M\mu_{t-}v^{n+1/2} \tag{B.28a}$$

$$\delta_{t-}u^{n+1} = v^{n+1/2}, \tag{B.28b}$$

with update given by

$$(1+\sigma_0 k)v^{n+1/2} = -\omega_0^2 k u^n + (1-\sigma_0 k)v^{n-1/2} \tag{B.29a}$$

$$u^{n+1} = kv^{n+1/2} + u^n. \tag{B.29b}$$

As in the lossless case, different implementations of (B.25) and (B.29) can lead to different results. Consider, in particular, a simulation with $M = 0.1$ kg, $f_0 = 1000$ Hz and $\sigma_0 = 0.1$ s^{-1} at a sampling rate of 44.1 kHz. Figure B.5 shows the normalised variations for the accumulated energy over a 10 s output. In both cases, the accumulated energy calculated with (B.27) is not conserved as one would expect. This is due to rounding errors related to the magnitude of the parameter σ_0. Simulations with a lower value for the loss coefficient, in fact, tend to give the worst results in terms of energy conservation. On the other hand, a short simulation with a large damping coefficient (see, e.g., [57]) will not be able to highlight this behaviour. Note that the second order update (Figure B.5b) presents the oscillations discussed above, when the new update form is not used.

Regardless of the choice for the update, the origin of these numerical errors lies in the factors $(1 + \sigma_0 k)$ and $(1 - \sigma_0 k)$ appearing in the recursion. The difference on the right hand side, $(1-\sigma_0 k)$, presents the same difficulties as the lossless case. Here, $\sigma_0 k$ can be several orders of magnitude smaller than 1, thus causing a loss of accuracy. This can be solved by performing the multiplication before the difference, see (B.19). The term on the left hand side, $(1+\sigma_0 k)$, presents similar challenges. Simply dividing the right hand side of the update by this quantity gives rise to losses of accuracy for the reasons outlined before. One possible way to circumvent this problem is by using iterative methods. The alternative update for (B.24), e.g., could be performed as follows:

```
$temp \leftarrow 2x^n - k^2\omega_0^2 x^n - x^{n-1} + \sigma_0 k x^{n-1}$        ▷ Calculate the right hand side
$u^{n+1} \leftarrow temp$                                                   ▷ Initialise $u^{n+1}$
for $i = 1 \to N_{\text{iter}}$ do
    $u^{n+1} = temp - \sigma_0 k u^{n+1}$                                   ▷ Iterative method for $u^{n+1}$
end for
```

Here, N_{iter} is the number of iterations required to achieve machine accuracy, typically on the order of 4-5 for values of σ_0 that are usually found in musical acoustics simulation. A similar approach can be applied to the first order system (B.28). When the update is performed in this manner, the first order implementation shows conservation of energy to machine accuracy, even for a long simulation (see Figure B.6a). Some improvements can be noticed also in the second order update (Figure B.6b), although the energy variations are far from machine accuracy. Once again, this could hint at some deeper issues involving second order schemes that have still to be investigated.

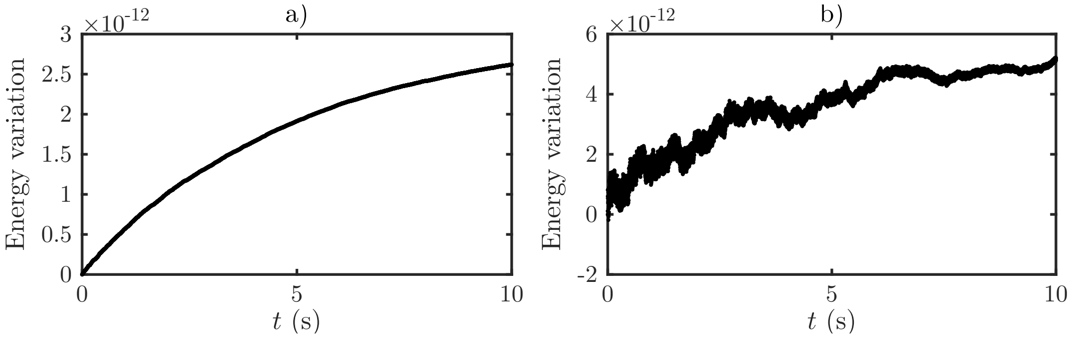


Figure B.5: Normalised variations in the accumulated energy (B.27) using the first order update (B.29) (Figure a) and using the second order update (B.25) (Figure b). In both cases the accumulated energy is not conserved to machine accuracy, as one would expect. Notice the oscillations appearing in Figure b, which have been discussed in the previous section.

B.4 Free particle

Consider now the apparently banal case of a free particle of mass M travelling on a straight line with constant velocity v_0, with position $u(t)$ at time $t \in \mathbb{R}^+$. The equations of motion in this case, both as a second order equation and a first order system, are

$$M\frac{\mathrm{d}^2 u}{\mathrm{d}t^2} = 0 \qquad \text{(B.30a)} \qquad\qquad \begin{cases} M\dfrac{\mathrm{d}v}{\mathrm{d}t} = 0 \\ \dfrac{\mathrm{d}u}{\mathrm{d}t} = v, \end{cases} \qquad \text{(B.30b)}$$

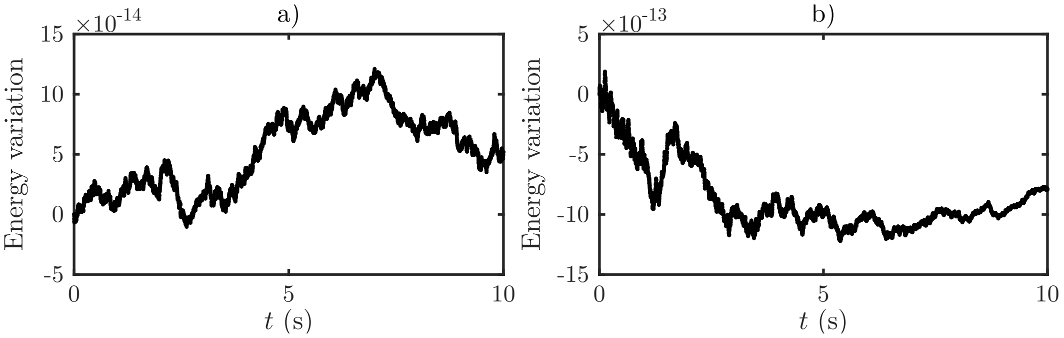


Figure B.6: Normalised variations in the accumulated energy (B.27) using the improved update for the first order system (Figure a) and for the second order equation (Figure b). Energy variations in the first case are on the order of machine accuracy, while in the second case they are not.

