

NUMERICAL SOUND SYNTHESIS

Numerical Sound Synthesis

Stefan Bilbao

November 27, 2007

Contents

Preface	xiii
1 Sound Synthesis and Physical Modeling	1
1.1 Abstract Digital Sound Synthesis	2
1.1.1 Additive Synthesis	3
1.1.2 Subtractive Synthesis	5
1.1.3 Wavetable Synthesis	5
1.1.4 AM and FM Synthesis	7
1.1.5 Other Methods	8
1.2 Physical Modeling	9
1.2.1 Lumped Mass-Spring Networks	10
1.2.2 Modal Synthesis	11
1.2.3 Digital Waveguides	13
1.2.4 Hybrid Methods	16
1.2.5 Direct Numerical Simulation	17
1.3 Physical Modeling: A Larger View	20
1.3.1 Physical Models as Descended from Abstract Synthesis	20
1.3.2 Connections: Direct Simulation and Other Methods	21
1.3.3 Complexity of Musical Systems	22
1.3.4 Why?	25
2 Time Series and Difference Operators	27
2.1 Time Series	28
2.2 Shift, Difference and Averaging Operators	29
2.2.1 Temporal Width of Difference Operators	30
2.2.2 Combining Difference Operators	31
2.2.3 Taylor Series and Accuracy	32
2.2.4 Identities	33
2.3 Frequency Domain Analysis	34
2.3.1 Laplace and z transforms	34
2.3.2 Frequency Domain Interpretation of Differential and Difference Operators	35
2.3.3 Recursions and Polynomials in z	36
2.3.4 Difference Operators and Digital Filters	38
2.4 Energetic Manipulations and Identities	41

2.4.1	Time Derivatives of Products of Functions or Time Series	42
2.4.2	Product Identities	44
2.4.3	Quadratic Forms	44
2.5	Problems	46
3	The Oscillator	49
3.1	The Simple Harmonic Oscillator	50
3.1.1	Sinusoidal Solution	50
3.1.2	Energy	52
3.1.3	As First-Order System	52
3.1.4	Coupled Systems of Oscillators	53
3.2	A Simple Scheme	53
3.2.1	As Recursion	54
3.2.2	Initialization	54
3.2.3	Numerical Instability	54
3.2.4	Frequency Domain Analysis	55
3.2.5	Accuracy	56
3.2.6	Energy Analysis	57
3.2.7	Computational Considerations	58
3.3	Other Schemes	59
3.3.1	Using Time-averaging Operators	59
3.3.2	A Second-order Family of Schemes	60
3.3.3	Wave Digital Filters	61
3.3.4	An Exact Solution	64
3.3.5	Further Methods	65
3.4	Lumped Mass-Spring Networks	65
3.5	Loss	66
3.5.1	Energy	67
3.5.2	Finite Difference Scheme	68
3.5.3	Numerical Decay Time	69
3.5.4	An Exact Solution	71
3.6	Sources	71
3.7	Problems	72
3.8	Programming Exercises	75
4	The Oscillator in Musical Acoustics	77
4.1	Nonlinear Oscillators	78
4.2	Lossless Oscillators	78
4.2.1	The Cubic Nonlinear Oscillator	79
4.2.2	Power Law Nonlinearities	82
4.2.3	One-sided Nonlinearities and Collisions: Hammers and Mallets	82
4.2.4	Center-limited Nonlinearities	86
4.3	Lossy Oscillators	87
4.3.1	The Bow	87
4.3.2	Reed and Lip Models	91
4.4	Problems	91

4.5	Programming Exercises	93
5	Grid Functions and Finite Difference Operators in One Dimension	95
5.1	Partial Differential Operators	95
5.1.1	Classification of PDEs	96
5.1.2	Laplace and Fourier Transforms	97
5.1.3	Inner Products and Energetic Manipulations	99
5.2	Grid Functions and Difference Operators in One Dimension	100
5.2.1	Time Difference and Averaging Operators	101
5.2.2	Spatial Difference Operators	101
5.2.3	Mixed Spatial-Temporal Difference Operators	102
5.2.4	Interpolation and Spreading Operators	103
5.2.5	Accuracy of Difference Operators	105
5.2.6	Frequency Domain Interpretation	106
5.2.7	Matrix Interpretation of Difference Operators	108
5.2.8	Boundary Conditions and Imaginary Grid Points	108
5.2.9	Inner Products and Identities	110
5.2.10	Summation by Parts	111
5.2.11	Some Bounds	111
5.2.12	Interpolation and Spreading Revisited	112
5.3	Coordinate Changes	112
5.4	Problems	113
5.5	Programming Exercises	115
6	The 1D Wave Equation	117
6.1	Definition and Properties	118
6.1.1	Linear Array of Masses and Springs	119
6.1.2	Non-dimensionalized Form	120
6.1.3	Initial Conditions	120
6.1.4	Strikes and Plucks: Time Evolution of Solution	121
6.1.5	Dispersion Relation	122
6.1.6	Phase and Group Velocity	122
6.1.7	Travelling Wave Solutions	123
6.1.8	Energy Analysis	123
6.1.9	Boundary Conditions	124
6.1.10	Bounds on Solution Size	126
6.1.11	Modes	128
6.2	A Simple Finite Difference Scheme	131
6.2.1	Initialization	132
6.2.2	von Neumann Analysis	132
6.2.3	Numerical Dispersion	134
6.2.4	Accuracy	136
6.2.5	Energy Analysis	137
6.2.6	Numerical Boundary Conditions	138
6.2.7	Bounds on Solution Size and Numerical Stability	139
6.2.8	State Space Form and Modes	142

6.2.9	Output and Interpolation	144
6.2.10	Implementation Details and Computational Requirements	145
6.2.11	Digital Waveguide Interpretation	147
6.3	Other Schemes	149
6.3.1	A Stencil-width Five Scheme	149
6.3.2	A Compact Implicit Scheme	150
6.4	Modal Synthesis	153
6.5	Loss	153
6.5.1	Finite Difference Scheme	154
6.6	A Comparative Study: Physical Modeling Sound Synthesis Methods	155
6.6.1	Accuracy	155
6.6.2	Memory Use	155
6.6.3	Precomputation	156
6.6.4	Operation Count	156
6.6.5	Stability	156
6.6.6	Generality and Flexibility	156
6.7	Problems	156
6.8	Programming Exercises	159
7	Bars and Linear Strings	161
7.1	The Ideal Uniform Bar	161
7.1.1	Dispersion	162
7.1.2	Energy Analysis and Boundary Conditions	163
7.1.3	Modes	164
7.1.4	A Simple Finite Difference Scheme	166
7.1.5	An Implicit Scheme	170
7.2	Stiff Strings	172
7.2.1	Energy and Boundary Conditions	173
7.2.2	Modes	174
7.2.3	Finite Difference Schemes	174
7.3	Frequency Dependent Loss	175
7.3.1	Energy and Boundary Conditions	177
7.3.2	Finite Difference Schemes	177
7.4	Coupling with Bow Models	178
7.5	Coupling with Hammer and Mallet Models	180
7.6	Multiple Strings	181
7.7	Prepared Strings	183
7.7.1	Springs and Dampers	183
7.7.2	Masses	187
7.7.3	Rattling Elements	188
7.8	Coupled Strings and Bars	188
7.8.1	Connection Types	190
7.8.2	Finite Difference Schemes	191
7.9	Spatial Variation and Stretched Coordinates	193
7.9.1	Strings of Varying Density	193
7.9.2	Bars of Varying Cross-sectional Area	196

7.10	Problems	199
7.11	Programming Exercises	201
8	Nonlinear String Vibration	203
8.1	The Kirchhoff-Carrier String Model	203
8.1.1	Amplitude-dependent Pitch	204
8.1.2	Energy Analysis and Boundary Conditions	205
8.1.3	Finite Difference Schemes	205
8.1.4	A Quasi-modal Form	208
8.1.5	Loss and Pitch Glides	209
8.1.6	A Digital Waveguide Form	209
8.1.7	A Nonlinear Bar Model	209
8.2	General Planar Nonlinear String Motion	209
8.2.1	Coupling Between Transverse and Longitudinal Modes and Phantom Partial	210
8.2.2	Energy	210
8.2.3	Series Approximations	210
8.2.4	A Conservative Finite Difference Scheme	212
8.2.5	An Improved Finite Difference Scheme	214
8.3	Nonplanar String Motion	215
8.3.1	Model Equations	215
8.3.2	Finite Difference Schemes	216
8.3.3	Instability of Planar Motion and Whirling Behaviour	216
8.4	Problems	216
8.5	Programming Exercises	216
9	Acoustic Tubes	217
9.1	Cylindrical Tubes	217
9.2	Conical Tubes	217
9.3	Tonehole Models	217
9.4	Coupling with Reed Models	217
9.5	Webster's Equation and the Vocal Tract	217
9.5.1	Boundary Conditions	218
9.5.2	Modes	218
9.5.3	Finite Difference Schemes	218
9.5.4	The Kelly-Lochbaum Speech Synthesis Model	219
9.5.5	Coupling with a Glottis Model	219
9.6	Problems	219
9.7	Programming Exercises	219
10	Grid Functions and Finite Difference Operators in 2D	221
10.1	Partial Differential Operators and PDEs in Two Space Variables	221
10.1.1	Cartesian and Radial Coordinates	222
10.1.2	Spatial Domains	222
10.1.3	Partial Differential Operators	223
10.1.4	Differential Operators in the Spatial Frequency Domain	223
10.1.5	Inner Products	224

10.2	Grid Functions and Difference Operators: Cartesian Coordinates	224
10.2.1	Frequency Domain Analysis	226
10.2.2	An Inner Product and Energetic Manipulations	226
10.3	Grid Functions and Difference Operators in Two Dimensions: Radial Coordinates . .	226
10.4	Problems	227
10.5	Programming Exercises	227
11	The 2D Wave Equation	229
11.1	The 2D Wave Equation	229
11.1.1	Phase and Group Velocity	230
11.1.2	Energy and Boundary Conditions	231
11.1.3	Modes	231
11.1.4	A Simple Finite Difference Scheme	232
11.1.5	Other Finite Difference Schemes	233
11.1.6	Digital Waveguide Meshes	235
11.1.7	Lumped Networks	237
11.1.8	Modal Synthesis	237
11.2	Problems	237
11.3	Programming Exercises	238
12	Linear Plate Vibration	239
12.1	The Kirchhoff Thin Plate Model	239
12.1.1	Phase and Group Velocity	240
12.1.2	Boundary Conditions	240
12.1.3	Finite Difference Schemes: Cartesian Coordinates	240
12.2	Realistic Plate Models	241
12.2.1	Finite Difference Schemes	241
12.3	Plate Reverberation	241
12.4	Soundboards	241
12.5	Problems	241
12.6	Programming Exercises	241
13	Nonlinear Plates	243
13.1	The Berger Plate Model	244
13.1.1	Definition	244
13.1.2	Energy Analysis	244
13.1.3	A Finite Difference Scheme	244
13.1.4	Pitch Glides	244
13.2	The von Karman Plate Model	244
13.2.1	The Operator $\mathcal{L}[\cdot, \cdot]$	244
13.2.2	Energy Analysis	245
13.2.3	A Finite Difference Scheme	245
13.3	Cymbals and Gongs	245
13.4	Problems	245
13.5	Programming Exercises	245

14 Finite Element Methods	247
14.1 Variational Forms	248
14.2 Shape Functions	248
14.3 Grid Generation	248
14.4 The Stiff String Revisited	248
14.5 The Linear Plate Revisited	248
14.6 Musical Instrument Modeling	248
14.7 Problems	248
14.8 Programming Exercises	248
15 Spectral Methods	249
15.1 Basis Functions	250
15.1.1 Fourier Series	250
15.1.2 Polynomials	250
15.2 Collocation Methods	250
15.3 Galerkin Methods	250
15.4 Spectral Derivatives	250
15.4.1 As Limiting Case of Finite Difference Approximations	250
15.4.2 Chebyshev Grids	250
15.4.3 Fast Fourier Transforms	250
15.5 High Order Time Stepping Methods	250
15.5.1 Forward and Backward Euler	250
15.5.2 Adams-Bashforth Methods	250
15.5.3 Adams-Moulton Methods	250
15.5.4 Runge-Kutta Methods	250
15.6 The Stiff String Again Revisited	250
15.7 The Linear Plate Again Revisited	250
15.8 Problems	250
15.9 Programming Exercises	250
16 Concluding Remarks	251
16.1 Complexity of Musical Systems	251
16.1.1 A Family of LSI Musical Systems	251
16.1.2 Sampling Theorems and Complexity Bounds	252
16.1.3 Modal Representations and Complexity Bounds	252
16.2 Physical Modeling Sound Synthesis Methods: Strengths and Weaknesses	253
16.2.1 Modal Synthesis	253
16.2.2 Digital Waveguides	254
16.2.3 Lumped Mass Spring Networks	254
16.2.4 Direct Simulation	255
16.3 The Future	255

A	MATLAB Code Examples	257
A.1	The Simple Harmonic Oscillator	257
A.2	Hammer Collision with Mass-Spring System	258
A.3	Bowed Mass-Spring System	259
A.4	The 1D Wave Equation: Finite Difference Scheme	259
A.5	The 1D Wave Equation: Modal Synthesis	260
A.6	The 1D Wave Equation: Digital Waveguide Synthesis	261
A.7	The Ideal Bar	262
A.8	The Stiff String	263
A.9	The Kirchhoff-Carrier Equation	265
A.10	Vocal Tract Synthesis	266
A.11	The 2D Wave Equation	266
A.12	Plate Reverberation	267
A.13	Gong	267
B	List of Symbols	269
	Bibliography	272

Preface

Digital sound synthesis is, at the time of writing in 2007, exactly fifty years old. Set against the glacial pace of the development and evolution of acoustic musical instruments in previous centuries, this is not a long time. But given the astonishing rate at which computational power has increased in the past decades, it is fair even to say that digital sound is, if not old, at least enjoying a robust middle age. Many of the techniques that developed early on, during a twelve- to fifteen-year burst of creativity beginning in the late 1950s, have become classics: wavetables, additive and FM synthesis are now the cornerstones of modern synthesis. All of these methods appeared at a time when very close attention had to be paid to operation counts and algorithmic simplicity. In the early days, these algorithms stretched the bounds of computing power, only to produce sound when shoehorned into the busy schedule of a university mainframe by a devoted and generally nocturnal composer. Now, however, sounds of this nature may be produced very cheaply, and are used routinely by musicians in all walks of life.

Beyond allowing musicians to do, faster, what was once grindingly slow, increased computational power has also opened the door to research into newer, more demanding techniques. Certainly, in the last 20 years, the most significant effort has gone into a set of methods known, collectively, as “physical modeling.” The hope is that by sticking to physical descriptions of musical objects, far better synthetic sound quality may be achieved. There are many such methods available—all, however, may ultimately be viewed as numerical simulation techniques, applied to generate an approximate solution to the equations which describe an acoustic entity, such as a string, or drum head, or xylophone bar. The interesting thing is that in one way, this is a step backwards; after all, modern numerical solution techniques are at least 80 years old, and predate the existence of not only the first digital sound synthesis methods, but in fact digital computers themselves! It is also a step backward in another sense—physical modeling techniques are far more computationally intensive than the classic methods, and again, like the old days, algorithm efficiency has become a concern. Is physical modeling a step forward? This is a question that may only be answered subjectively—by listening to the sounds which may be produced in this way.

As mentioned above, physical modeling sound synthesis is an application of numerical simulation techniques. Regardless of the application, when one is faced with solving a problem numerically, many questions arise before any algebraic manipulations or computer coding are attempted. Or, rather, after one has made one or many such attempts, these questions are begged. There are many, but the most important are:

- How faithfully is the solution to be rendered? (Accuracy)
- How long should one reasonably expect to wait for the solution to be computed? (Efficiency)
- How bad is it if, for some unexpected reason, the simulation fails? (Stability)

Though the ideal answers are, of course, “very, not long, and bad,” one might guess that rarely will one be able to design a method which behaves accordingly. Compromises are necessary, and the types of compromises to be made will depend on the application at hand. Regarding the first question above, one might require different levels of accuracy in, for instance, a weather prediction problem, as compared with the design of a novel parachute. As for the second, though in all cases, speedier computation is desirable, in most mainstream simulation applications, the premium is placed rather on accuracy (as per the first question), though in some, such as for instance, control systems built to reduce panel flutter, efficient on-line performance is essential. Finally, because many mainstream simulation applications are indeed intended to run off-line, many techniques have developed over the years in order to control the usual problems in simulation, such as oscillatory behaviour, and instability. In some applications, typically in an offline design scenario, such as in the design of an airfoil, if one encounters numerical results which suffer from these problems, one can adjust a parameter or two, and run the simulation again. But in an online situation, or, if the application is to be used by a non-expert (such as might occur in the case of 3D graphics rendering), the simulation algorithm needs to produce acceptable results with little or no intervention from the user. In other words, it must be robust.

What about sound synthesis then? Numerical simulation methods have indeed, for some time, played a role in pure studies of the acoustics of musical instruments, divorced from sound synthesis applications, which are the subject of this book. For this reason, one might assume that such methods could be applied directly to synthesis. But in fact, the constraints and goals of synthesis are somewhat different from those of scientific research in musical acoustics. Synthesis is a rather special case of an application of numerical methods, in that the result is judged subjectively. Sometimes there is a target sound from a real-world instrument to be reproduced, but another perhaps longer-term goal is to produce sounds from instruments which are wholly imaginary, yet still based on physical principles. Furthermore, these methods are destined, eventually, to be used by composers and musicians, who surely will have little interest in the technical side of sound synthesis, and who are becoming increasingly accustomed to working in a real-time environment. For this reason, it seems sensible to put more emphasis on efficiency and stability, rather than on computing extremely accurate solutions.

Such considerations, as well as the auxiliary concern of programming ease, naturally lead one to employ the simplest numerical methods available, namely finite difference schemes. These have been around for quite a long time, and, in many mainstream applications, have been superseded by newer techniques (some examples of which appear in the later chapters), which are better suited to the complexities of real-world simulation problems. On the other hand, there are many advantages to sticking with a relatively simple framework: these methods are efficient, quite easy to program, and, best of all, one can use quite basic mathematical tools in order to arrive quickly at conclusions regarding the behaviour of such methods. The trick in synthesis, however, is to understand this behaviour in an audio setting, and, unfortunately, the way in which numerical techniques such as finite difference schemes are presented in many standard texts do not address the peculiarities of sound production. This has been one of the main motivations for writing this book.

Every book has a latent agenda. Frequency domain analysis techniques play a central role in both musical acoustics and numerical analysis, and such techniques are not neglected here. The reason for this is, of course, that many important features of real-world systems (such as musical instruments) may be deduced through linearization. But frequency domain techniques constitute a only single point of view—there are many others. The use of energetic principles amounts to more than just

a different slant on the analysis of numerical methods than that provided by frequency domain methods; it is in fact much more general, though at the same time, less revealing—the dynamics of a system is compressed into the time evolution of a single scalar function. The information it does yield, however, is usually exactly what one needs in order to answer thorny questions about, say, the stability of nonlinear numerical methods, as well as how to properly set numerical boundary conditions. It is the key to solid design of numerical methods, and of immense practical utility, and for this reason, is given an elaborate treatment in this book. Besides—it’s interesting.

This work is not really intended directly for musicians or practising acousticians, but rather for working engineers and (especially) doctoral students and researchers working on the more technical side of digital audio and sound synthesis. Nor is it meant as a collection of recipes, despite the inclusion of a body of code examples on an accompanying website. I realize that the audience for this book will be narrowed somewhat (and maybe a little disappointed) because of this. The reason for this is that physical modeling synthesis is really numerical simulation, a discipline which is somewhat more removed from audio processing than many might like to believe. There is a need, I think, to step back from the usual body of techniques which has been employed for this purpose, generally those which evolved out of the language and tools of electrical engineers, namely digital signal processing, and to take a look at things the way a simulation specialist might. The body of techniques is different enough to require a good deal of mathematics which may be unfamiliar to the audio engineer. At the same time, the audio-informed point of view taken here may seem foreign to the simulation specialist. It is my greatest hope that this book will serve to engender curiosity in the union of these two sets of people—in the ultimate interest, of course, of producing new and beautiful sounds.

Book Summary

Chapter 1 is a historical overview of digital sound synthesis techniques—though far from complete, it highlights the (sometimes overlooked) links between abstract sound synthesis methods, based essentially on signal processing manipulations, and more modern physical modeling sound synthesis methods, as well as the connections among the various physical modeling methodologies.

In Chapter 2, time series and difference operators are introduced, and some time is spent on the frequency domain interpretation of such operators, as well as on the development of certain manipulations which are of use in energy analysis of finite difference schemes. Special attention is paid to the correspondence between finite difference operations and simple digital filter designs.

The simple harmonic oscillator is introduced in Chapter 3, and serves as a model for many of the systems which appear throughout the rest of the book. Various difference schemes are analyzed, especially with respect to numerical stability and accuracy, using both frequency domain and energetic principles; the linear loss mechanism is also introduced.

Chapter 4 introduces various nonlinear excitation mechanisms in musical acoustics, many of which reduce to nonlinear generalizations of the harmonic oscillator, as well as associated finite difference schemes.

Chapter 5 is designed as a reference chapter for the remainder of the book, with a complete introduction to the tools for the construction of finite difference schemes for partial differential equations in time and one spatial dimension, including grid functions, difference operators, as well as a description of frequency domain techniques and inner product formulations, which are useful for nonlinear problems and the determination of numerical boundary conditions.

As a test problem, the 1D wave equation and a variety of numerical methods are presented

in Chapter 6. Various features of interest in musical simulations, including proper settings for boundary conditions, readout and interpolation, numerical dispersion and its perceptual significance, and numerical stability conditions are discussed. In addition, finite difference schemes are related to modal methods, digital waveguides and lumped networks, and relative strengths and weaknesses are evaluated.

Chapter 7 deals with more musical extensions of the 1D wave equation and finite difference schemes to the case of transverse vibration of bars and stiff strings, and considerable time is spent on loss modelling as well as the coupling with hammer/mallet and bow models, and coupling with lumped elements and between bars. The chapter ends with a short treatment of spatially varying string and bar systems.

The first serious foray into numerical methods for distributed nonlinear systems occurs in Chapter 8, with a discussion of nonlinear string vibration. Various models, of differing degrees of complexity are presented, and certain important perceptual effects of string nonlinearity, such as pitch glides and phantom partial generation are discussed. The construction of sound synthesis methods makes use of purely energetic techniques in this case.

Chapter 9 picks up from the end of Chapter 6 to deal with linear wave propagation in acoustic tubes, which are the resonating elements in woodwind and brass instruments. Webster's equation, and finite difference methods are introduced, as are features of musical interest such as tone hole modelling, bell radiation, and coupling to reed-like excitation mechanisms.

Chapters 10, 11 and 12 are analogous to Chapters 5, 6 and 7, but in two spatial dimensions. Chapter 10 is a concise survey of difference operators and grid functions in both Cartesian and radial coordinates. Chapter 11 deals with the important test case of the 2D wave equation, and Chapter 12 constitutes the first discussion of two-dimensional musical instruments, and in particular membranes and plates. Special cases such as drum membranes, plate reverberation and piano soundboards are dealt with in detail. Mallet interaction, two-dimensional interpolation necessary for sound output, and direction-dependent numerical dispersion in finite difference schemes, as well as loss modeling are also discussed.

Chapter 13 continues with the topic of two-dimensional vibration, in the nonlinear case, in order to deal with perceptually crucial effects such as crashes in instruments such as gongs and cymbals, and, as in Chapter 8, energy methods are developed.

Finally, in Chapters 14 and 15, some other numerical techniques with potential applications to musical sound synthesis, in particular finite element methods and spectral methods, are introduced. Concluding remarks appear in Chapter 16. Appendix A contains some rudimentary Matlab scripts which yield synthetic sound output based on many of the models discussed in this book. A glossary of symbols is provided in Appendix B.

As Teaching Aid

This book could be used as a teaching aid, for students at the Master's or PhD level. A strong background in digital signal processing, physics and computer programming is essential. Particular topics in applied mathematics which are assumed prerequisites are the usual ones found in a respectable undergraduate program in the physical sciences or engineering: differential and integral calculus (multivariable), linear algebra, complex analysis, ordinary and partial differential equations, Fourier, Laplace and z transforms, and some familiarity with functional analysis, and in particular the notion of L_2 inner product spaces. For a full year course at the Master's or PhD level, one could comfortably cover the part of the book which deals mainly with linear systems, namely Chapters 1,

2, 3, 5, 6, 7, 9, 10, 11 and 12. The material on nonlinear systems and other topics in the remainder of the book would perhaps be best left to a subsequent seminar-based course. The programming exercises and examples are all based around the use of the Matlab language, which is, of course, ideal for prototyping sound synthesis algorithms, but not for practical applications; the translation of some of these algorithms to a more suitable (perhaps real time) environment would make for an excellent, and practically useful independent study project.

Other Reading

For those new to sound synthesis and physical modeling, there are various texts which are worth consulting:

The physics of musical instruments is covered in the texts by Fletcher and Rossing [96], Rossing, Moore and Wheeler, [216] and Campbell and Greated [47]. A more advanced treatment for many of the systems encountered in this book is given in the classic texts by Morse and Ingard [174], Graff [111] and Nayfeh and Mook [178]. Many interesting aspects of musical instrument physics are covered in the collection edited by Hirschberg, Kergomard and Weinreich [122].

For a general overview of digital sound synthesis, see the books by Roads [205], Dodge and Jerse [75], and Moore [170], and various edited collections [206, 72, 207]. Special topics in physical modeling sound synthesis are covered in various texts. For an exhaustive presentation of digital waveguides, see the text by Smith [242], readily available on line, and certainly the best reference in existence on physical modeling. Functional transformation approaches, which are similar to modal synthesis methods, are discussed in Trautmann and Rabenstein [257]. A variety of sound synthesis techniques, including a good deal of material on both digital waveguides and modal methods are found in the book by Cook [64].

A good introduction to finite difference methods is the text by Strikwerda [244], which covers frequency domain analysis in great detail, and from a point of view that will be accessible to those with an audio signal processing background; indeed, some of the notation used here is borrowed from Strikwerda's book. The text of Gustaffson, Kreiss, and Oliger [113], which is written at a more advanced level, deals with energy techniques as well. The text by Ames [3], though much older, is an invaluable reference. That of Evans, Blackledge and Yardley [86] is an accessible introduction to finite element as well as finite difference methods. A lively reference on spectral methods, conceived around the use of Matlab, is the book by Trefethen [260].

Acknowledgements

The writing of this book was greatly facilitated through the support of the Engineering and Physical Sciences Research Council UK, and the Leverhulme Trust.

Edinburgh, November 27, 2007

Chapter 1

Sound Synthesis and Physical Modeling

Before entering into the main development of this book, it is worth stepping back to get a larger picture of the history of digital sound synthesis. It is, of course, impossible to present a complete treatment of all that has come before, and unnecessary, considering that there are several books which cover the classical core of such techniques in great detail; those of Moore [170], Dodge and Jerse [75] and Roads [205], and the collections of Roads et al. [206], Roads and Strawn [207] and DePoli et al. [72] are perhaps the best known. For a more technical viewpoint, see the report of Tolonen et al. [254], the online text of Puckette [194], and, for physical modeling techniques, the review article of Välimäki et al. [267]. This chapter is intended to give the reader a basic familiarity with the development of such methods, and some of the topics will be examined in much more detail later in this book. Indeed, many of the earlier developments are perceptually intuitive, and involve only basic mathematics; this is less so in the case of more recent work, but every effort will be made to keep the technical jargon in this chapter to a bare minimum.

It is convenient to make a distinction between earlier, or abstract digital sound synthesis methods, to be discussed in §1.1, and those built around physical modeling principles, as detailed in §1.2. (Other, more elaborate taxonomies have been proposed [236, 254], but the above is sufficient for the present purposes.) That this distinction is perhaps less clear-cut than it is often made out to be is a matter worthy of discussion—see §1.3, where some more general comments on physical modeling sound synthesis are offered, namely regarding the relationship among the various physical modeling methodologies and with earlier techniques, and the fundamental limitations of computational complexity.

In Figure 1.1, for the sake of reference, a timeline showing the development of digital sound synthesis methods is presented; dates are necessarily approximate. For brevity, only those techniques which bear some relation to physical modeling sound synthesis are noted—such a restriction is a subjective one, and is surely a matter of some debate.

variety of hybrids and refinements has resulted; only a few of these will be detailed here.

The word “abstract,” though it appears seldom in the literature [240, 254], is used to describe the techniques mentioned above because, in general, they do not possess an associated underlying physical interpretation—the resulting sounds are produced according to perceptual and mathematical, rather than physical principles. There are some loose links with physical modeling, most notably between additive methods and modal synthesis (see §1.1.1), subtractive synthesis and source-filter models (see §1.1.2), and wavetables and wave propagation in one-dimensional media (see §1.1.3), but in general, it is best to think of these methods as pure constructs in digital signal processing, informed by perceptual, programming and sometimes efficiency considerations. For more discussion of the philosophical distinctions between abstract techniques and physical modeling, see the articles by Smith [240] and Borin et al. [36].

1.1.1 Additive Synthesis

Additive synthesis, which dates back at least as far as the work of Risset in the 1960s [202], though not the oldest digital synthesis method, is a convenient starting point; for more information on the history of the development of such methods, see [205] and [163]. A single sinusoidal oscillator, with output $u(t)$ is defined, in continuous time, as

$$u(t) = A \cos(2\pi f_0 t + \phi) \quad (1.1)$$

Here, t is a time variable, and A , f_0 and ϕ are the amplitude, frequency, and initial phase of the oscillator, respectively. In the simplest, strictest configuration of additive synthesis, these parameters are constants: A scales roughly with perceived loudness, and f_0 with pitch. For a single oscillator in isolation, the initial phase ϕ is of minimal perceptual relevance, and is usually not represented in typical symbolic representations of the oscillator—see Figure 1.2. In discrete time, where the sample rate is given by f_s , the oscillator with output u^n is defined similarly by

$$u^n = A \cos(2\pi f_0 n / f_s + \phi) \quad (1.2)$$

where n is an integer, indicating the time step.