which must be complemented with the initial conditions $v(0) = v_0$ and $u(0) = u_0$. The energy for the system is given by the constant kinetic energy

$$\mathfrak{H}_{\text{free}} = \frac{M}{2}v^2 = \frac{M}{2}v_0^2. \tag{B.31}$$

Two possible finite difference discretisations for the equations of motion are

$$M\delta_{tt}u^n = 0 \qquad \text{(B.32a)} \qquad \begin{cases} M\delta_{t+}v^{n-1/2} = 0 \\ \delta_{t-}u^{n+1} = v^{n+1/2}, \end{cases} \qquad \text{(B.32b)}$$

where staggered variables in time have been chosen in the first order system in order to keep the scheme explicit. Supplying u^0 and u^1, or equivalently u^0 and $v^{1/2}$, allows both schemes to be updated recursively. The discrete energy in both cases is

$$\mathfrak{h}_{\text{free}}^{n+1/2} = \frac{M}{2}\left(v^{n+1/2}\right)^2. \tag{B.33}$$

Notice that, given the non-negativity of $\mathfrak{h}_{\text{free}}$, no condition on k is required for the schemes to be stable. Once again, the two schemes behave differently from a numerical point of view. The energy variables in the two cases for typical simulation parameters (indicated in the caption) are plotted in Figure B.7. The energy for the first order scheme is indeed constant as expected. This is hardly surprising, given that the velocity remains strictly the same at every iteration. On the other hand, the energy evolution for the second order implementation presents jumps with increasing amplitude. These are similar to the energy gaps reported in the mallet collision case (Section 4.1), after the impact with the barrier. Energy variations in the second order scheme are easy to explain. Here, the velocity $v^{n+1/2}$ is calculated at every time step from the values u^{n+1} and u^n of the position. Starting from a negative position $-u_0 < 0$, the mass eventually reaches and passes the value $u_0 > 0$. At this point, every time the position u^{n+1} crosses a power of two, one bit is lost in the representation with respect to the value u^n. Thus, round off errors appear in the calculation of the velocity and propagate in the energy expression, causing it to change.

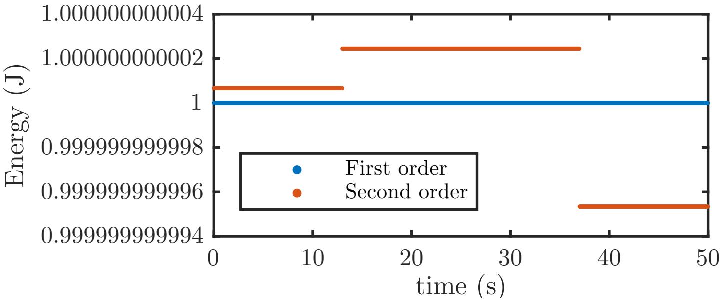


Figure B.7: Energy for a free particle simulation with finite difference implementation (B.32b) (blue) and (B.32a) (red), respectively. Simulation parameters are $M = 0.5$ kg, $v_0 = 2$ m/s, $u_0 = -10$ m, and the sampling rate is 1 kHz.

B.5 3D wave equation

Round off errors are present also in the implementation of the 3D wave equation scheme (5.27). Consider, in fact, the simulation presented in Section 5.3. When the scheme is implemented as in (5.28), where the matrix $\mathbf{B} = 2\mathbb{1} + c_a^2 k^2 \mathbf{D}_\Delta^{(3)}$ is computed first and then applied to the vector $\mathbf{\Psi}^n$, a linear drift can be seen in the energy pattern for both Dirichlet and Neumann conditions applied at the boundary. This behaviour can be seen in Figure B.8, where the same parameters of Section 5.3 are used.

A simple way to correct this problem is to perform the calculation as follows

$$\mathbf{\Psi}^{n+1} = (2\mathbf{\Psi}^n) + \left(c_a^2 k^2 \mathbf{D}_\Delta^{(3)} \mathbf{\Psi}^n\right) - \mathbf{\Psi}^{n-1}, \tag{B.34}$$

where the parentheses indicate the correct order of the operations. This produces the plots in Figure 5.2. Particular attention must be given to the factor $c_a^2 k^2$, that is combined with the factor $1/h_a^2$ present in the Laplacian matrix. At the stability limit (5.33), one has

$$\frac{c_a^2 k^2}{h_a^2} = \frac{1}{3}. \tag{B.35}$$

For the best results in implementation, it is worth checking that the term on the left hand side be exactly 1/3. One can treat $\mathbf{D}_\Delta^{(3)}$ as a matrix of integer values, performing the matrix-vector multiplication first. The result can then be divided by 3, thus eliminating possible sources of errors.

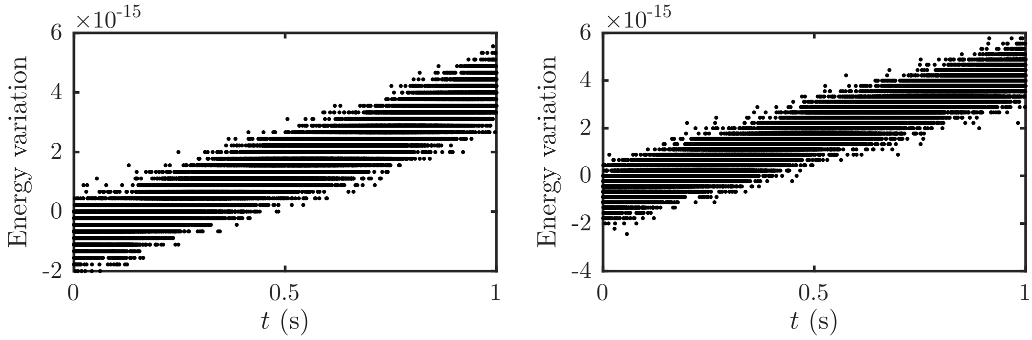


Figure B.8: Energy variations for a naive implementation of the 3D wave equation scheme (5.28) with Dirichlet (left) and Neumann conditions (right) applied at the boundaries. A linear drift in the energy pattern is apparent in both cases, and it is due to the presence of round off errors in the scheme.

Appendix C

Iterative methods for linear systems

In many fields of science, an often encountered challenge is the solution of linear systems of the form

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{C.1}$$

where $\mathbf{b}$ is a known vector of length N and $\mathbf{A}$ is a square invertible matrix with dimensions $N \times N$. A common approach to the numerical solution of this set of equations is the use of direct methods. One example is Gauss-Jordan elimination, which can be used to achieve an exact solution in a finite number of steps, typically on the order of $N^3/3$ [167]. One major drawback with this technique is that the calculation of the j-th entry x_j of the unknown vector $\mathbf{x}$ requires the knowledge of the values $x_1, \ldots, x_{j-1}$, and is therefore an intrinsically serial operation.

An alternative approach is the use of iterative methods [167, 157]. Starting from an initial guess $\mathbf{x}_0$ to the exact solution $\mathbf{x}$, these techniques produce a better approximation $\mathbf{x}_i$ at every i-th iteration, and they stop once a desired tolerance is achieved. In this case, the convergence properties can vary significantly between different methods, but a considerable speed-up can often be achieved over direct methods.