The sinusoidal oscillator, in computer music applications, is often represented using the symbolic shorthand shown in Figure 1.2(a). Using Fourier theory, it is possible to show that any real-valued continuous or discrete waveform (barring some technical restrictions relating to continuity) may be decomposed into an integral over a set of such sinusoids. In continuous time, if the waveform to be decomposed is periodic with period T , then an infinite sum of such sinusoids, with frequencies which are integer multiples of $1/T$ suffices to describe the waveform completely. In discrete time, if the waveform is periodic with integer period $2N$, then a finite collection of N oscillators yields a complete characterization.

The musical interest of additive synthesis, however, is not necessarily in exact decompositions of given waveforms. Rather, it is a loosely defined body of techniques based around the use of combinations of such oscillators in order to generate musical sounds, given the underlying assumption that sinusoids are of perceptual relevance in music. (Some might find this debatable, but the importance of pitch throughout the history of acoustic musical instruments across almost all cultures

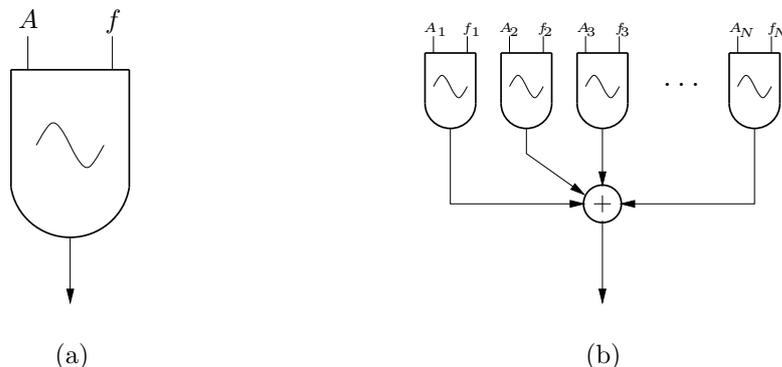


Figure 1.2: (a) Symbolic representation of a single sinusoidal oscillator, output at bottom, dependent on the parameters A , representing amplitude, and f , representing frequency. In this representation, the specification of the phase ϕ has been omitted, though some authors replace the frequency control parameter by a phase increment, and indicate the base frequency in the interior of the oscillator symbol. (b) An additive synthesis configuration, consisting of a parallel combination of N such oscillators, with parameters A_l , and f_l , $l = 1, \dots, N$, according to (1.3).

appears to favor this assertion.) A simple configuration is given, in discrete time, by the sum

$$u^n = \sum_{l=1}^N A_l \cos(2\pi f_l n / f_s + \phi_l) \quad (1.3)$$

where in this case, N oscillators, of distinct amplitudes, frequencies and phases A_l , f_l and ϕ_l , for $l = 1, \dots, N$ are employed. See Figure 1.2(b). If the frequencies f_l are close to integer multiples of a common “fundamental” frequency f_0 , then, in general, the result will be a tone at a pitch corresponding to f_0 . But unpitched inharmonic sounds (such as those of bells) may be generated as well, through avoidance of common factors among the chosen frequencies. With a large enough N , one can, as mentioned above, generate any imaginable sound. But the generality of such an approach is mitigated by the necessity of specifying perhaps thousands of amplitudes, frequencies and phases. For a large enough N , and taking the entire space of possible choices of parameters, the set of sounds which will *not* sound simply like a steady unpitched tone, is vanishingly small. Unfortunately, using such a simple sum of sinusoids, almost any musically interesting sound will certainly lie in the realm of large N .

Various strategies (probably hundreds) have been employed to render additive synthesis more musically tractable. Perhaps the simplest is to employ slowly time-varying amplitude envelopes to the outputs of single oscillators or combinations of oscillators, allowing global control of the attack/decay characteristics of the resulting sound without having to rely on delicate phase cancellation phenomena. Another is to allow oscillator frequencies to vary, at sub-audio rates, so as to approximate changes in pitch. In this case, the definition (1.1) should be extended to include the notion of instantaneous frequency—see §1.1.4. For an overview of these techniques, and others, see the standard texts mentioned in the opening remarks of this chapter.

Another related approach adopted by many composers has been that of analysis-resynthesis, based on sampled waveforms. This is not, strictly speaking, a pure synthesis technique, but it has

become so popular that it is worth mentioning here. Essentially, an input waveform is decomposed into sinusoidal components, at which point the frequency-domain data (amplitudes, phases, and sometimes frequencies) are modified in a perceptually meaningful way, and the sound is then reconstructed through inverse Fourier transformation. Perhaps the best known tool for analysis-synthesis is the phase vocoder [94, 191, 76], which is based on the use of the short-time Fourier transformation, which employs the fast Fourier transformation [67]. Various effects, including pitch transposition and time-stretching, as well as cross-synthesis of spectra can be obtained, through judicious modification of frequency domain data. Even more refined tools, such as spectral modeling synthesis (SMS) [230], based around a combination of Fourier and stochastic modeling, as well as methods employing tracking of sinusoidal partials [165], allow very high quality manipulation of audio waveforms.

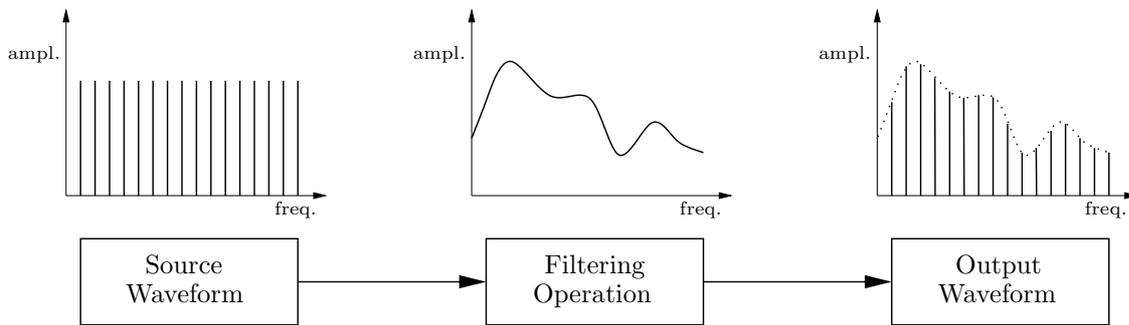
1.1.2 Subtractive Synthesis

If one is interested in producing sounds with rich spectra, additive synthesis, requiring a separate oscillator for each desired frequency component, can obviously become quite a costly undertaking. Instead of building up a complex sound, one partial at a time, another way of proceeding is to begin with a very rich sound, typically simple to produce and lacking in character such as white noise or an impulse train, and then shape the spectrum using digital filtering methods. This technique is often referred to as subtractive synthesis—see Figure 1.3. It is especially powerful when the filtering applied is time-varying, allowing for a good first approximation to the timbre of many musical tones of unsteady timbre (this is generally the norm).

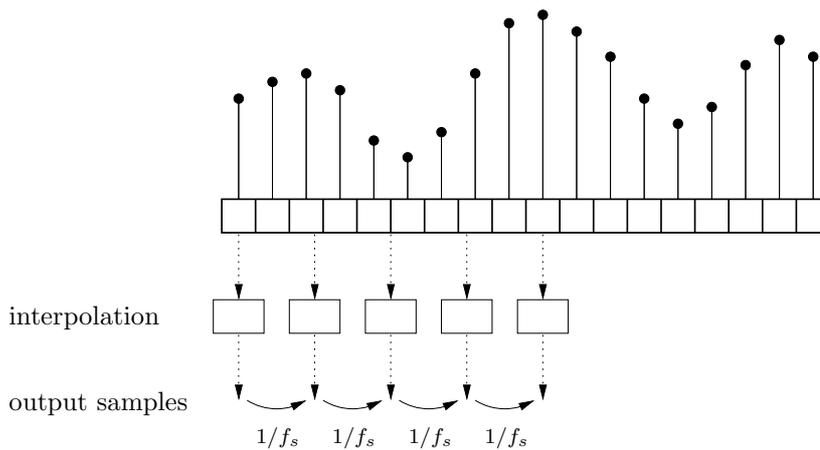
Subtractive synthesis is often associated with physical models [170], but this association is a very loose one at best. What is meant is that many linear models of sound production may be broken down into source and filtering components [285]. This is particularly true of models of human speech, in which case the glottis is assumed to produce a wide band signal (i.e., a signal somewhat like an impulse train under voiced conditions, and white noise under unvoiced conditions) which is filtered by the vocal tract, yielding a spectrum with pronounced peaks (formants) which indicate a particular vocal timbre. In this book, however, because of the emphasis on time domain methods, the source/filter methodology will not be explicitly employed. Indeed, for distributed nonlinear problems, for which frequency domain analysis is not well suited, it is of little use, and relatively uninformative. Even in the linear case, it is worth keeping in mind that the connection of two objects will, in general, modify the characteristic frequencies of both—strictly speaking, one cannot speak of the “frequencies” of the individual components in a coupled system. Still, the breakdown of a system into an lumped/distributed pair representing an excitation mechanism and the instrument body is a very powerful one, even if, in some cases, the behaviour of the body cannot be explained in terms of filtering concepts.

1.1.3 Wavetable Synthesis

The most common computer implementation of the sinusoidal oscillator is not through direct calculation of values of the cosine or sine function, but, rather, through the use of a stored table containing values of one period of a sinusoidal waveform. A sinusoid at a given frequency may then be generated by reading through the table, circularly, at an appropriate rate. If the table contains N values, and the sample rate is f_s , then the generation of a sinusoid at frequency f_0 will require a jump of f_s/f_0N values in the table over each sample period, using interpolation of some form. Clearly, the quality of the output will depend on the number of values stored in the table, as well

Figure 1.3: *Subtractive synthesis.*

as on the type of interpolation employed. Linear interpolation is the norm [170], but other more accurate methods, built around higher order Lagrange interpolation are also used—some material on fourth order interpolation appears in §5.2.4. Allpass filter approximations to fractional delays are also possible, and are of special interest in physical modeling applications [153].

Figure 1.4: *Wavetable synthesis.* A buffer, filled with values, is read through at intervals of $1/f_s$ s, where f_s is the sample rate. Interpolation is employed.

It should be clear that one can store values of an arbitrary waveform in the table, not merely those corresponding to a sinusoid. See Figure 1.4. Reading through such a table at a fixed rate will generate a quasi-periodic waveform with, in general, a full harmonic spectrum, all at the price of a single table read and interpolation operation per sample period—it is no more expensive, in terms of computer arithmetic, than a single oscillator. As will be seen shortly, there is an extremely fruitful physical interpretation of wavetable synthesis, namely the digital waveguide, which revolutionized physical modeling sound synthesis through the same efficiency gains—see §1.2.3. Various other variants of wavetable synthesis have seen use, in particular wavetable stacking, involving multiple

wavetables, the outputs of which are combined using crossfading techniques [205]. The use of tables of data in order to generate sound is perhaps the oldest form of sound synthesis, dating back to the work of Mathews in the late 1950s.

Tables of data are also associated with so-called sampling synthesis techniques, as a *de facto* means of data reduction. Many musical sounds consist of a short attack, followed by a steady pitched tone. Such a sound may be efficiently reproduced through storage of only the attack and a single period of the pitched part of the waveform, which is stored in a wavetable and looped [254]. Such methods are the norm in most commercial digital piano emulators.

1.1.4 AM and FM Synthesis

Some of the most important developments in early digital sound synthesis derived from extensions of the oscillator, through time-variation of the control parameters at audio rates.

AM, or amplitude modulation synthesis, in continuous time, and employing a sinusoidal carrier (of frequency f_0), and modulator (of frequency f_1) generates a waveform of the following form:

$$u(t) = (A_0 + A_1 \cos(2\pi f_1 t)) \cos(2\pi f_0 t)$$

where A_0 and A_1 are free parameters. The symbolic representation of AM synthesis is shown in Figure 1.5(a). Such an output consists of three components, as also shown in Figure 1.5(a), where the strength of the component at the carrier frequency is determined by A_0 , and those of the side components, at frequencies $f_0 \pm f_1$ by A_1 . If $A_0 = 0$, then ring modulation results. Though the above example is concerned with the product of sinusoidal signals, the concept of amplitude modulation (and frequency modulation, discussed below) extends to more general signals with ease.

Frequency modulation (FM) synthesis, the result of a serendipitous discovery by John Chowning at Stanford in the late 1960s, was perhaps the greatest single breakthrough in digital sound synthesis [56]. Instantly, it became possible to generate a wide variety of spectrally rich sounds using a bare minimum of computer operations. FM synthesis requires no more computing power than a few digital oscillators, which is not surprising, considering that FM refers to the modulation of the frequency of a digital oscillator. As a result, real-time synthesis of complex sounds became possible in the late 1970s, as the technique was incorporated into various special purpose digital synthesizers—see [207] for details. In the 1980s, FM synthesis was very successfully commercialized by the Yamaha corporation, and thereafter permanently altered the synthetic soundscape.

FM synthesis, like AM, is also a direct descendant of synthesis based on sinusoids, in the sense that in its simplest manifestation, it makes use of only two sinusoidal oscillators, one behaving as a carrier, and the other as a modulator. See Figure 1.5(b). The functional form of the output, in continuous time, is usually written in terms of sine functions, and not cosines, as

$$u(t) = A_0(t) \sin(2\pi f_0 t + I \sin(2\pi f_1 t)) \tag{1.4}$$

where here, f_0 is the carrier frequency, f_1 the modulation frequency, and I the so-called modulation index. It is straightforward to show [56] that the spectrum of this signal will exhibit components at frequencies $f_0 + qf_1$, for integer q , as illustrated in Figure 1.5(b). The modulation index I determines the strengths of the various components, which can vary in a rather complicated way, depending on the values of associated Bessel functions. $A_0(t)$ can be used to control the envelope of the resulting sound.

In fact, a slightly better formulation of the output waveform (1.4) is:

$$u(t) = A_0(t) \sin \left(2\pi \int_0^t f_0 + I f_1 \cos(2\pi f_1 t') dt' \right)$$

where the instantaneous frequency at time t may be seen to be (or rather defined as) $f_0 + I f_1 \cos(2\pi f_1 t)$. The quantity $I f_1$ is often referred to as the peak frequency deviation, and written as Δf [170]. Though this is a subtle point, and not one which will be returned to in this book, the symbolic representation in Figure 1.5(b) should be viewed in this respect.

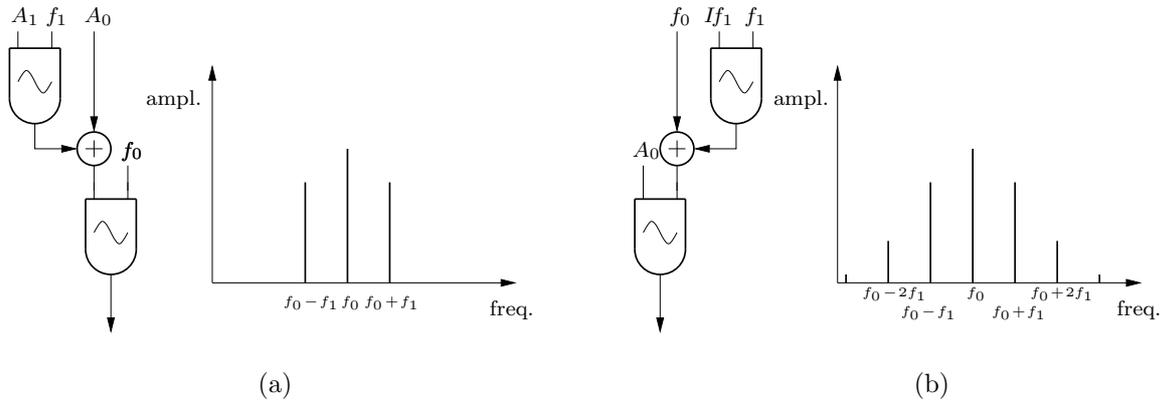


Figure 1.5: *Symbolic representation, and frequency domain description of output for (a) amplitude modulation and (b) frequency modulation.*

FM synthesis has been exhaustively researched, and many variations have resulted. Among the most important are feedback configurations, useful in regularizing the behaviour of the side component magnitudes and various series and parallel multiple oscillator combinations.

1.1.5 Other Methods

There is no shortage of other techniques which have been proposed for sound synthesis; some are variations on the techniques described in the above sections, but there are several which do not fall neatly into any of the above categories. This is not to say that such techniques have not seen success; it is rather that they do not fit naturally into the evolution of abstract methods into physically-inspired sound synthesis methods, the subject of this book.

One of the more interesting is a technique called waveshaping [155, 8, 204], in which case an input waveform (of natural or synthetic origin) is used as a time-varying index to a table of data. This, like FM synthesis, is a nonlinear technique—generally, a sinusoid at a given frequency used as the input will generate an output which contains a number of harmonic components, whose relative amplitudes depend on the values stored in the table. Similar to FM, it is capable of generating rich spectra for the computational cost of a single oscillator, accompanied by a table read; a distinction is that the level of control over the amplitudes of the various partials can be controlled in a direct way, through the use of Chebyshev polynomials expansions as a representation of the table data.

Granular synthesis [50], which is very popular among composers, refers to a large body of techniques, sometimes very rigorously defined (particularly when related to wavelet decompositions [85]), sometimes very loosely. In this case, the idea is to build complex textures using short duration sound “grains,” which are either synthetic, or derived from analysis of an input waveform. The grains, regardless of how they are obtained, may then be rearranged and manipulated in a variety of ways. Granular synthesis encompasses so many different techniques and methodologies that it is perhaps better thought of as a philosophy, rather than a synthesis technique. See [203] for a historical overview.

Distantly related to granular synthesis are methods based on overlap-adding of pulses of short duration, sometimes, but not always, to emulate vocal sounds. The pulses are of a specified form, and depend on a number of parameters which serve to alter the timbre; in a vocal setting, the rate at which the pulses recur determines the pitch, and a formant structure, dependent on the choice of the free parameters, is imparted to the sound output. Perhaps the best known are the so-called FOF [212], and the VOSIM [130] techniques.

1.2 Physical Modeling

The algorithms mentioned above, despite their structural elegance and undeniable power, share several shortcomings. The issue of actual sound quality is difficult to address directly, as it is inherently subjective—it is difficult to deny, however, that in most cases, abstract sound synthesis output is synthetic-sounding. This can be desirable or not, depending on one’s taste. On the other hand, it is worth noting that perhaps the most popular techniques employed by today’s composers are based on modification and processing of sampled sound, indicating that the natural quality of acoustically-produced sound is not easily abandoned. Indeed, many of the earlier refinements of abstract techniques such as FM were geared towards emulating acoustic instrument sounds [171, 228]. The deeper issue, however, is one of control. Some of the algorithms mentioned above, such as additive synthesis, require the specification of an inordinate amount of data, perhaps thousands of parameters to describe a single sound. Others, such as FM synthesis, involve many fewer parameters, but it can be extremely difficult to determine rules for the choice and manipulation of parameters, especially in a complex configuration involving more than a few such oscillators. See [37, 36, 254] for a fuller discussion of the difficulties inherent in abstract synthesis methods.

Physical modeling synthesis, which has developed more recently, involves a physical description of the musical instrument as the starting point for algorithm design. For most musical instruments, this will be a coupled set of partial differential equations, describing, e.g., the displacement of a string, membrane, bar, or plate, or the motion of the air in a tube, etc. The idea, then, is to solve the set of equations, invariably through a numerical approximation, to yield an output waveform, subject to some input excitation (such as glottal vibration, bow or blowing pressure, etc.). The issues mentioned above, namely those of the synthetic character and control of sounds are rather neatly dealt with in this case—there is a virtual copy of the musical instrument available to the algorithm designer or performer, embedded in the synthesis algorithm itself, which serves as a reference. For instance, simulating the plucking of a guitar string at a given location may be accomplished by sending an input signal to the appropriate location in computer memory, corresponding to an actual physical location on the string model; plucking it strongly involves sending a larger signal. The control parameters, for a physical modeling sound synthesis algorithm are typically few in number, and physically and intuitively meaningful, as they relate to material properties, instrument geometry,

and input forces and pressures.

The main drawback to using physical modelling algorithms is, and has been, their relatively large computational expense; in many cases, this amounts to hundreds, if not thousands of arithmetic operations to be carried out per sample period, at a high audio sample rate (such as 44.1 kHz). In comparison, a bank of six FM oscillators will require probably at most twenty arithmetic operations/table lookups per sample period. For this reason, research into such methods has been slower to take root, even though the first such work on musical instruments began with Ruiz in the late 1960s and early 1970s [220], and digital speech synthesis based on physical models can be dated back even further, to the work of Kelly and Lochbaum [141]. On the other hand, computer power has grown enormously in the past decades, and presumably will continue to do so, and, thus, efficiency (an obsession in the earlier days of digital sound synthesis) will become less and less of a concern.

1.2.1 Lumped Mass-Spring Networks

The use of a lumped network, generally of mechanical elements such as masses and springs, as a musical sound synthesis construct, is an intuitively appealing one. It was proposed by Cadoz, Luciani and Florens in the early 1980s [44], and became the basis for the CORDIS, and CORDIS-ANIMA synthesis environments [97, 45, 248]; as such, it constituted the first large scale attempt at physical modeling sound synthesis. It is also the technique which is most similar to the direct simulation approaches discussed throughout the remainder of this book, though the emphasis here is entirely on fully distributed modelling, rather than lumped representations.

The framework is very simply described in terms of interactions among lumped masses, connected by springs and damping elements; when Newton’s laws are employed to describe the inertial behaviour of the masses, the dynamics of such a system may be described by a set of ordinary differential equations. Interaction may be introduced through so-called “conditional links,” which can represent nonlinear contact forces. Time integration strategies, similar to those introduced in Chapter 3 in this book, operating at the audio sample rate (or sometimes above, in order to reduce frequency warping effects), are employed in order to generate sound output. The basic operation of this method will be described in more detail in §3.4.

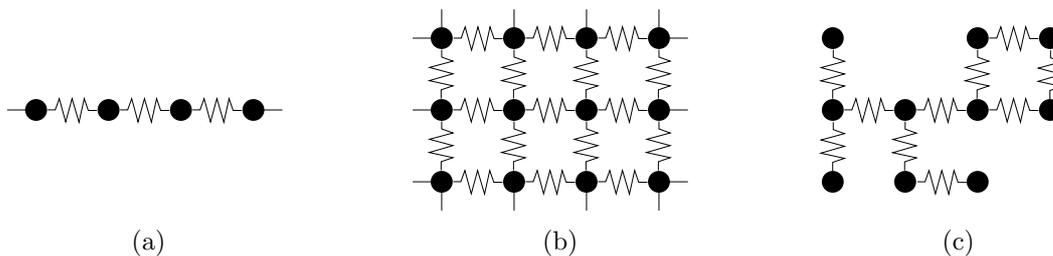


Figure 1.6: *Lumped mass-spring networks: (a) In a linear configuration corresponding to a model of a lossless string, (b) in a 2D configuration corresponding to a model of a lossless membrane, and (c) an unstructured network, without a distributed interpretation.*

A little imagination might lead one to guess that, with a large enough collection of interconnected masses, a distributed object such as a string, as shown in Figure 1.6(a), or membrane, as shown in

Figure 1.6(b), may be modelled. Such configurations will be discussed explicitly in §6.1.1 and §11.1.7, respectively. A rather large philosophical distinction between the CORDIS framework and that described here is that one can develop lumped networks which are, in a sense, only quasi-physical, in that they do not correspond to recognizable physical objects, though the physical underpinnings of Newton's Laws remain. See Figure 1.6(c). Accurate simulation of complex distributed systems has not been a major concern of the designers of the CORDIS; rather, the interest is in user issues such as the modularity of lumped network structures, and the ability to interact through external control. In short, it is perhaps best to think of CORDIS as a system designed for artists and composers, rather than scientists.

1.2.2 Modal Synthesis

A different approach, with a long history of use in physical modeling sound synthesis, is based on a frequency-domain, or modal description of vibration of distributed objects. Modal synthesis [2, 172], as it is called, is appealing, in that the complex dynamic behaviour of a vibrating object may be decomposed into contributions from a set of modes (the spatial forms of which are eigenfunctions of the particular problem at hand, and are dependent on boundary conditions). Each such mode oscillates at a single complex frequency. (Generally, for real-valued problems, these complex frequencies will occur in complex conjugate pairs, and the "mode" may be considered to be the pair of such eigenfunctions and frequencies.) Considering the particular significance of sinusoids in human audio perception, such a decomposition can lead to useful insights, especially in terms of sound synthesis. Modal synthesis forms the basis of the popular MOSAIC [172] and Modalys [79] sound synthesis software packages (among others [35]), and, along with CORDIS, was one of the first such comprehensive systems to make use of physical modeling principles. More recently, various researchers, primarily Rabenstein have developed a related method, called the functional transformation method (FTM) [257] which uses modal techniques to derive point-to-point transfer functions. Sound synthesis applications of FTM are under development. Independently, Hélie and his associates at IRCAM have developed a formalism suitable for broad nonlinear generalizations of modal synthesis, based around the use of Volterra series approximations [218]. Such methods include FTM as a special case. A finite element based approach to modal synthesis has been discussed recently by Bruyns [43]. An interesting general viewpoint on the relationship between time and frequency domain methods is given by Rocchesso [208].

A physical model of a musical instrument, such as a vibrating string or membrane, may be described in terms of two sets of data: 1) the PDE description itself, including all information about material properties and geometry, and associated boundary conditions, and 2) excitation information, including initial conditions and/or an excitation function and location, and readout location(s). The basic modal synthesis strategy is as outlined in Figure 1.7. The first set of information is used, in an initial offline step, to determine modal shapes and frequencies of vibration; this involves, essentially, the solution of an eigenvalue problem, and may be performed in a variety of ways. (In the functional transformation approach, this is referred to as the solution of a Sturm-Liouville problem [257]). Generally, this information must be stored, the modal shapes themselves in a so-called shape-matrix. Then, the second set of information is employed: the initial conditions and/or excitation are expanded onto the set of modal functions (which under some conditions form an orthogonal set) through an inner product, giving a set of weighting coefficients. The weighted combination of modal functions then evolves, each at its own natural frequency. In order to obtain a sound output at a given time, the modal functions are projected (again through inner products) onto an observation

state, which, in the simplest case, is of the form of a delta function at a given location on the object.

Though modal synthesis had been called a “frequency domain” method here, this is not quite a correct description of the workings of a modal synthesis algorithm, and is worth clarifying. In particular, temporal Fourier transforms are not employed, and the output waveform is generated directly in the time domain. Essentially, the behaviour of each mode is described by a scalar second-order ordinary differential equation, and various time-integration techniques (some of which will be described in Chapter 3) may be employed to obtain a numerical solution. In short, it is perhaps better to think of modal synthesis not as a frequency domain method, but rather a numerical method for a linear problem which has been diagonalized (to borrow a term from state space analysis [71]). As such, in contrast with a direct time domain approach, the state itself is not observable directly, except through reversal of the diagonalization process (i.e., the projection operation mentioned above). This lack of direct observability has a number of implications in terms of multiple channel output, time-variation of excitation and readout locations, and, most importantly, memory usage. Modal synthesis continues to develop, with newer research directions focussing on the problem of interaction with nonlinear point excitation mechanisms.

Modal synthesis techniques will be discussed at various points in this book, in a general way towards the end of this chapter, and in full technical detail in Chapters 6 and 12.

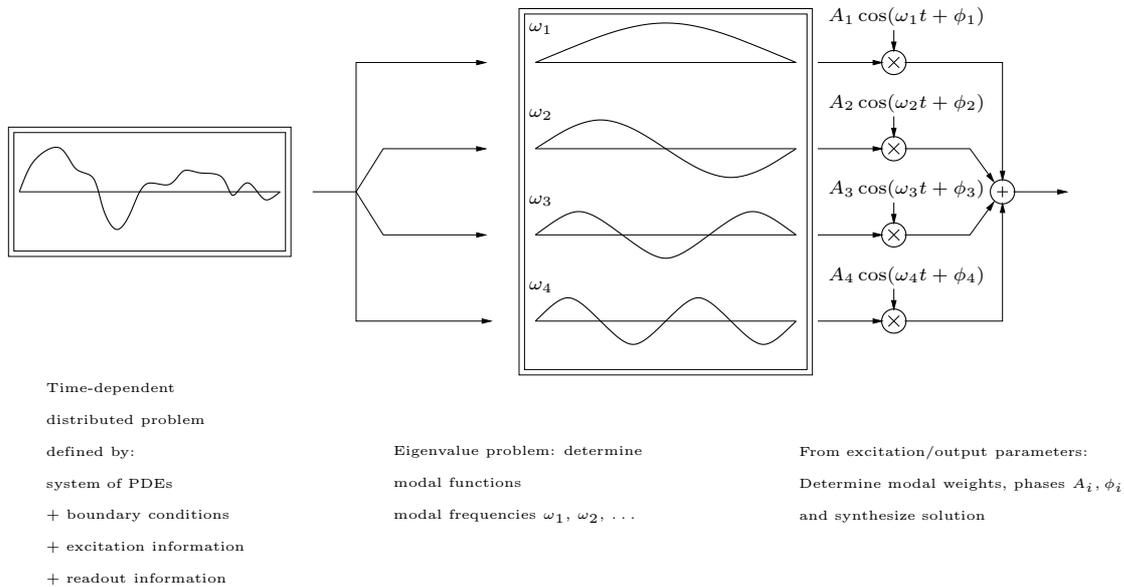


Figure 1.7: *Modal synthesis: The behaviour of a linear, distributed time-dependent problem can be decomposed into contributions from various modes, each of which possesses a particular vibrating frequency. Sound output may be obtained through a precise recombination of such frequencies, depending on excitation and output parameters.*

1.2.3 Digital Waveguides

Physical modeling sound synthesis is, to say the least, computationally very intensive. Compared with earlier methods, and in particular FM synthesis, which requires only a handful of operations per clock cycle, physical modeling methods may need to make use of hundreds or thousands of such operations per sample period in order to create reasonably complex musical timbres. Physical modeling sound synthesis, twenty years ago, was a distinctly offline activity.