C.1 Gauss-Seidel method

One of the simplest iterative methods is Gauss-Seidel. Starting from an initial guess $\mathbf{x}_0$ and given a splitting of $\mathbf{A}$ into two matrices such that $\mathbf{A} = \mathbf{A}_1 - \mathbf{A}_2$, the solution of the linear system can be obtained recursively using the following equation:

$$\mathbf{A}_1\mathbf{x}_{i+1} = \mathbf{A}_2\mathbf{x}_i + \mathbf{b}. \tag{C.2}$$

Denoting with $\overline{\mathbf{x}}$ the theoretical exact solution to the linear system, it is easy to show that the error $\mathbf{e}_j = \overline{\mathbf{x}} - \mathbf{x}_j$ is governed by the relation

$$\mathbf{A}_1\mathbf{e}_{j+1} = \mathbf{A}_2\mathbf{e}_j. \tag{C.3}$$

Therefore, Gauss-Seidel method converges to the exact solution when every eigenvalue λ of the matrix $\mathbf{A}_1^{-1}\mathbf{A}_2$ satisfies $|\lambda| < 1$ [167]. Furthermore, the rate of convergence of the algorithm depends on the spectral radius $r(\mathbf{A}_1^{-1}\mathbf{A}_2)$ of the matrix $\mathbf{A}_1^{-1}\mathbf{A}_2$, defined as

$$r(\mathbf{A}_1^{-1}\mathbf{A}_2) = \max|\lambda_i|, \qquad \text{with } i = 1, \ldots, N. \tag{C.4}$$

The quantity $-\log_{10}(r)$, in particular, is defined as the rate of convergence of the algorithm, that is the increase in the number of correct decimal places in the solution per iteration [67]. The smaller is r, the faster is the convergence of the linear system solution.

C.2 Krylov subspace methods

Alternative approaches to the solution of linear systems are methods based on Krylov subspaces [67, 157]. Many different solution methods exist which are based on projection procedures that, starting from the residual $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, with $\mathbf{x}_0$ an initial guess for the solution, create a subspace $\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$ defined as

$$\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0) = \mathit{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \ldots \mathbf{A}^{m-1}\mathbf{r}_0\}, \tag{C.5}$$

where m is the dimension of the matrix, and $\mathit{span}\{\cdot\}$ indicates the space spanned by the vectors. The linear system solution proceeds by repeating simple matrix-vector operations that do not involve the inversion of $\mathbf{A}$, until the residual $\mathbf{r}_j$ at the j-th iteration is made small to the desired accuracy. The choice of the norm used to estimate the magnitude of $\mathbf{r}_j$ and the properties of the matrix $\mathbf{A}$ determine which of the various Krylov subspace methods can be adopted (a decision tree for choosing the appropriate technique based on the problem at hand is given by Demmel in [67, p. 321]). When $\mathbf{A}$ is symmetric and positive definite, the conjugate gradient (CG) method is often used. For non-symmetric matrices, methods such as GMRES, BiCGSTAB, CGS and QMR are generally used. The algorithms for these methods can be found in the classic text by Saad [157].

The typical problem with these methods is that the convergence of the algorithm is generally not guaranteed. This, in fact, depends on the condition number of the matrix $\mathbf{A}$, and is thus very sensitive to the particular problem at hand. One way to overcome this difficulty is the use of preconditioners. *Preconditioning* attempts to increase the stability of the algorithm by replacing the original problem $\mathbf{A}\mathbf{x} = \mathbf{b}$ with the equivalent $\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b}$. Here, $\mathbf{M}^{-1}$ is a matrix that approximates $\mathbf{A}$ and has the following properties [67]:

- $\mathbf{M}$ is symmetric and positive definite;
- $\mathbf{M}^{-1}\mathbf{A}$ is well conditioned;
- $\mathbf{M}\mathbf{x} = \mathbf{b}$ is easier to solve than the original problem.

Preconditioning of iterative methods is now a very active field of research (see the work of Benzi [12] for a recent survey). Several possibilities are available for the choice of $\mathbf{M}^{-1}$, that range from considering only the diagonal elements of $\mathbf{A}$, to sparse approximate inverses such as incomplete Cholesky or LU factorisations [12]. When a factorisation such as $\mathbf{M} = \mathbf{M}_1\mathbf{M}_2$ is available, the system to solve can be written as

$$\mathbf{M}_1^{-1}\mathbf{A}\mathbf{M}_2^{-1}\mathbf{u} = \mathbf{b}, \qquad \mathbf{x} = \mathbf{M}_2^{-1}\mathbf{u}, \tag{C.6}$$

for an auxiliary variable **u**.

Bibliography

[1] J.-M. Adrien and X. Rodet. Physical models of instruments: A modular approach, application to strings. In *Proceedings of the International Computer Music Conference*, pages 85–89, Vancouver, Canada, 1985.

[2] M.-L. Aird. *Musical instrument modelling using digital waveguides.* PhD thesis, University of Bath, UK, 2002.

[3] H. Akay. Dynamic large deflection analysis of plates using mixed finite elements. *Computers & Structures*, 11(1):1–11, 1980.

[4] M. Amabili. *Nonlinear vibrations and stability of shells and plates.* Cambridge University Press, Cambridge, UK, 2008.

[5] K. Arcas. Simulation numérique d'un réverbérateura plaque. Master's thesis, Université Pierre et Marie Curie, Paris, France, 2008.

[6] F. Avanzini and R. Marogna. A modular physically based approach to the sound synthesis of membrane percussion instruments. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(4):891–902, 2010.

[7] F. Avanzini, R. Marogna, and B. Bank. Efficient synthesis of tension modulation in strings and membranes based on energy estimation. *The Journal of the Acoustical Society of America*, 131(1):897–906, 2012.

[8] F. Avanzini and D. Rocchesso. Modeling collision sounds: Non-linear contact force. In *Proc. COST-G6 Conf. Digital Audio Effects (DAFx-01)*, Limerick, Ireland, 2001.

[9] I. Babuška. The stability of the domain of definition with respect to basic problems of the theory of partial differential equations especially with respect to the theory of elasticity. *Czechoslovak. Math. J*, 11(86):165–203, 1961.

[10] R. Bacon and J. Bowsher. A discrete model of a struck string. *Acta Acustica united with Acustica*, 41(1):21–27, 1978.

[11] Z. P. Bažant and L. Cedolin. *Stability of structures: elastic, inelastic, fracture and damage theories.* World Scientific, 2010.

[12] M. Benzi. Preconditioning techniques for large linear systems: a survey. *Journal of computational Physics*, 182(2):418–477, 2002.

[13] J.-P. Berenger. Three-dimensional perfectly matched layer for the absorption of electromagnetic waves. *Journal of computational physics*, 127(2):363–379, 1996.

[14] H. M. Berger. *A new approach to the analysis of large deflections of plates.* PhD thesis, California Institute of Technology, Pasadena, California, 1954.