In the mid 1980s, however, with the advent of digital waveguide methods [242] due to Julius Smith, all this changed. These algorithms, with their roots in digital filter design and scattering theory, and closely allied to wave digital filters [88], offered a convenient solution to the problem of computational expense for a certain class of musical instrument, in particular those whose vibrating parts can be modelled as one-dimensional linear media described, to a first approximation, by the wave equation. Among these may be included many stringed instruments, as well as most woodwind and brass instruments. In essence, the idea is very simple: the motion of such a medium may be modelled as two travelling non-interacting waves, and in the digital simulation, this is dealt with elegantly by using two “directional” delay lines, which require no computer arithmetic at all! Digital waveguide techniques have formed the basis for at least one commercial synthesizer (the Yamaha VL1), and serve as modular components in many of the increasingly common software synthesis packages (such as Max/MSP [288], STK [65], and Csound [39]). Now, some twenty years on, they are considered the state of the art in physical modelling synthesis, and the basic design has been complemented by a great number of variations intended to deal with more realistic effects (discussed below), usually through more advanced digital filtering blocks. In general, digital waveguides will not be covered in this book, mainly because there already exists a large literature on this topic, as well as a comprehensive, and perpetually growing monograph by Smith himself [242]. The relationship between digital waveguides and more standard time domain numerical methods has been addressed by various authors [241, 135, 24], and will be revisited in some detail in §6.2.11. A succinct overview is given in [238] and [206].

The path to the invention of digital waveguides is an interesting one, and is worth elaborating here. In approximately 1983, Karplus and Strong [138] developed an efficient algorithm for generating musical tones strongly resembling string tones, which was almost immediately noticed and subsequently extended by Jaffe and Smith [124]. The Karplus Strong structure is no more than a delay line, or wavetable, in a feedback configuration, in which data is recirculated; generally, the delay line is initialized with random numbers, and is terminated with a low order digital filter, usually with a low-pass characteristic—see Figure 1.8. Tones produced in this way are spectrally rich, and exhibit a decay which is indeed characteristic of plucked string tones, due to the terminating filter. The pitch is determined by the delay-line length and the sample rate: generally, for an N -sample delay line, as pictured in Figure 1.8, with an audio sample rate of f_s Hz, the pitch of the tone produced will be at f_s/N , though this may be modified through interpolation, just as in the case of wavetable synthesis. In all, the only operations required in a computer implementation are the digital filter additions and multiplications, and the shifting of data in the delay line. The computational cost is on the order of that of a single oscillator, yet instead of producing a single frequency, Karplus-Strong yields an entire harmonic series. The Karplus-Strong plucked string synthesis algorithm is an abstract synthesis technique, in that in its original formulation, though the sounds produced resembled those of plucked strings, there was no immediate physical interpretation offered.

There are two important conceptual steps leading from the Karplus-Strong algorithm to a digital waveguide structure. The first is to associate a spatial position with the values in the wavetable—in

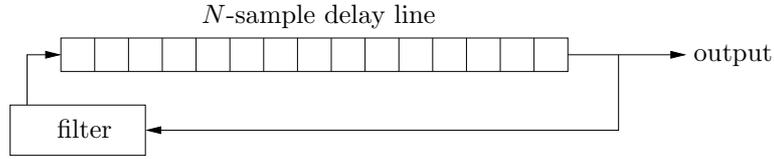


Figure 1.8: *The Karplus-Strong plucked string synthesis algorithm. An N -sample delay line is initialized with random values, which are allowed to recirculate, while undergoing a filtering operation.*

other words, a wavetable has a given physical length. The other is to show that the values propagated in the delay lines behave as individual traveling wave solutions to the 1D wave equation; only their sum is a physical variable (such as displacement, or pressure, etc.). The link between the Karplus-Strong algorithm and digital waveguide synthesis, especially in the “single-delay-loop” form, is elaborated by Karjalainen et al. [137]. Excitation elements, such as bows, hammer interactions, reeds, etc., are usually modelled as lumped, and are connected to waveguides via scattering junctions, which are, essentially, power-conserving matrix operations (more will be said about scattering methods in the next section). The details of the scattering operation will be very briefly covered here in §3.3.3. These were the two steps taken initially by Smith in work on bowed strings and reed instruments [235], though it is important to note the link with earlier work by McIntyre and Woodhouse [169], McIntyre, Schumacher and Woodhouse [168], which was also concerned with efficient synthesis algorithms for these same systems, though without an explicit use of delay line structures.

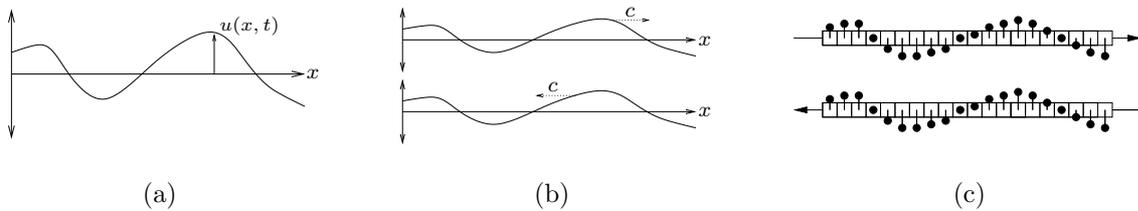


Figure 1.9: *The solution to the 1D wave equation, (a), may be decomposed into a pair of traveling wave solutions, which move to the left and right at a constant speed c determined by the system under consideration. This constant speed of propagation leads immediately to a discrete time implementation employing delay lines, as shown in (c).*

Waveguide models have been successfully to a multitude of systems; several representative configurations are shown in Figure 1.10.

String vibration has seen perhaps the most activity, probably owing to the relationship between waveguides and the Karplus-Strong algorithm. As shown in Figure 1.10(a), the basic picture is of a pair of waveguides separated by a scattering junction connecting to an excitation mechanism, such as a hammer or plectrum; at either end, the structure is terminated by digital filters which

model boundary terminations, or potentially coupling to a resonator or other strings. The output is generally read from a point along the waveguide, through a sum of wave variables traveling in opposite directions. Early work was due to Smith [241] and others. In recent years, the Acoustics group at the Helsinki University of Technology has systematically tackled a large variety of stringed instruments using digital waveguides, yielding sound synthesis of extremely high quality. Some of the target instruments have been standard keyboard instruments such as the harpsichord [268] and clavichord [266], but more exotic instruments, such as the Finnish kantele [82, 188], have been approached as well. There has also been a good deal of work on the extension of digital waveguides to deal with the particular “tension-modulation,” or pitch-glide nonlinearity in string vibration [269, 81, 255], a topic which will be taken up in great detail in §8.1. Some more related recent areas of activity have included banded waveguides [83, 84], which are designed to deal with systems with a high degree of inharmonicity, commuted synthesis techniques [239, 136], which allow for the interconnection of string models with harder-to-model resonators, through the introduction of sampled impulse responses, and the association of digital waveguide methods with underlying PDE models of strings [19].

Woodwind and brass instruments are also well-modelled by digital waveguides; a typical waveguide configuration is shown in Figure 1.10(b), where a digital waveguide is broken up by scattering junctions connected to models of (in the case of woodwind instruments) toneholes. At one end, the waveguide is connected to an excitation mechanism, such as a lip or reed model, and at the other end, output is taken after processing by a filter representing bell and radiation effects. Early work was carried out by Smith, for reed instruments [235], and for brass instruments by Cook [61]. Work on tone hole modeling has appeared [226, 78, 276], sometimes involving wave digital filter implementations [277], and efficient digital waveguide models for conical bores have also been developed [237, 264].

Vocal tract modeling using digital waveguides was first approached by Cook [60, 62]; see Figure 1.10(c). Here, due to the spatial variation of the cross-sectional area of the vocal tract, multiple waveguide segments, separated by scattering junctions, are necessary. The model is driven at one end by a glottal model, and output is taken from the other end after filtering to simulate radiation effects. Such a model is reminiscent of the Kelly-Lochbaum speech synthesis model [141], which in fact predates the appearance of digital waveguides altogether, and can be calibrated using linear predictive techniques [197], and wave digital speech synthesis models [245]. The Kelly Lochbaum model is presented in §9.5.4.

Networks of digital waveguides have also been used in a quasi-physical manner in order to effect artificial reverberation—in fact, this was the original application of the technique [234]. In this case, a collection of waveguides, of varying impedances and delay lengths is used; such a network is shown in Figure 1.10(d). Such networks are passive, so that signal energy injected into the network from a dry source signal will produce an output whose amplitude will gradually attenuate, with frequency-dependent decay times dependent on the delays and immittances of the various waveguides—some of the delay lengths can be interpreted as implementing ‘early reflections’[234]. Such networks provide a cheap and stable way of generating rich impulse responses. Generalizations of waveguide networks to feedback delay networks (FDNs) [209] and circulant delay networks [211] have also been explored, also with an eye towards applications in digital reverberation. When a waveguide network is constructed in a regular arrangement, in two or three spatial dimensions, it is often referred to as a waveguide mesh [272, 273, 274, 24]—see Figure 1.10(e). In 2D, such structures are may be used to model the behaviour of membranes, and in 3D, potentially for full-scale room acoustics

simulation (i.e., for artificial reverberation), though real-time implementations of such techniques are probably decades away. Some work on the use of waveguide meshes for the calculation of room impulse responses has appeared recently [16, 176]. The waveguide mesh is briefly covered here in §11.1.6.

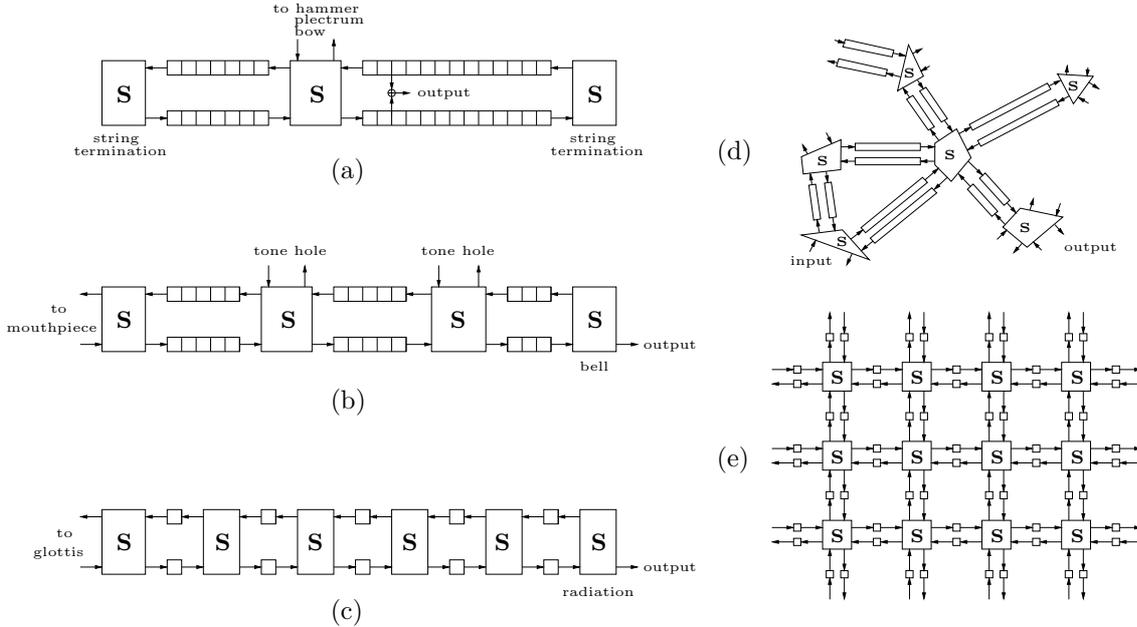


Figure 1.10: *Typical digital waveguide configurations for musical sound synthesis. In all cases, boxes marked S represent scattering operations. (a) A simple waveguide string model, involving an excitation at a point along the string and terminating filters, and output read from a point along the string length, (b) a woodwind model, with scattering at tonehole junctions, input from a reed model at the left end, and output read from the right end, (c) a similar vocal tract configuration, involving scattering at junctions between adjacent tube segments of differing cross-sectional areas, (d) an unstructured digital waveguide network, suitable for quasi-physical artificial reverberation, and (e) a regular waveguide mesh, modelling wave propagation in a 2D structure such as a membrane.*

1.2.4 Hybrid Methods

Digital waveguides are but one example of a scattering-based numerical method [24], for which the underlying variables propagated are of wave type, which are reflected and transmitted throughout a network by power-conserving scattering junctions (which can be viewed, under some conditions, as orthogonal matrix transformations). Such methods have appeared in various guises across a wide range of (often non-musical) disciplines. Perhaps the best known is the transmission-line matrix method [57, 123], which is popular in the field of electromagnetic field simulation, and dates back to the early 1970s [127], but multidimensional extensions of wave digital filters [88, 87] intended for numerical simulation have also been proposed [92, 24]. Most such methods are designed based on electrical circuit network models, and make use of scattering concepts borrowed from microwave filter design [17]; their earliest roots are in the work of Kron in the 1940s [149].

Scattering-based methods also appear in standard areas of signal processing, such as inverse estimation [42], fast factorization and inversion of structured matrices [132], and linear prediction [197] for speech signals (leading directly to the Kelly-Lochbaum speech synthesis model, which is a direct antecedent to digital waveguide synthesis).

In the musical sound synthesis community, scattering methods, employing wave (sometimes called “W”) variables are sometimes viewed [38] in opposition to methods which employ physical (correspondingly called “K,” for Kirchhoff) variables, such as lumped networks, and, as will be mentioned shortly, direct simulation techniques, which are employed in the vast majority of simulation applications in the mainstream world.

In recent years, moves have been made towards modularizing physical modeling [267]; instead of simulating the behaviour of a single musical object, such as a string or tube, the idea is to allow the user to interconnect various predefined objects in any way imaginable. In many respects, this is the same point of view as that of those working on lumped network models—this is reflected by the use of hybrid or “mixed” K-W methods, i.e., methods employing both scattering methods, such as wave digital filters and digital waveguides, and finite difference modules (typically lumped) [135, 134]. See Figure 1.11. In some situations, particularly those involving the interconnection of physical “modules,” representing various separate portions of a whole instrument, the wave formulation may be preferable, in that there is a clear means of dealing with the problem of non-computability, or delay-free loops—the concept of the reflection-free wave port, introduced by Fettweis long ago in the context of digital filter design [91], can be fruitfully employed in this case. The automatic generation of recursive structures, built around the use of wave digital filters, is a key component of such methods [187], and can be problematic when multiple nonlinearities are present, requiring specialized design procedures [223]. One result of this work has been a modular software system for physical modeling sound synthesis, incorporating elements of both types, called BlockCompiler [133]. More recently the scope of such methods has been hybridized even further through the incorporation of functional transformation (modal) methods into the same framework [189, 196].

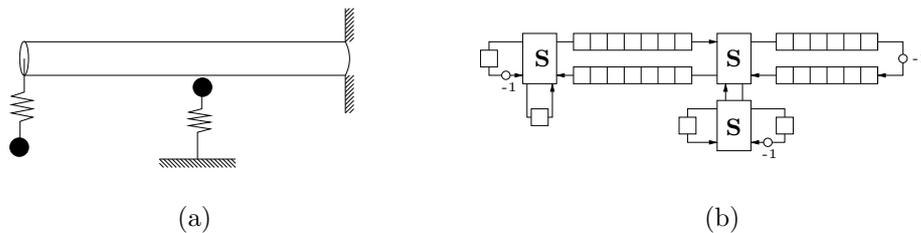


Figure 1.11: (a) A distributed system, such as a string, connected with various lumped elements, and (b) a corresponding discrete scattering network. Boxes marked **S** indicate a scattering operation.

1.2.5 Direct Numerical Simulation

Digital waveguides and related scattering methods, as well as modal techniques have undeniably become a very popular means of designing physical modeling sound synthesis algorithms. There are several reasons for this, but the main one is that such structures, built from delay lines, digital filters and Fourier decompositions, fit naturally into the framework of digital signal processing, and

form a natural extension of more abstract techniques from the pre-physical modeling synthesis era—note, for instance, the direct link between modal synthesis and additive synthesis, as well as that between digital waveguides and wavetable synthesis, via the Karplus Strong algorithm. Such a body of techniques, with linear system theory at its heart, is home turf to the trained audio engineer. See §1.3.1 for more comments on the relationship between abstract methods and physical modeling sound synthesis.

For some time, however, a separate body of work in the simulation of musical instruments has grown; this work, more often than not, has been carried out by musical acousticians whose primary interest is not so much synthesis, but rather the pure study of the behaviour of musical instruments, often with an eye towards comparison between a model equation and measured data, and possibly potential applications towards improved instrument design. The techniques used by such researchers are of a very different origin, and are couched in a distinct language; as will be seen throughout the rest of this book, however, there is no shortage of links to be made with more standard physical modeling sound synthesis techniques, provided one takes the time to “translate” between the sets of terminology! In this case, one speaks of time-stepping, and grid resolution; there is no reference to delays or digital filters, and sometimes, the frequency domain is not invoked at all, which is unheard of in the more standard physical modeling sound synthesis setting.

Perhaps the most straightforward approach makes use of a *finite difference approximation* to a set of partial differential equations [244, 113, 201], which serves as a mathematical model of a musical instrument. (When applied to dynamic, or time-dependent systems, such techniques are sometimes referred to as “finite difference time domain” (FDTD) methods, a terminology which originated in numerical methods for the simulation of electromagnetics [249, 286, 250].) Such methods have a very long history in applied mathematics, which can be traced back at least as far as the work of Courant, Friedrichs and Lewy in 1928 [68], especially as applied to the simulation of fluid dynamics [121] and electromagnetics [249]. Needless to say, the literature on finite difference methods is vast. As mentioned above, they have been applied for some time for sound synthesis purposes, though definitely without the success or widespread acceptance of methods such as digital waveguides, primarily because of computational cost—or, rather, preconceived notions about computational cost—relative to other methods.

The procedure, which is similar across all types of systems, is very simply described: the spatial domain of a continuous system, described by some model PDE, is restricted to a grid composed of a finite set of points (see Figure 1.12), at which values of a numerical solution are computed. Time is similarly discretized, and the numerical solution is advanced, through a recursion derived from the model PDE. Derivatives are approximated by differences among values at nearby grid points. The great advantage of finite difference methods (among other time domain techniques), in comparison with all the other methods discussed here, is their generality and simplicity, and the wide range of systems to which they may be applied, including strongly nonlinear distributed systems; these can not be approached using waveguides or modal synthesis, and by lumped models only in a very ad hoc and non-rigorous manner. The primary disadvantage is that one must pay great attention to the problem of numerical instability—indeed numerical stability, and the means for ensuring it in sound synthesis algorithms is one of the subjects that will be dealt with in depth in this book. Computational cost is an issue, but no more so than in any other synthesis method (with the exception of digital waveguides), and so cannot be viewed as a disadvantage of finite difference methods in particular.

The most notable early finite difference sound synthesis work was concerned with string vibration, dating back to the work of Ruiz in 1969 [220] and others [119, 10, 40]; the first truly sophisticated

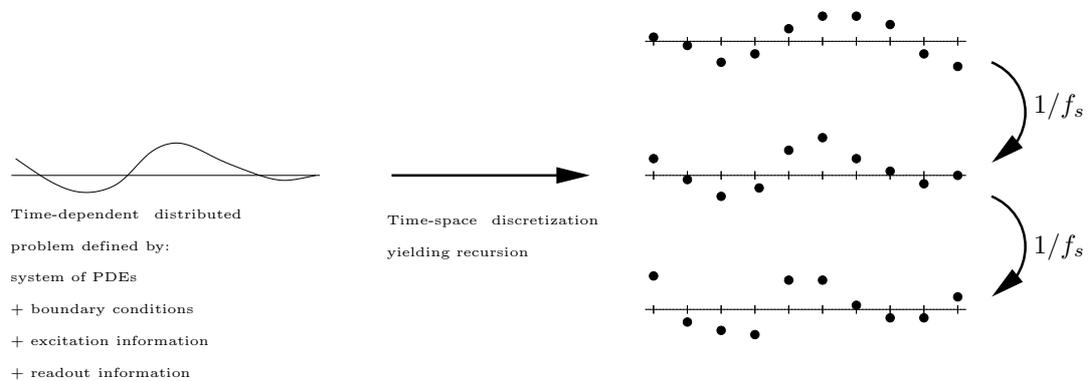


Figure 1.12: *Direct simulation via finite differences.* A distributed problem (at left) is discretized in time and space, yielding a recursion over a finite set of values (at right), to be updated with a given time step (usually corresponding to the inverse of the audio sample rate f_s).

use of finite difference methods for sound synthesis was due to Chaigne in the case of plucked string instruments [51] and piano string vibration [52, 53]; this latter work has been extended considerably by Giordano through connection to a soundboard model [106, 108]. Finite difference methods have also been applied to various percussion instruments, including those based on vibrating membranes [98] (i.e., for drum heads), such as kettledrums [199], stiff vibrating bars such as those used in xylophones [116, 77] (i.e., for xylophones), and plates [54, 227]. Finite difference schemes for nonlinear musical systems, such as strings and plates, have been treated by this author [33, 25, 26] and others [13, 12, 14]. Sophisticated difference scheme approximations to lumped nonlinearities in musical sound synthesis (particularly in the case of excitation mechanisms and contact problems) have appeared [9, 198, 229] under the remit of the Sounding Object project [210]. A useful text, which discusses finite difference methods (among other techniques) in the context of musical acoustics of wind instruments, is that of Kausel [139].

Finite difference methods, in the mainstream engineering world, are certainly the oldest method of designing a computer simulation. They are simply programmed, generally quite efficient, and there is an exhaustive literature on the subject. Best of all, in many cases they are sufficient for high-quality physical modeling sound synthesis. For the above reasons, they will form the core of this book. On the other hand, time domain simulation has undergone many developments, and some of these will be discussed in this book. Perhaps best known, particularly to mechanical engineers, are finite element methods (FEM) [86, 66] which also have long roots in simulation, but crystallized into their more modern form some time in the 1960s. The theory behind FEM is somewhat different from finite differences, in that the deflection of a vibrating object is modeled in terms of so-called shape functions, rather than in terms of values at a given set of grid points. Perhaps the biggest benefit of finite element methods is the ease with which relatively complex geometries may be modeled; this is of great interest for model validation in musical acoustics. In the end, however, the computational procedure is quite similar to that of finite difference schemes, involving a recursion of a finite set of values representing the state of the object. Finite element methods are briefly

introduced in Chapter 14. Various researchers, [11, 199] have applied finite element methods to problems in musical acoustics, though generally not for synthesis. A number of other techniques have appeared more recently, which could be used profitably for musical sound synthesis. Perhaps the most interesting are so-called spectral or pseudospectral methods [260, 100], which will be discussed in Chapter 15. Spectral methods, which may be thought of, crudely speaking, as limiting cases of finite difference schemes, allow for computation with extreme accuracy, and, like finite difference methods, are well-suited to problems in regular geometries. They have not, at the time of writing, appeared in physical modeling applications, but would appear to be a very good match—indeed, modal synthesis is an example of a very simple Fourier-based spectral method.

For linear musical systems, and some distributed nonlinear systems, finite difference schemes (among other time-domain methods) have a state space interpretation [131], which is often referred to, in the context of stability analysis, as the “matrix method” [244]. Matrix analysis/state space techniques will be discussed at various points in this book (see, e.g., §6.2.8). State-space methods have seen some application in musical sound synthesis, though not through finite difference approximations [71].

1.3 Physical Modeling: A Larger View

This is perhaps a good moment to step back and examine some general constraints on physical modeling sound synthesis, connections among the various existing methods and with earlier abstract techniques, and to address some general philosophical questions about the utility of such methods.

1.3.1 Physical Models as Descended from Abstract Synthesis

Among the most interesting observations one can make about some (but not all) physical modeling methods concerns their relationship to abstract methods, which is somewhat deeper than it might appear to be. Abstract techniques, especially those described in §1.1, set the stage for many later developments, and determined some of the basic building blocks for synthesis, as well as the accompanying notation, which is derived from digital signal processing. This influence has had its advantages and disadvantages, as will be detailed below.

As mentioned earlier, digital waveguides, certainly the most successful physical modeling technique to date, can be thought of as a physical interpretation of wavetable synthesis. Perhaps even more important than the direct association between a lossless string and a wavetable was the recognition that systems with a low degree of inharmonicity could be efficiently modelled using a pair of delay lines terminated by lumped low-order digital filters—this effectively led the way to efficient synthesis algorithms for many 1D musical systems producing pitched tones. No such efficient techniques have been reported for similar systems in the mainstream literature, and it is clear that such efficiency gains were made possible only by association with abstract synthesis methods (and digital signal processing concepts in particular), and through an appreciation of the importance of human auditory perception to the resulting sound output. On the other hand, such lumped modeling of effects such as loss and inharmonicity is also a clear departure from physicality; this is also true of newer developments such as banded waveguides and commuted synthesis.

Similarly, modal synthesis may be viewed as a direct physical interpretation of additive synthesis; a modal interpretation (like that of any physical model) has the advantage of drastically reducing the amount of control information which must be supplied. On the other hand, it is restrictive in

the sense that, with minor exceptions, it may only be applied usefully to linear and time invariant systems, which is a side-effect of a point of view informed by Fourier decomposition.

As mentioned above, there is not always a direct link between abstract and physical modeling techniques. Lumped network models and direct simulation methods, unlike the other techniques mentioned above, have distinct origins in numerical solution techniques, and not in digital signal processing. Those working on hybrid KW methods have gone a long way towards viewing such methods in terms of abstract synthesis concepts [196, 135]. Similarly, there is not a strong physical interpretation (to the knowledge of this author) of abstract techniques such as FM or granular synthesis.

1.3.2 Connections: Direct Simulation and Other Methods

Because direct simulation methods are, in fact, the subject of this book, it is worth saying a few words about the correspondence with the various other physical modeling methods discussed in the previous section. Indeed, after some exposure to these methods, it becomes clear that all can be related to one another, and to mainstream simulation methods.

Perhaps the closest relative of the direct techniques discussed here is the lumped mass-spring network methodology [44]; in some ways, this is more general than direct simulation approaches for distributed systems, in that one could design a lumped network without a distributed counterpart—this could indeed be attractive to a composer. As a numerical method however, it is designed as a large ordinary differential equation solver, which puts it in line with various simulation techniques based on semi-discretization, and in particular finite element methods. As mentioned in §1.2.1, distributed systems may be dealt with through large collections of lumped elements, and in this respect, the technique differs considerably from purely distributed models based on the direct solution of PDEs, because it can be quite cumbersome to design more sophisticated numerical methods, and to deal with systems more complex than a simple linear string or membrane using a lumped approach. The main problem is the “local” nature of connections in such a network; in more modern simulation approaches (such as, e.g., spectral methods [260]), approximations at a given point in a distributed system are rarely modelled using nearest-neighbour connections between grid variables. From the distributed point of view, network theory may be dispensed with entirely. Still, it is possible to view the integration of lumped network systems in terms of distributed finite difference schemes—see §6.1.1 and §11.1.7 for details.

It should also come as no surprise that digital waveguide methods may also be rewritten as finite difference schemes. It is interesting that although the exact discrete traveling wave solution to the 1D wave equation has been known in the mainstream simulation literature for some time (since the 1960s at least [3]), and is a direct descendant of the method of characteristics [103], the efficiency advantage was apparently not taken advantage of to the same spectacular effect as in musical sound synthesis. (This is perhaps because the 1D wave equation is seen, in the mainstream world, as a model problem, and not of inherent practical interest.) Equivalences between finite differences and digital waveguide methods, in the 1D case and the multidimensional case of the waveguide mesh, have been established by various authors [272, 274, 242, 241, 24, 81, 225], and, as mentioned earlier, those at work on scattering based modular synthesis have incorporated ideas from finite difference schemes into their strategy [134, 135]. This correspondence will be revisited with regard to the 1D wave equation in §6.2.11, and the 2D wave equation in §11.1.6. It is worth warning the reader, at this early stage, that the efficiency advantage of the digital waveguide method with respect to an equivalent finite difference scheme does not carry over to the multidimensional case [241, 24].

Modal analysis and synthesis was in extensive use long before it appeared in musical sound synthesis applications, particularly in association with finite element analysis of vibrating structures—see [180] for an overview. In essence, a time-dependent problem, under some conditions, may be reduced to an eigenvalue, or statics problem, greatly simplifying analysis. It may also be viewed under the umbrella of more modern so-called spectral or pseudo-spectral methods [48], which predate modal synthesis by many years. Spectral methods essentially yield highly accurate numerical approximations through the use of various types of function approximations to the desired solution; many different varieties exist. If the solution is expressed in terms of trigonometric functions, the method is often referred to as a Galerkin Fourier method—this is exactly modal synthesis in the current context. Other types of spectral methods, perhaps more appropriate for sound synthesis purposes (and in particular collocation methods) will be discussed in Chapter 15. Modal synthesis methods will be discussed in more detail in §6.1.11 and §11.1.8.

Modular or “hybrid” methods, though nearly always framed in terms of the language of signal processing may also be seen as finite difference methods; the correspondence between lumped models and finite difference methods is direct, and that between wave digital filters and numerical integration formulae has been known for many years [93], and may be related directly to the even older concept of absolute-, or A-stability [105, 70]. The key feature of modularity, however, is new to this field, and is not something which has been explored in depth in the mainstream simulation community.

This is not the place to evaluate the relative merit of the various physical modeling synthesis methods; this will be performed exhaustively with regard to two useful model problems, the 1D and 2D wave equations, in Chapters 6 and 12, respectively. For the impatient reader, some concluding remarks on relative strengths and weaknesses of these methods with respect to direct simulation methods appear in Chapter 16.

1.3.3 Complexity of Musical Systems

In the physical modeling sound synthesis literature (as well as that of the mainstream) it is commonplace to see claims of better performance of a certain numerical method over another. Performance may be measured in terms of the number of floating point operations required, or memory requirements, or, more characteristically, better accuracy for a fixed operation count. It is worth keeping in mind, however, that even though these claims are (sometimes) justified, for a given system, there are certain limits as to “how fast” or “how efficient” a simulation algorithm can be. These limits are governed by system complexity; one cannot expect to reduce an operation count for a simulation below that which is required for an adequate representation of the solution.