[15] S. Bilbao. Parameterized families of finite difference schemes for the wave equation. *Numerical Methods for Partial Differential Equations*, 20(3):463–480, 2004.

[16] S. Bilbao. A digital plate reverberation algorithm. *Journal of the Audio Engineering Society*, 55(3):135–144, 2007.

[17] S. Bilbao. A family of conservative finite difference schemes for the dynamical von Karman plate equations. *Numerical Methods for Partial Differential Equations*, 24(1):193–216, 2007.

[18] S. Bilbao. A modular percussion synthesis environment. In *Proc. of the 12th Int. Conference on Digital Audio Effects (DAFx-09)*, Como, Italy, 2009.

[19] S. Bilbao. *Numerical sound synthesis: finite difference schemes and simulation in musical acoustics.* Wiley, Chichester, UK, 2009.

[20] S. Bilbao. Time domain simulation and sound synthesis for the snare drum. *The Journal of the Acoustical Society of America*, 131(1):914–925, 2012.

[21] S. Bilbao. Modeling of complex geometries and boundary conditions in finite difference/finite volume time domain room acoustics simulation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 21(7):1524–1533, 2013.

[22] S. Bilbao and J. Chick. Finite difference time domain simulation for the brass instrument bore. *The Journal of the Acoustical Society of America*, 134(5):3860–3871, 2013.

[23] S. Bilbao, B. Hamilton, A. Torin, C. Webb, P. Graham, A. Gray, and J. Perry. Large scale physical modeling sound synthesis. In *Proceedings of the Stockholm Music Acoustics Conference 2013 (SMAC 2013)*, Stockholm, Sweden, 2013.

[24] S. Bilbao, L. Savioja, and J. O. Smith. Parameterized finite difference schemes for plates: Stability, the reduction of directional dispersion and frequency warping. *IEEE transactions on audio, speech, and language processing*, 15(4):1488–1495, 2007.

[25] S. Bilbao, O. Thomas, C. Touzé, and M. Ducceschi. Conservative numerical methods for the full von Kármán plate equations. *Numerical Methods for Partial Differential Equations*, 2015.

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

[27] S. Bilbao, A. Torin, and V. Chatziioannou. Numerical modeling of collisions in musical instruments. *Acta Acustica united with Acustica*, 101(1):155–173, 2015.

[28] S. Bilbao, A. Torin, P. Graham, J. Perry, and G. Delap. Modular physical modeling synthesis environments on GPU. In *Proc. 2014 International Computer Music Conference*, Athens, Greece, 2014.

[29] S. Bilbao and C. Webb. Timpani drum synthesis in 3D on GPGPUs. In *Proc. of the 15th Int. Conference on Digital Audio Effects (DAFx '12)*, York, UK, 2012.

[30] S. Bilbao and C. J. Webb. Physical modeling of timpani drums in 3D on GPGPUs. *Journal of the Audio Engineering Society*, 61(10):737–748, 2013.

[31] J. Blades. *Percussion instruments and their history.* Bold Strummer Limited, Wesport, Connecticut, 1992.

[32] D. Botteldooren. Finite difference time domain study of acoustical cavities with vibrating plate boundaries. In *Proc. of Internoise 93*, pages 1577–1581, Leuven, Belgium, 1993.

[33] D. Botteldooren. Acoustical finite-difference time-domain simulation in a quasi-Cartesian grid. *The Journal of the Acoustical Society of America*, 95(5):2313–2319, 1994.

[34] D. Botteldooren. Finite-difference time-domain simulation of low-frequency room acoustic problems. *The Journal of the Acoustical Society of America*, 98(6):3302–3308, 1995.

[35] J. Botts and L. Savioja. Integrating finite difference schemes for scalar and vector wave equations. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 171–175. IEEE, 2013.

[36] X. Boutillon. Model for piano hammers: Experimental determination and digital simulation. *The Journal of the Acoustical Society of America*, 83(2):746–754, 1988.

[37] J. Bretos, C. Santamarıa, and J. A. Moral. Vibrational patterns and frequency responses of the free plates and box of a violin obtained by finite element analysis. *The Journal of the Acoustical Society of America*, 105(3):1942–1950, 1999.

[38] G. Brien, M. Hyman, and S. Kaplan. A study of the numerical solutions of partial differential equations. *Journal of Mathematical Physics*, 29:223–251, 1951.

[39] D. Brouwer. On the accumulation of errors in numerical integration. *The Astronomical Journal*, 46:149–153, 1937.

[40] M. Bruneau, P. Herzog, J. Kergomard, and J. Polack. General formulation of the dispersion equation in bounded visco-thermal fluid, and application to some simple geometries. *Wave motion*, 11(5):441–451, 1989.

[41] C. Cadoz. Synthèse sonore par simulation de mécanismes vibratoires. Thèse de Docteur Ingénieur, I.N.P.G., Grenoble, France, 1979.

[42] C. Cadoz, A. Luciani, and J. L. Florens. CORDIS-ANIMA: a modeling and simulation system for sound and image synthesis: the general formalism. *Computer Music Journal*, 17(1):19–29, 1993.

[43] C. Camier. *Modélisation et étude numérique des vibrations non-linéaires de plaques circulaires minces imparfaites: application aux cymbales.* PhD thesis, Ecole Polytechnique X, 2009.

[44] A. C. Cangellaris and D. B. Wright. Analysis of the numerical error caused by the stair-stepped approximation of a conducting boundary in FDTD simulations of electromagnetic phenomena. *Antennas and Propagation, IEEE Transactions on*, 39(10):1518–1525, 1991.

[45] G. F. Carrier. On the nonlinear vibration of an elastic string. *Q. Appl. Math.*, 3:157–165, 1945.

[46] J. Chabassier. *Modélisation et simulation numérique d'un piano par modèles physiques.* PhD thesis, Ecole Polytechnique X, 2012.

[47] J. N. Chadwick, S. S. An, and D. L. James. Harmonic shells: a practical nonlinear sound model for near-rigid thin shells. *ACM Transactions on Graphics (SIGGRAPH ASIA Conference Proceedings)*, 28(5), 2009.

[48] A. Chaigne. On the use of finite differences for musical synthesis. Application to plucked stringed instruments. *J. Acoustique*, 5(2):181–211, 1992.

[49] A. Chaigne and A. Askenfelt. Numerical simulations of piano strings. I. A physical model for a struck string using finite difference methods. *The Journal of the Acoustical Society of America*, 95(2):1112–1118, 1994.

[50] A. Chaigne and A. Askenfelt. Numerical simulations of piano strings. II. Comparisons with measurements and systematic exploration of some hammer-string parameters. *The Journal of the Acoustical Society of America*, 95(3):1631–1640, 1994.

[51] A. Chaigne and V. Doutaut. Numerical simulations of xylophones. I. Time-domain modeling of the vibrating bars. *The Journal of the Acoustical Society of America*, 101(1):539–557, 1997.