System complexity is, of course, very difficult to define. Perhaps most amenable to discussions of complexity are linear and time-invariant (LTI) systems, which form a starting point for many models of musical instruments. Consider any lossless distributed LTI system (such as a string, bar, membrane, plate, or acoustic tube), freely vibrating at low amplitude due to some set of initial conditions, without any external excitation. Considering the continuous case, one is usually interested in reading an output $y(t)$ from a single location on the object. This solution can almost always¹ be written in the form:

$$y(t) = \sum_{q=1}^{\infty} A_q \cos(2\pi f_q t + \phi_q) \quad (1.5)$$

¹The formula must be altered slightly if the frequencies are not all distinct.

the form is exactly that of pure additive synthesis or modal synthesis; here, A_q and ϕ_q are determined by the initial conditions and constants which define the system, and the frequencies f_q are assumed non-negative, and to lie in an increasing order. Such a system has a countably infinite number of degrees of freedom; each oscillator at a given frequency f_q requires the specification of two numbers, A_q and ϕ_q .

Physical modeling algorithms generally produce sound output at a given sample rate, say f_s . This is true of all the methods discussed in the previous section. There is thus no hope of (and no need for) simulating frequency components² which lie above $f_s/2$. Thus, as a prelude to a discrete time implementation, the representation (1.5) may be truncated to

$$y(t) = \sum_{q=1}^N A_q \cos(2\pi f_q t + \phi_q) \quad (1.6)$$

where only the N frequencies f_1 to f_N are inferior to $f_s/2$. Thus the number of degrees of freedom is now finite: $2N$.

Even for a vaguely defined system such as this, from this information one may go slightly farther and calculate both the operation count and memory requirements, assuming a modal-type synthesis strategy. As described in §1.2.2, each frequency component in the expression (1.5) may be computed using a single two-pole digital oscillator, which requires two adds, one multiply, and two memory locations, giving, thus, $2N$ adds and N multiplies per time step, and a necessary $2N$ units of memory. Clearly if fewer than N oscillators are employed, the resulting simulation will not be complete, and the use of more than N oscillators is superfluous. Not surprisingly, such a measure of complexity is not restricted to frequency domain methods only; in fact, *any* method (including direct simulation methods such as finite differences and finite element methods) for computing the solution to such a system must require roughly the same amount of memory and number of operations; for time domain methods, complexity is intimately related to conditions for numerical stability. Much more will be said about this in Chapters 6 and 11, which deal with time domain and modal solutions for the wave equation.

There is, however, at least one very interesting exception to this rule. Consider the special case of a system for which the modal frequencies are multiples of a common frequency f_1 , i.e., in (1.5), $f_q = qf_1$. In this case, (1.5) is a Fourier series representation of a periodic waveform, of period $T = 1/f_1$, or, in other words,

$$y(t) = y(t - T)$$

The waveform is thus completely characterized by an single period of duration T . In a discrete setting, it is obvious that it would be wasteful to employ separate oscillators for each of the components of $y(t)$; far better would be to simply store one period of the waveform in a table, and read through it at the appropriate rate, employing simple interpolation, at a cost of $O(1)$ operations per time step instead of $O(N)$. Though this example might seem trivial, it is worth keeping in mind that many

²In the nonlinear case, however, one might argue that the use of higher sampling rates is justifiable, due to the possibility of aliasing. On the other hand, in most physical systems, loss becomes extremely large at high frequencies, so a more sound, and certainly much more computationally efficient approach is to introduce such losses into the model itself. Another argument for using an elevated sample rate, employed by many authors, is that numerical dispersion (leading to potentially audible distortion) may be reduced; this, however, is disastrous in terms of computational complexity, as the total operation count often scales with the square or cube of the sample rate. It is nearly always possible to design a scheme with much better dispersion characteristics, which still operates at a reasonable sample rate.

pitched musical sounds are approximately of this form, and in particular those produced by musical instruments based on strings and acoustic tubes. The efficiency gain noted above is at the heart of the digital waveguide synthesis technique. Unfortunately, however, for musical sounds which do not generate harmonic spectra, there does not appear to be any such efficiency gain possible; this is the case, in particular, for 2D percussion instruments, and moderately stiff strings and bars. Though extensions of digital waveguides do indeed exist in the multidimensional setting, in which case they are usually known as digital waveguide meshes), there is no efficiency gain relative to modal techniques, or standard time differencing methods; indeed, the computational cost of solution by any of these methods is roughly the same³.

For distributed nonlinear systems, such as strings and percussion instruments, it is difficult to even approach a definition of complexity—perhaps the only thing one can say is that for a given nonlinear system, which reduces to an LTI system at low vibration amplitudes (this is the usual case in most of physics, and musical acoustics in particular), the complexity, or required operation count and memory requirements for an algorithm simulating the nonlinear system will be at least that of the associated linear system. Efficiency gains through digital waveguide techniques are no longer possible, except under very restricted conditions—one of these, the string under a tension-modulated nonlinearity, will be discussed in §8.1.6.

One question that will not be approached in detail in this book is of model complexity in the perceptual sense. This is a very important issue, in that psychoacoustic criteria could lead to reductions in both the operation count and memory requirements of a synthesis algorithm, in much the same way as they have impacted on audio compression. For instance, the description of the complexity of a linear system in terms of the number of modal frequencies up to the Nyquist frequency is mathematically sound, but for many musical systems (particularly in 2D), the modal frequencies become very closely spaced in the upper range of the audio spectrum. Taking into consideration the concepts of the critical band and frequency domain masking, it may not be necessary to render the totality of the components. Such psychoacoustic model reduction techniques have been used, with great success, in many efficient (though admittedly non-physical) artificial reverberation algorithms. The impact of psychoacoustics on physical models of musical instruments has seen some investigation recently, in the case of string inharmonicity[125], and also for impact sounds [7], and it would be extremely useful to develop practical complexity-reducing principles and methods, which could be directly related to numerical techniques.

This main point of this section is to signal to the reader that for general systems, there is not a physical modeling synthesis method which acts as a magic bullet. There is a minimum price to be paid for the proper simulation of any system. For a given system, the operation counts for modal, finite difference and lumped network models are always nearly the same; in terms of memory requirements, modal synthesis methods can incur a much heavier cost than time domain methods. One great misconception which has appeared time and time again in the literature [37] is that time domain methods are wasteful, in the sense that the entire state of an object must be updated, even though one is interested, ultimately, in only a scalar output, generally from a single location on the virtual instrument. Thus point-to-point “black-box” type models, perhaps based on a transfer function representation are more efficient. But, as will be shown repeatedly throughout this book, the order of any transfer function description (and thus the memory requirements) will be roughly

³It is possible, for certain systems such as the ideal membrane, under certain conditions, to extract groups of harmonic components from a highly inharmonic spectrum, and deal with them individually using waveguides [6, 28], leading to an efficiency gain, albeit a much more modest one than in the 1D case. Such techniques, unfortunately, are rather restrictive in that only extremely regular geometries may be dealt with.

the same as the size of the physical state of the object in question.

1.3.4 Why?

The question most often asked by musicians and composers (and perhaps least often by engineers) about physical modeling sound synthesis is: *Why?* More precisely: *Why bother to simulate the behaviour of an instrument which already exists?* Surely the best that can be hoped for is an exact reproduction of the sound of an existing instrument. This is not an easy question to answer, but nonetheless, various answers do exist.

The most common answer is almost certainly: *Because it can be done.* This is a very good answer from the point of view of musical acoustician, whose interest may be to prove the validity of a model of a musical instrument, perhaps by comparing simulation results (i.e., synthesis) to measured output, or perhaps by psychoacoustic comparison of recorded and model-synthesized audio output. Beyond the academic justification, there are boundless opportunities for improvement in musical instrument design using such techniques. From a commercial point of view, too, it would be extremely attractive to have a working sound synthesis algorithm to replace sampling synthesis, which relies on a large database of recorded fragments. (Consider, for example, the number of samples that would be required to completely represent the output of an acoustic piano, with 88 notes, with 60 dB decay times on the order of tens of seconds, struck over a range of velocities and pedal configurations.) On the other hand, such an answer will satisfy neither a composer of modern electroacoustic music in search of new sounds, nor a composer of acoustic orchestral music, who will find the entire idea somewhat artificial and pointless.

Another answer, closer in spirit to the philosophy of this author, is that physical modeling sound synthesis is far more than just a means of aping sounds produced by acoustic instruments, and it is much more than merely a framework for playing mix and match with components of existing acoustic instruments (the bowed flute, the flutter-tongued piano, etc.); though interesting, this might well be an application which would appeal to engineers only. Acoustically-produced sound is definitely a conceptual point of departure for many composers of electroacoustic music, given the early body of work on rendering the output of abstract sound synthesis algorithms less synthetic-sounding [171, 228], and, more importantly, the current preoccupation with real-time transformation of natural audio input. In this latter case, though, it might well be true (and one can never really guess these things) that a composer would jump at the chance to be freed from the confines of acoustically-produced sound if indeed an alternative, possessing all the richness and interesting unpredictability of natural sound, yet somehow different, were available. This is what physical modeling is all about, at least for this author.

Chapter 2

Time Series and Difference Operators

In this short chapter, the basics of finite difference operations, as applied to time-dependent ordinary differential equations (ODEs) in the next two chapters, and subsequently to partial differential equations (PDEs) are presented. Though the material that appears here is rudimentary, and may be skipped by any reader with experience with finite difference schemes, it is advisable to devote at least a few minutes to familiarizing oneself with the notation, which is necessarily a bit of a hybrid between that used by those in the simulation field, and by audio and electrical engineers (but skewed towards the former). There are many old and venerable texts [201, 3, 233] and some more modern ones which may be of special interest to those with a background in electrical engineering or audio [244, 113, 281, 86, 263] which cover this material in considerably more detail, as well as the text of Kausel [139] which deals directly with difference methods in musical acoustics, but the focus here is on those aspects which will later pertain directly to physical modeling sound synthesis. Though the following presentation is mainly abstract and context-free, there are many comments (as well as problems and programming exercises) which relate specifically to digital audio.

The use of discrete time series, taking on values at a finite set of time instants, in order to approximate continuous processes is natural in audio applications, but its roots far predate the appearance of digital audio, and even the modern digital computer itself. Finite difference type approximations to ordinary differential equations go back a very long way, and modern methods have their roots in work from the early 20th century—see the opening pages of Ames [3] for a historical overview and references. Time series and simple difference operators are presented in §2.1 and §2.2, followed by a review of frequency domain analysis in §2.3, which includes some discussion of the z transform, and the association between difference operators and digital filter designs, which are currently the methodology of choice in musical sound synthesis. Finally, energy concepts are introduced in §2.4; these are rather nonstandard and do not appear in most introductory texts on finite difference methods. They are geared towards the construction and analysis of finite difference schemes for nonlinear systems, which are of great importance in musical acoustics and physical modeling sound synthesis, and will be heavily used in the later sections of this book.

2.1 Time Series

In a finite difference setting, continuously variable functions of t , such as $u(t)$, are approximated by time series, often indexed by integer n . For instance, the time series u_d^n represents an approximation to $u(t_n)$, where $t_n = nk$, for a time step¹ k . In audio applications, perhaps more familiar is the sampling frequency f_s is defined as

$$f_s = 1/k$$

Note here that the symbol u has been used here to denote both the continuously variable function $u(t)$ and the approximating time series u_d^n ; the “ d ” appended in the subscript for the time series stands for “discrete” and is simply a reminder of the distinction between the two quantities. In subsequent chapters, it will be dropped, in an attempt at avoiding a proliferation of notation; this ambiguity should lead to little confusion, as such forms rarely appear together in the same expression, except in the initial stages of definition of finite difference operators. The use of the same notation also helps to indicate the fundamental similarities in the bodies of analysis techniques which may be used in the discrete and continuous settings.

Before introducing these difference operators and examining discretization issues, it is worth making a few comments which relate specifically to audio. First, consider a function $u(t)$ which appears as the solution to an ODE. If some difference approximation to the ODE is derived, which generates a solution time series u_d^n , it is important to note that in all but a few pathological cases, u_d^n is *not* simply a sampled version of the true solution, i.e.,

$$u_d^n \neq u(nk)$$

Though obvious, it is especially important for those with an electrical or audio engineering background (i.e., those accustomed to dealing with sampled data systems) to be aware of this at the most subconscious level, so as to avoid arriving at false conclusions based on familiar results such as, e.g., the Shannon sampling theorem [183]. In fact, one can indeed incorporate such results into the simulation setting, but in a manner which may be counterintuitive (see §3.2.4). In sum, it is best to remember that in the strict physical modeling sound synthesis framework, there occurs no sampling of recorded audio material (though in practice, and particularly in commercial applications, there are many exceptions to this rule). Second, in audio applications, as opposed to standard simulation in other domains, the sample rate f_s and thus the time step k are generally set before run time, and are not varied; in audio, in fact, one nearly always takes f_s as constant, not merely over the duration of a single run of a synthesis algorithm, but over all applications (most often it is set to 44.1 kHz, sometimes to 32 kHz or 48 kHz). This, in contrast to the first comment above, is intuitive for audio engineers, but not for those involved with numerical simulation in other areas, who often are interested in developing numerical schemes which allow a larger time step with little degradation in accuracy. Though the benefits of such schemes may be interpreted in terms of numerical dispersion (see §6.2.3), in an audio synthesis application, there is no point in developing a scheme which runs with increased efficiency at a larger time step (i.e., at a lower sampling rate), as such a scheme will be incapable of producing potentially audible frequencies in the upper range of human hearing. A third major distinction is that the duration of a simulation, in sound synthesis applications, is extremely long by most simulation standards (on the order of hundreds of thousands, or millions of

¹Though k has been chosen as the symbol representing the time step in this book, the same quantity goes by a variety of different names in the various sectors of the simulation literature, including T , Δt , h_t , etc.

time steps). A variety of techniques which are commonly used in mainstream simulation can lead to audible distortion over such long durations. As an example, the introduction of so-called artificial viscosity into a numerical scheme in order to reduce spurious oscillations will result in long-time solution decay, which will have an impact on the global envelope of the resulting sound output. Fourth, and finally, due to the nature of the system of human aural perception, synthesis output is always scalar—that is, it can be represented by a single time series, or, in the multichannel case, a small number of such series, which is not the case in other applications. There are thus opportunities for algorithmic simplification, with digital waveguides as a supremely successful example. Again, the perceptual considerations listed above are all peculiar to digital audio.