[52] A. Chaigne, P. Joly, and L. Rhaouti. Numerical modeling of the timpani. In *Proc. of the European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS)*, Barcelona, Spain, 2000.

[53] A. Chaigne and C. Lambourg. Time-domain simulation of damped impacted plates. I. Theory and experiments. *The Journal of the Acoustical Society of America*, 109(4):1422–1432, 2001.

[54] A. Chaigne, C. Touzé, and O. Thomas. Nonlinear vibrations and chaos in gongs and cymbals. *Acoustical science and technology*, 26(5):403–409, 2005.

[55] D. Chapelle and K.-J. Bathe. *The finite element analysis of shells - Fundamentals.* Springer Science & Business Media, Berlin, Germany, 2010.

[56] V. Chatziioannou and M. van Walstijn. An energy conserving finite difference scheme for simulation of collisions. In *Proc. of the Sound and Music Computing Conference (SMC 2013)*, pages 584–591, Stockholm, Sweden, 2013.

[57] V. Chatziioannou and M. van Walstijn. Discrete-time conserved quantities for damped oscillators. In *Proc. 3rd Vienna Talk on Music Acoustics*, pages 135–139, Vienna, Austria, 2015.

[58] V. Chatziioannou and M. van Walstijn. Energy conserving schemes for the simulation of musical instrument contact dynamics. *Journal of Sound and Vibration*, 339:262–279, 2015.

[59] G. A. Chechkin, D. Lukkassen, and A. Meidell. On the Sapondzhyan–Babuška paradox. *Applicable Analysis*, 87(12):1443–1460, 2008.

[60] R. S. Christian, R. E. Davis, A. Tubis, C. A. Anderson, R. I. Mills, and T. D. Rossing. Effects of air loading on timpani membrane vibrations. *The Journal of the Acoustical Society of America*, 76(5):1336–1345, 1984.

[61] I. Chueshov and I. Lasiecka. *Von Karman Evolution Equations: Well-posedness and Long Time Dynamics*. Springer Science & Business Media, Berlin, Germany, 2010.

[62] P. Console and E. Hairer. Long-term stability of symmetric partitioned linear multistep methods. In *Current Challenges in Stability Issues for Numerical Differential Equations*, pages 1–37. Springer, 2014.

[63] P. Console and E. Hairer. Reducing round-off errors in symmetric multistep methods. *Journal of Computational and Applied Mathematics*, 262:217–222, 2014.

[64] P. Cook. *Identification of Control Parameters in an Articulatory Vocal Tract Model, With Applications to the Synthesis of Singing*. PhD thesis, Department of Electrical Engineering, Stanford University, USA, 1990.

[65] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen*, 100(1):32–74, 1928.

[66] S. Dahl. Spectral changes in the tom-tom related to striking force. *Speech Music and Hearing Quarterly Progress and Status Report*, 38(1):059–065, 1997.

[67] J. W. Demmel. *Applied numerical linear algebra*. Siam, Philadelphia, PA, 1997.

[68] G. Derveaux, A. Chaigne, P. Joly, and E. Bécache. Time-domain simulation of a guitar: Model and method. *The Journal of the Acoustical Society of America*, 114(6):3368–3383, 2003.

[69] C. Desvages and S. Bilbao. Physical modeling of nonlinear player-string interactions in bowed string sound synthesis using finite difference methods. In *Proceedings of the International Symposium on Musical Acoustics*, 2014.

[70] J. Diaz and M. J. Grote. Energy conserving explicit local time stepping for second-order wave equations. *SIAM Journal on Scientific Computing*, 31(3):1985–2014, 2009.

[71] C. Dodge and T. A. Jerse. *Computer Music: synthesis, composition and performance*. Schirmer Books, New York, New York, 1985.

[72] V. Doutaut, D. Matignon, and A. Chaigne. Numerical simulations of xylophones. II. Time-domain modeling of the resonator and of the radiated sound pressure. *The Journal of the Acoustical Society of America*, 104(3):1633–1647, 1998.

[73] M. Ducceschi. *Nonlinear Vibrations of Thin Rectangular Plates: A Numerical Investigation with Application to Wave Turbulence and Sound Synthesis.* PhD thesis, ENSTA ParisTech, Paris, France, 2014.

[74] M. Ducceschi and C. Touzé. Modal approach for nonlinear vibrations of damped impacted plates: Application to sound synthesis of gongs and cymbals. *Journal of Sound and Vibration*, 344:313–331, 2015.

[75] M. Ducceschi and C. Touzé. Simulations of nonlinear plate dynamics: An accurate and efficient modal algorithm. In *Proc. of the 18th Int. Conference on Digital Audio Effects (DAFx15)*, Trondheim, Norway, 2015.

[76] G. Eckel, F. Iovino, and R. Caussé. Sound synthesis by physical modelling with Modalys. In *Proc. International Symposium on Musical Acoustics*, pages 479–482, Dourdan, France, 1995.

[77] M. Elejabarrieta, A. Ezcurra, and C. Santamaría. Vibrational behaviour of the guitar soundboard analysed by the finite element method. *Acta Acustica united with Acustica*, 87(1):128–136, 2001.

[78] B. Engquist and A. Majda. Absorbing boundary conditions for numerical simulation of waves. *Proceedings of the National Academy of Sciences*, 74(5):1765–1766, 1977.

[79] L. Euler. De motu vibratorio tympanorum. *Novi Commentari Academiae Scientiarum Petropolitanae*, 10:243–260, 1766.

[80] A. Falaize and T. Hélie. Guaranteed-passive simulation of an electro-mechanical piano: A port-Hamiltonian approach. In *18th Int. Conference on Digital Audio Effects (DAFx-15)*, 2015.

[81] A. Falaize, N. Lopes, T. Hélie, D. Matignon, and B. Maschke. Energy-balanced models for acoustic and audio systems: a port-Hamiltonian approach. In *Unfold Mechanics for Sounds and Music*, pages 1–1, 2014.

[82] A. Fettweis. Wave digital filters: Theory and practice. *Proceedings of the IEEE*, 74(2):270–327, 1986.

[83] H. Fletcher and I. G. Bassett. Some experiments with the bass drum. *The Journal of the Acoustical Society of America*, 64:1570, 1978.

[84] N. H. Fletcher and T. D. Rossing. *The physics of musical instruments.* Springer Science & Business Media, New York, 1998.

[85] A. Föppl. *Vorlesungen über Technische Mechanik*, volume 5, Die wichtigsten Lehren der höheren Elastizitätslehre. Teubner, Leipzig, 1907.

[86] G. E. Forsythe, W. R. Wasow, et al. *Finite-difference methods for partial differential equations.* Wiley, New York, 1960.

[87] A. Gärder. Physical modeling of percussion instruments. Master's thesis, Chalmers University of Technology, Götheborg, Sweden, 2005.

[88] I. M. Gelfand and S. V. Fomin. *Calculus of variations.* Prentice-Hall, Englewood Cliffs, N.J., USA, 1963.

[89] N. Giordano. Simple model of a piano soundboard. *The Journal of the Acoustical Society of America*, 102(2):1159–1168, 1997.

[90] D. Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys (CSUR)*, 23(1):5–48, 1991.

[91] K. F. Graff. *Wave motion in elastic solids.* Dover Publications, New York, 1991.

[92] B. Gustafsson, H.-O. Kreiss, and J. Oliger. *Time dependent problems and difference methods*, volume 24. John Wiley & Sons, New York, 1995.

[93] J. Häggblad and O. Runborg. Accuracy of staircase approximations in finite-difference methods for wave propagation. *Numerische Mathematik*, 128(4):741–771, 2014.

[94] E. Hairer, R. I. McLachlan, and A. Razakarivony. Achieving Brouwer's law with implicit Runge–Kutta methods. *BIT Numerical Mathematics*, 48(2):231–243, 2008.

[95] D. E. Hall. Piano string excitation. VI: Nonlinear modeling. *The Journal of the Acoustical Society of America*, 92(1):95–105, 1992.

[96] B. Hamilton. Finite volume perspectives on finite difference schemes and boundary formulations for wave simulation. *Proc. Digital Audio Effects (DAFx 14), Erlangen, Germany*, 2014.

[97] B. Hamilton. *Finite difference and finite volume methods for wave-based modelling of room acoustics.* PhD thesis, University of Edinburgh, 2016.

[98] B. Hamilton and A. Torin. Finite difference schemes on hexagonal grids for thin linear plates with finite volume boundaries. In *Proc. of the 17th Int. Conference on Digital Audio Effects (DAFx 14)*, Erlangen, Germany, 2014.

[99] B. Hamilton and C. J. Webb. Room acoustics modelling using GPU-accelerated finite difference and finite volume methods on a face-centered cubic grid. *Proc. Digital Audio Effects (DAFx 13), Maynooth, Ireland*, pages 336–343, 2013.

[100] J. R. Hauser. *Numerical methods for nonlinear engineering models.* Springer Science & Business Media, Berlin, Germany, 2009.

[101] P. Henrici. *Discrete variable methods in ordinary differential equations.* Wiley, New York, USA, 1962.

[102] G. Herrmann. Influence of large amplitudes on flexural motions of elastic plates. Technical Report 3578, National Advisory Committee for Aeronautics (NACA), May 1956.

[103] N. J. Higham. *Accuracy and stability of numerical algorithms.* Siam, Philadelphia, PA, 2002.

[104] L. Hiller and P. Ruiz. Synthesizing musical sounds by solving the wave equation for vibrating objects: Part 1. *Journal of the Audio Engineering Society*, 19(6):462–470, 1971.

[105] L. Hiller and P. Ruiz. Synthesizing musical sounds by solving the wave equation for vibrating objects: Part 2. *Journal of the Audio Engineering Society*, 19(7):542–551, 1971.

[106] T. Humbert, O. Cadot, G. Düring, C. Josserand, S. Rica, and C. Touzé. Wave turbulence in vibrating plates: The effect of damping. *EPL (Europhysics Letters)*, 102(3):30002, 2013.

[107] E. V. Jansson. *Acoustics for violin and guitar makers.* Department of Speech, Music and Hearing, Stockholm, Sweden, 2002.

[108] C. H. Jenkins. Nonlinear dynamic response of membranes: state of the art-update. *Applied Mechanics Reviews*, 49(10S):S41–S48, 1996.

[109] C. H. Jenkins and J. W. Leonard. Nonlinear dynamic response of membranes: State of the art. *Applied Mechanics Reviews*, 44(7):319–328, 1991.

[110] P. Joly and L. Rhaouti. A qualitative analysis of a simplified model for the non linear membrane-mallet interaction. Technical Report RR-3234, INRIA, 1997.

[111] M. Karjalainen. BlockCompiler: Efficient simulation of acoustic and audio systems. In *Audio Engineering Society Convention 114*, 2003.

[112] J. L. Kelly and C. C. Lochbaum. Speech synthesis. In *Proceedings of the Fourth International Congress on Acoustics*, pages 1–4, Copenhagen, Denmark, 1962.

[113] R. Kirby and Z. Yosibash. Solution of von-Kármán dynamic non-linear plate equations using a pseudo-spectral method. *Computer methods in applied mechanics and engineering*, 193(6):575–599, 2004.

[114] G. Kirchhoff. Über das gleichgewicht und die bewegung einer elastischen scheibe. *Journal für die reine und angewandte Mathematik*, 40:51–88, 1850 (German).

[115] K. Kowalczyk and M. van Walstijn. Modeling frequency-dependent boundaries as digital impedance filters in FDTD and K-DWM room acoustics simulations. *Journal of the Audio Engineering Society*, 56(7/8):569–583, 2008.

[116] K. Kowalczyk and M. van Walstijn. Room acoustics simulation using 3-D compact explicit FDTD schemes. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(1):34–46, 2011.

[117] J. A. Laird. *The physical modelling of drums using digital waveguides.* PhD thesis, University of Bristol, UK, 2001.

[118] C. Lambourg. *Modèle temporel pour la simulation numérique de plaques vibrantes. Application à la synthèse sonore.* PhD thesis, Université du Maine, Le Mans, France, 1997.

[119] C. Lambourg, A. Chaigne, and D. Matignon. Time-domain simulation of damped impacted plates. II. Numerical model and results. *The Journal of the Acoustical Society of America*, 109(4):1433–1447, 2001.

[120] L. Landau and E. Lifshitz. *Elasticity theory.* Pergamon Press Oxford, 1986.

[121] G. F. Lawler and V. Limic. *Random walk: a modern introduction.* Cambridge University Press, Cambridge, UK, 2010.

[122] P. D. Lax and R. D. Richtmyer. Survey of the stability of linear finite difference equations. *Communications on Pure and Applied Mathematics*, 9(2):267–293, 1956.

[123] L. P. Lebedev and M. J. Cloud. *Tensor analysis.* World Scientific, Singapore, 2003.

[124] A. W. Leissa. Vibration of plates. *NASA SP-160*, 1969.

[125] A. W. Leissa. *Vibration of shells.* Scientific and Technical Information Office, National Aeronautics and Space Administration, Washington, DC, USA, 1973.

[126] A. Leung and S. Mao. A symplectic Galerkin method for non-linear vibration of beams and plates. *Journal of Sound and Vibration*, 183(3):475–491, 1995.

[127] R. J. LeVeque. *Finite volume methods for hyperbolic problems.* Cambridge University Press, Cambridge, UK, 2002.

[128] S. Li and L. Vu-Quoc. Finite difference calculus invariant structure of a class of algorithms for the nonlinear Klein-Gordon equation. *SIAM Journal on Numerical Analysis*, 32(6):1839–1875, 1995.

[129] A. E. H. Love. The small free vibrations and deformation of a thin elastic shell. *Philosophical Transactions of the Royal Society of London. A*, pages 491–546, 1888.

[130] K. S. Miller. On the inverse of the sum of matrices. *Mathematics Magazine*, pages 67–72, 1981.

[131] R. D. Mindlin. Influence of rotatory inertia and shear on flexural motions of isotropic, elastic plates. *Journal of Applied Mechanics*, 18:31–38, 1951.

[132] F. R. Moore. *Elements of computer music.* Prentice-Hall, Inc., Englewood Cliffs, N.J., USA, 1990.

[133] J. D. Morrison and J.-M. Adrien. MOSAIC: A framework for modal synthesis. *Computer Music Journal*, 17(1):45–56, 1993.

[134] P. M. Morse. *Vibration and sound.* McGraw-Hill, New York, USA, second edition, 1948.

[135] P. M. Morse and K. U. Ingard. *Theoretical acoustics.* McGraw-Hill Inc., USA, 1986.

[136] A. H. Nayfeh and D. T. Mook. *Nonlinear oscillations.* John Wiley and Sons, New York, 1979.

[137] A. H. Nayfeh and P. F. Pai. *Linear and nonlinear structural mechanics.* John Wiley & Sons, New York, 2008.

[138] A. Neumaier. *Introduction to numerical analysis.* Cambridge University Press, Cambridge, UK, 2001.

[139] J. Nowinski and H. Ohnabe. On certain inconsistencies in Berger equations for large deflections of plastic plates. *International Journal of Mechanical Sciences*, 14(3):165–170, 1972.

[140] M. L. Overton. *Numerical computing with IEEE floating point arithmetic.* Siam, Philadelphia, PA, 2001.

[141] A. Parret-Fréaud, B. Cotté, and A. Chaigne. Time-domain damping models in structural acoustics using digital filtering. *Mechanical Systems and Signal Processing*, 2015.

[142] J. Perry, S. Bilbao, and A. Torin. Hierarchical parallelism in a physical modelling synthesis code. In *ParCo Conference*, Edinburgh, UK, 2015.

[143] M. R. Portnoff. *A quasi-one-dimensional digital simulation for the time-varying vocal tract.* PhD thesis, Massachusetts Institute of Technology, 1973.

[144] L. Reinhart. On the numerical analysis of the von Karman equations: Mixed finite element approximation and continuation techniques. *Numerische Mathematik*, 39(3):371–404, 1982.

[145] E. Reissner. The effect of transverse shear deformation on the bending of elastic plates. *Journal of applied Mechanics*, 12:69–77, 1945.

[146] L. Rhaouti, A. Chaigne, and P. Joly. Time-domain modeling and numerical simulation of a kettledrum. *The Journal of the Acoustical Society of America*, 105:3545, 1999.

[147] H. C. Rhee and S. N. Atluri. Polygon-circle paradox in the finite element analysis of bending of a simply supported plate. *Computers & Structuresw*, 22(4):553–558, 1986.

[148] L. F. Richardson. *Weather prediction by numerical process.* Cambridge University Press, Cambridge, UK, 1922.

[149] P. G. Richardson, E. R. Toulson, and D. J. Nunn. Analysis and manipulation of modal ratios of cylindrical drums. *The Journal of the Acoustical Society of America*, 131(1):907–913, 2012.

[150] C. Roads. *The computer music tutorial.* MIT press, Cambridge, Massachusetts, 1996.

[151] T. D. Rossing. The physics of kettledrums. *Scientific American*, 247:172–178, 1982.

[152] T. D. Rossing. *Science of percussion instruments*, volume 3 of *Series in Popular Science.* World Scientific, 2000.

[153] T. D. Rossing, I. Bork, H. Zhao, and D. O. Fystrom. Acoustics of snare drums. *The Journal of the Acoustical Society of America*, 92:84, 1992.

[154] T. D. Rossing, R. F. Moore, P. A. Wheeler, D. M. Huber, and R. E. Runstein. *The Science of Sound.* Addison Wesley, 2009.

[155] J. Rudnick and G. Gaspari. *Elements of the random walk: an introduction for advanced students and researchers.* Cambridge University Press, Cambridge, UK, 2004.

[156] P. M. Ruiz. *A technique for simulating the vibration of strings with a digital computer.* PhD thesis, University of Illinois at Urbana-Champaign, 1970.

[157] Y. Saad. *Iterative methods for sparse linear systems.* PWS Publishing, Boston, MA, USA, 1996.

[158] O. Sapondzhyan. Bending of a simply supported polygonal plate. *Akad. Nauk Armyan. SSR. Izv. Fiz.-Mat. Estest. Tehn. Nauki*, 5(2), 1952.

[159] L. Savioja, D. Manocha, and M. Lin. Use of GPUs in room acoustic modeling and auralization. In *Proc. Int. Symposium on Room Acoustics*, Melbourne, Australia, 2010.

[160] J. Sherman and W. J. Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, pages 124–127, 1950.

[161] E. Skrodzka, E. Hojan, and R. Proksza. Vibroacoustic investigation of a batter head of a snare drum. *Archives of Acoustics*, 31(3):289–297, 2006.

[162] J. O. Smith III. A new approach to digital reverberation using closed waveguide networks. In *Proceedings of International Computer Music Conference*, Simon Frasier University, Canada, 1985.

[163] J. O. Smith III. Efficient simulation of the reed-bore and bow-string mechanisms. In *Proceedings of the International Computer Music Conference*, The Hague, The Netherlands, 1986.

[164] J. O. Smith III. Music applications of digital waveguides. Tech. Rep. STAN-M-39, CCRMA, Stanford University, CA, 1987.

[165] J. O. Smith III. *Spectral Audio Signal Processing.* `http://ccrma.stanford.edu/~jos/sasp/`, accessed 12/10/2015. online book, 2011 edition.

[166] A. J. W. Sommerfeld. *Partial differential equations in physics.* Academic Press, New York, USA, 1949.

[167] G. Strang. *Linear algebra and its applications.* Academic Press, New York, USA, 1976.

[168] J. C. Strikwerda. *Finite difference schemes and partial differential equations.* Siam, Philadelphia, PA, 2004.

[169] J. W. Strutt (Baron Rayleigh). *The theory of sound.* (Macmillan 1896) reprinted Dover Publications, New York, USA, 1945.

[170] A. Stulov. Hysteretic model of the grand piano hammer felt. *The Journal of the Acoustical Society of America*, 97(4):2577–2585, 1995.

[171] H. Suzuki. Model analysis of a hammer–string interaction. *The Journal of the Acoustical Society of America*, 82(4):1145–1151, 1987.

[172] A. Taflove and S. C. Hagness. *Computational electrodynamics.* Artech House, Norwood, MA, USA, 2005.

[173] D. Tataru and M. Tucsnak. On the Cauchy problem for the full von Karman system. *Nonlinear Differential Equations and Applications NoDEA*, 4(3):325–340, 1997.

[174] O. Thomas. *Analyse et modélisation de vibrations non-linéaires de milieux minces élastiques-Application aux instruments de percussion.* PhD thesis, Université Pierre et Marie Curie-Paris VI; Telecom ParisTech, 2001.

[175] O. Thomas and S. Bilbao. Geometrically nonlinear flexural vibrations of plates: In-plane boundary conditions and some symmetry properties. *Journal of Sound and Vibration*, 315(3):569–590, 2008.

[176] O. Thomas, C. Touzé, and A. Chaigne. Asymmetric non-linear forced vibrations of free-edge circular plates. Part II: experiments. *Journal of Sound and Vibration*, 265(5):1075–1101, 2003.

[177] O. Thomas, C. Touzé, and A. Chaigne. Non-linear vibrations of free-edge thin spherical shells: modal interaction rules and 1:1:2 internal resonance. *International Journal of Solids and Structures*, 42(11):3339–3373, 2005.

[178] V. Thomée. From finite differences to finite elements: A short history of numerical analysis of partial differential equations. *Journal of Computational and Applied Mathematics*, 128(1):1–54, 2001.

[179] L. L. Thompson and S. Sankar. Dispersion analysis of stabilized finite element methods for acoustic fluid interaction with Reissner–Mindlin plates. *International Journal for Numerical Methods in Engineering*, 50(11):2521–2545, 2001.

[180] S. Timoshenko and S. Woinowsky-Krieger. *Theory of plates and shells*, volume 2. McGraw-Hill Inc., New York, USA, 1959.

[181] A. Torin. Acoustical study of a snare drum: Physical and numerical approach. Master's thesis, Università degli Studi di Padova, Italy, 2011.

[182] A. Torin and S. Bilbao. A 3D Multi-Plate environment for sound synthesis. In *Proc. of the 16th Int. Conference on Digital Audio Effects (DAFx 13)*, Maynooth, Ireland, 2013.

[183] A. Torin and S. Bilbao. Numerical experiments with non-linear double membrane drums. In *Proceedings of the Stockholm Music Acoustics Conference 2013 (SMAC 2013)*, Stockholm, Sweden, 2013.

[184] A. Torin, B. Hamilton, and S. Bilbao. An energy conserving finite difference scheme for the simulation of collisions in snare drums. In *Proc. of the 17th Int. Conference on Digital Audio Effects (DAFx 14)*, Erlangen, Germany, 2014.

[185] A. Torin and M. Newton. Collisions in drum membranes: a preliminary study on a simplified system. In *Proc. of the Int. Symposium on Musical Acoustics (ISMA 2014)*, Le Mans, France, 2014.

[186] A. Torin and M. Newton. Nonlinear effects in drum membranes. In *Proc. of the Int. Symposium on Musical Acoustics (ISMA 2014)*, Le Mans, France, 2014.

[187] C. Touzé. *Analyse et modélisation de signaux vibratoires et acoustiques chaotiques. Application aux instruments de percussion non-linéaires.* PhD thesis, Université Pierre et Marie Curie, Paris, 2000.

[188] C. Touzé, S. Bilbao, and O. Cadot. Transition scenario to turbulence in thin vibrating plates. *Journal of Sound and Vibration*, 331(2):412–433, 2012.

[189] C. Touzé and A. Chaigne. Lyapunov exponents from experimental time series: application to cymbal vibrations. *Acta Acustica United with Acustica*, 86(3):557–567, 2000.

[190] C. Touzé, O. Thomas, and M. Amabili. Transition to chaotic vibrations for harmonically forced perfect and imperfect circular plates. *International Journal of non-linear Mechanics*, 46(1):234–246, 2011.

[191] V. Välimäki, J. Pakarinen, C. Erkut, and M. Karjalainen. Discrete-time modelling of musical instruments. *Reports on progress in physics*, 69(1):1, 2006.

[192] H. A. Van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on scientific and Statistical Computing*, 13(2):631–644, 1992.

[193] S. A. Van Duyne. *Digital Filter Applications to Modeling Wave Propagation in Springs, Strings, Membranes and Acoustical Space.* PhD thesis, Center for Computer Research in Music and Acoustics, Stanford University, Stanford, CA, 2007.

[194] M. van Walstijn and G. Scavone. The wave digital tonehole model. In *Proceedings of International Computer Music Conference*, pages 465–468, Berlin, Germany, 2000.

[195] C. Vendhan. A study of Berger equations applied to non-linear vibrations of elastic plates. *International Journal of Mechanical Sciences*, 17(7):461–468, 1975.

[196] E. Ventsel and T. Krauthammer. *Thin plates and shells: theory, analysis, and applications.* Marcel Dekker, New York, USA, 2001.

[197] G. Vilmart. Reducing round-off errors in rigid body dynamics. *Journal of computational physics*, 227(15):7083–7088, 2008.

[198] T. von Kármán. *Festigkeitsprobleme im maschinenbau.* Encyklopädie der Mathematischen Wissenschaften, 1910.

[199] J. von Neumann and R. D. Richtmyer. A method for the numerical calculation of hydrodynamic shocks. *Journal of applied physics*, 21(3):232–237, 1950.

[200] A. Wagner. Analysis of drumbeats–interaction between drummer, drumstick and instrument. Master's thesis, Department of Speech, Music and Hearing (TMH), Stockholm, Sweden, 2006.

[201] C. J. Webb. *Parallel computation techniques for virtual acoustics and physical modelling synthesis.* PhD thesis, University of Edinburgh, 2014.

[202] C. J. Webb and S. Bilbao. Computing room acoustics with CUDA-3D FDTD schemes with boundary losses and viscosity. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 317–320, Prague, Czech Republic, 2011.

[203] C. J. Webb and S. Bilbao. On the limits of real-time physical modelling synthesis with a modular environment. In *Proc. of the 18th Int. Conference on Digital Audio Effects (DAFx 15)*, Trondheim, Norway, 2015.

[204] C. P. Wilson. Electronic music synthesis: Physical modelling of the drum. MPhys Project Report, University of Edinburgh, 2010.

[205] R. Worland and B. Boe. Measurements of coupled drumhead vibrations using electronic speckle-pattern interferometry. *The Journal of the Acoustical Society of America*, 134(5):4157–4157, 2013.

[206] P. Wriggers. *Computational Contact Mechanics*. Springer-Verlag Berlin Heidelberg, second edition, 2006.

[207] K. Yasuda and T. Asano. Nonlinear forced oscillations of a rectangular membrane with degenerate modes. *Bulletin of JSME*, 29(255):3090–3095, 1986.

[208] K. S. Yee et al. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Trans. Antennas Propag*, 14(3):302–307, 1966.

[209] Z. Yosibash and R. Kirby. Dynamic response of various von-Kármán non-linear plate models and their 3-D counterparts. *International journal of solids and structures*, 42(9):2517–2531, 2005.

[210] O. C. Zienkiewicz and R. L. Taylor. *The finite element method for solid and structural mechanics*. Butterworth-Heinemann, 2005.