2.2 Shift, Difference and Averaging Operators

In time domain simulation applications, just as in digital filtering, the fundamental operations which may be applied to a time series u_d^n are shifts. The forward and backward shifts, and the identity operation “1” are defined as

$$e_{t+}u_d^n = u_d^{n+1} \quad e_{t-}u_d^n = u_d^{n-1} \quad 1u_d^n = u_d^n$$

and are to be regarded as applying to the time series u_d^n at all values of the index n . The identity operator behaves as a simple scalar multiplication by unity; multiples of the identity behave accordingly, and will be indicated by multiplicative factors, such as “2” or “ α ,” where α is a real constant. A set of useful difference and averaging operations may be derived from these elementary shifts. For example, various approximations to the first derivative operator (the nature of this approximation will be explained shortly) may be given as

$$\delta_{t+} \triangleq \frac{1}{k}(e_{t+} - 1) \quad \cong \quad \frac{d}{dt} \quad (2.1a)$$

$$\delta_{t-} \triangleq \frac{1}{k}(1 - e_{t-}) \quad \cong \quad \frac{d}{dt} \quad (2.1b)$$

$$\delta_t \triangleq \frac{1}{2k}(e_{t+} - e_{t-}) \quad \cong \quad \frac{d}{dt} \quad (2.1c)$$

These are often called forward, backward, and centered difference approximations, respectively. The behaviour of any such operator is most easily understood by expanding its action onto a time series u_d^n , where the time index n is made explicit. For the operators defined above, for example, one has

$$\delta_{t+}u_d^n = \frac{1}{k}(u_d^{n+1} - u_d^n) \quad \delta_{t-}u_d^n = \frac{1}{k}(u_d^n - u_d^{n-1}) \quad \delta_t u_d^n = \frac{1}{2k}(u_d^{n+1} - u_d^{n-1}) \quad (2.2)$$

Also useful, especially in the construction of so-called implicit schemes (which will be touched upon briefly with regard to the oscillator in §3.3 and in much more detail in the distributed setting

in §6.3 and subsequently) and in energetic analysis (see §2.4) are various averaging operators:

$$\mu_{t+} \triangleq \frac{1}{2}(e_{t+} + 1) \quad \cong \quad 1 \quad (2.3a)$$

$$\mu_{t-} \triangleq \frac{1}{2}(1 + e_{t-}) \quad \cong \quad 1 \quad (2.3b)$$

$$\mu_t \triangleq \frac{1}{2}(e_{t+} + e_{t-}) \quad \cong \quad 1 \quad (2.3c)$$

All of these averaging operators are approximations to the identity operation. (One might wonder why one would introduce an approximation to the continuous time identity operation, which, after all, may be perfectly approximated through the identity in discrete time. The answer comes when examining finite difference schemes in their entirety; in many cases, the accuracy of a scheme involves the counterbalancing of the effects of various operators, not just one in isolation. See, e.g., §3.3.4 and §6.2.4)

To avoid doubling of terminology, averaging and difference operators will be referred to as “difference” operators in this book, although “discrete” would probably be a better term. It should be clear that all the operators defined in (2.1) and (2.3) commute with one another individually. The Greek letters δ and μ are mnemonics for “difference” and “mean,” respectively.

When combined, members of the small set of “canonical” simple difference operators in (2.1) and (2.3) above can yield almost any imaginable type of approximation or difference scheme for an ODE or system of ODEs. As an important example, an approximation to the second derivative may be obtained through the composition of the operators δ_{t+} and δ_{t-} :

$$\delta_{tt} \triangleq \delta_{t+}\delta_{t-} = \frac{1}{k^2}(e_{t+} - 2 + e_{t-}) \approx \frac{d^2}{dt^2} \quad (2.4)$$

Again, the constant “2” is to be thought of as twice the identity operation, under the application of δ_{tt} to a grid function.

In musical acoustics, the appearance of time derivatives of order higher than two is extremely rare (one case is that of higher-order models of beam and plate vibration [111], another being that of frequency dependent loss in some models of string vibration [52, 220]), and for this reason, difference approximations to higher time derivatives will not be discussed further in any detail in this book. See Problem 2.2. Approximations to higher spatial derivatives, however, play a central role in various models of bar and plate vibration, but a treatment of the related difference operators will be postponed until Chapter 5.

2.2.1 Temporal Width of Difference Operators

Though the property of width, or stencil width of a difference operator is usually discussed with reference to spatial difference operators, this is good place to introduce the concept. The temporal width of an operator, such as any of those defined at the beginning of §2.2, is defined as the number of distinct time steps (or levels) required to form the approximation. For example, the operator δ_{t+} , as defined in (2.1a), when expanded out as in (2.2), clearly requires two adjacent values of the time series u_d in order to be evaluated. The same is true of the operator δ_{t-} , as well as the averaging operators μ_{t+} and μ_{t-} ; all such operators are thus of width two. The operators δ_t and μ_t , as well as δ_{tt} will be of width three. See Figure 2.1. In general, greater accuracy (to be discussed in §2.2.3)

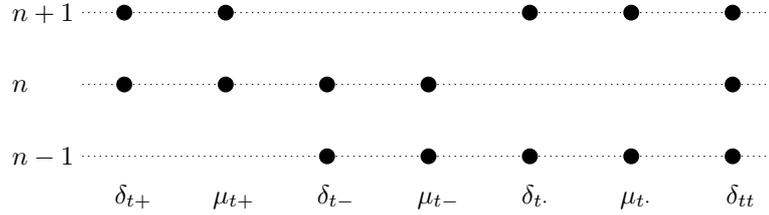


Figure 2.1: Illustration of temporal widths for various operators (as indicated above), when operating on a grid function at time step n .

is obtained at the expense of greater operator width, which can complicate an implementation in various ways. For time difference operators, there will be additional concerns with initialization of difference schemes, as well as the potential appearance of parasitic solutions, and stability analysis can become enormously complex. For spatial difference operators, finding appropriate numerical boundary conditions becomes more difficult. When accuracy is not a huge concern, it is often wise to stick with simple, low-width operators. In audio synthesis, the importance of greater accuracy may be evaluated with respect to psychoacoustic criteria—this issue will be broached in more detail in §2.3.4, and in §3.2.4.

2.2.2 Combining Difference Operators

For any collection of difference operators, each of which approximates the same continuous operator, any linear combination will approximate the same operator to within a multiplicative factor. For instance, one may form a difference operator approximating a first derivative by

$$\alpha\delta_{t+} + (1 - \alpha)\delta_{t-}$$

for any α (generally constrained to be a real number). In this case, the operator δ_t , defined in (2.1c), can be obtained with $\alpha = 1/2$.

Similarly, for averaging operators, one may form the combination

$$\alpha\mu_{t+} + \beta\mu_{t-} + (1 - \alpha - \beta)\mu_t.$$

for any scalar α and β (again generally constrained to be real).

Difference operators may also be combined by composition (operator multiplication) as well, as was seen in the definition of the second difference δ_{tt} , in (2.4) which can be viewed as the composition of the operators δ_{t+} and δ_{t-} . It is easy to show that any composition of averaging operators, such as

$$\mu_{t+}\mu_{t-}, \quad \mu_{t+}\mu_{t-}\mu_t, \quad \mu_{t-}\mu_{t-}\mu_t\mu_t, \quad \dots$$

is itself an averaging operator. See Problem 2.6. The first such combination above is useful enough to warrant the use of a special symbol:

$$\mu_{tt} \triangleq \mu_{t+}\mu_{t-} \tag{2.5}$$

Any composition of averaging operators, which is itself composed with a single first difference operator, such as

$$\mu_{t+}\delta_{t-}, \quad \mu_{t+}\mu_t\delta_t, \quad \mu_{t+}\mu_t-\mu_t\delta_{t+}, \quad \dots$$

itself behaves as an approximation to a first derivative. See Problem 2.7. In general, the composition of difference operators tends to increase temporal width.

2.2.3 Taylor Series and Accuracy

The interpretation of the various operators defined in the previous section as approximations to differential operators is direct—indeed, the definitions of the first differences δ_{t+} and δ_{t-} are in fact none other than the approximate forms (right-hand limit and left-hand limit) from which the classical definition of the derivative follows, in the limit as k approaches 0. It is useful, however, to be slightly more precise about the accuracy of these approximations, especially from the point of view of sound synthesis.

A good starting point is in the standard comparison between the behaviour of difference operators and differential operators as applied to continuous time functions, through simple Taylor series analysis. In a slight abuse of notation, one may apply such difference operators to continuous time functions as well as to time series. For instance, for the forward time difference operator δ_{t+} applied to the function u at time t , one may write

$$\delta_{t+}u(t) = \frac{1}{k} (u(t+k) - u(t))$$

Assuming $u(t)$ to be infinitely differentiable, and expanding $u(t+k)$ in Taylor series about t , one then has

$$\delta_{t+}u(t) = \frac{du}{dt} + \frac{k}{2} \frac{d^2u}{dt^2} + \dots$$

where \dots refers to terms which depend on higher derivatives of u , and which are accompanied by higher powers of k , the time step. The operator δ_{t+} thus approximates the first time derivative to an *accuracy* which depends on the first power of k , the time step; as k is made small, the difference approximation approaches the exact value of the derivative with an error proportional to k . Such an approximation is thus often called first-order accurate. (This can be slightly misleading in the case of operators acting in isolation, as the order of accuracy is dependent on the point at which the Taylor series is centered. See Problem 2.1. Such ambiguity is removed when finite difference schemes are examined in totality.) The backward difference operator is, similarly, a first-order accurate approximation to the first time derivative.

The centered difference operator, as might be expected, is more accurate. One may write, employing Taylor series centered about t ,

$$\delta_t.u(t) = \frac{1}{2k} (u(t+k) - u(t-k)) = \frac{du}{dt} + \frac{k^2}{6} \frac{d^3u}{dt^3} + \dots$$

which illustrates that the centered approximation is accurate to second order (i.e., the error depends on the second power of k , the time step). Notice, however, that the width of the centered operator is three, as opposed to two for the forward and backward difference operators—in general, as mentioned in §2.2.1, the better the accuracy of the approximation, the more adjacent values of the time series will be required. This leads to the usual tradeoff between performance and latency that one sees in,

e.g., digital filters. Fortunately, in physical modeling sound synthesis applications, due to perceptual considerations, it is probably true that only rarely will one need to make use of highly accurate operators. This issue will be discussed in a more precise way in §3.3.2, in the context of the simple harmonic oscillator.

The orders of accuracy of the various averaging operators may also be demonstrated in the same manner, and one has

$$\mu_{t+} = 1 + O(k) \quad \mu_{t-} = 1 + O(k) \quad \mu_{t.} = 1 + O(k^2)$$

where again, “1” refers to the identity operation, and $O()$ signifies “order of.” Similarly, the approximation to the second derivative δ_{tt} is second order accurate; it will be useful later to have the full Taylor series expansion for this operator:

$$\delta_{tt} = \sum_{l=1}^{\infty} \frac{2k^{2(l-1)}}{(2l)!} \frac{d^{2l}}{dt^{2l}} = \frac{d^2}{dt^2} + O(k^2) \quad (2.6)$$

It is straightforward to arrive at difference and averaging approximations which are accurate to higher order. This is a subject which was explored extensively early on in the literature—see Problem 2.4 for an example. As mentioned previously, higher-order accurate difference operators are more difficult to employ with regard to time rather than space differentiation due to stability considerations.

It is important to keep in mind that though these discussions of accuracy of difference and averaging operators have employed continuous time functions, the operators themselves will be applied to time series; in a sense, the analysis here is incomplete until an entire equation (i.e., an ODE) has been discretized, at which point one may determine the accuracy of the approximate solution computed using a difference scheme with respect to the true solution to the ODE. As a rule of thumb, the accuracy of an operator acting in isolation is indeterminate—in this section, it has been taken to refer to the order of the error term when an expansion is taken about the “reference” time instant $t = nk$.

2.2.4 Identities

Various equivalences exist among the various operators defined in this section. Here are a few of interest:

$$\mu_{t.} = 1 + \frac{k^2}{2} \delta_{tt} \quad (2.7a)$$

$$\delta_{t.} = \delta_{t+} \mu_{t-} = \delta_{t-} \mu_{t+} \quad (2.7b)$$

$$\delta_{tt} = \frac{1}{k} (\delta_{t+} - \delta_{t-}) \quad (2.7c)$$

$$1 = \mu_{t\pm} \mp \frac{k}{2} \delta_{t\pm} \quad (2.7d)$$

$$e_{t\pm} = \mu_{t\pm} \pm \frac{k}{2} \delta_{t\pm} \quad (2.7e)$$

$$e_{t\pm} = 1 \pm k \delta_{t\pm} \quad (2.7f)$$

2.3 Frequency Domain Analysis

In this section, a very brief introduction to frequency domain analysis of finite difference operators is presented. As may be expected, such techniques can allow one to glean much perceptually important information from the model system, and also to compare the performance of a given difference scheme to the model system in an intuitive manner. It also allows a convenient means of analyzing numerical stability, if the model problem is linear and time invariant; this is indeed the case for many simplified systems in musical acoustics. For the same reason, however, one must be wary of any attempts to make use of such techniques in a nonlinear setting, though one can indeed come to some (generally qualitative) conclusions under such conditions.

2.3.1 Laplace and z transforms

For a continuously variable function $u(t)$, one definition of the Laplace transform $\hat{u}(s)$ is as follows:

$$\hat{u}(s) = \int_{-\infty}^{\infty} u(t)e^{-st} dt \quad (2.8)$$

where $s = j\omega + \sigma$ is a complex-valued frequency variable. In this case, the definition is two-sided, allowing one to ignore initial conditions. The transformation may be abbreviated as

$$u(t) \xLeftrightarrow{\mathcal{L}} \hat{u}(s)$$

If the transformation may be restricted to $s = j\omega$, then the Laplace transform reduces to a Fourier transform.

The definition of the Laplace transform above is two-sided or bilateral, and useful in steady state applications—in many parts of the literature, however, a one-sided definition is employed, allowing initial conditions to be directly incorporated into the resulting frequency domain analysis:

$$\hat{u}(s) = \int_0^{\infty} u(t)e^{-st} dt \quad (2.9)$$

In general, for the analysis of well-posedness of differential equations and numerical stability, a two-sided definition may be used. Though it might be tempting to make use of Fourier transforms in this case, it is important to retain the complex frequency variable s in order to simplify the analysis of loss.

For a time series u_d^n , the z -transform $\hat{u}_d(z)$, again two-sided, may be defined as

$$\hat{u}_d(z) = \sum_{n=-\infty}^{\infty} z^{-n} u_d^n$$

where $z = e^{sk}$ is the discrete frequency variable, again with $s = j\omega + \sigma$, and where the superscript of z now indicates a power. The z transform may be abbreviated as

$$u_d^n \xLeftrightarrow{\mathcal{Z}} \hat{u}_d(z)$$

Again, as for the case of the Laplace transform, a one-sided definition could be employed. The

symbol z is often written as g , and called the *amplification factor* in the analysis of finite difference recursions). If z is restricted to $z = e^{j\omega k}$, the z transformation becomes a discrete-time Fourier transformation.

The Laplace and z -transforms are covered in great detail in hundreds of other texts, and some familiarity with them is assumed—see, e.g., [197] or [192] for more information. In particular, the subject of inverse Laplace and z transformations, though important in its own right, is not covered here, as it seldom appears in practical analysis and design of finite difference schemes.

Frequency Domain Ansatz

In PDE and numerical analysis, full Laplace and z transform analysis is usually performed through the use of an ansatz. For instance, for a continuous time LTI problem, it is wholly sufficient to simplify frequency domain analysis by examining a single-frequency solution

$$u(t) = e^{st}$$

Similarly, in discrete time, the ansatz

$$u_d^n = z^n$$

is also frequently employed. Various authors discuss the equivalence between this approach and full Laplace or z transform analysis [113, 244].

2.3.2 Frequency Domain Interpretation of Differential and Difference Operators

The frequency domain interpretation of differential operators is well-known to anyone who has taken an undergraduate course in electric circuits. In continuous time, at steady state, one has, immediately from (2.8),

$$\frac{d^m}{dt^m} u(t) \quad \xrightarrow{\mathcal{L}} \quad s^m \hat{u}(s)$$

In discrete time, for the unit shift, one has

$$e_{t\pm} u_d^n \quad \xrightarrow{\mathcal{Z}} \quad z^{\pm 1} \hat{u}_d(z)$$

and for the various first differences and averaging operators, one has:

$$\begin{aligned} \delta_{t+} u_d^n &\xrightarrow{\mathcal{Z}} \frac{1}{k} (z - 1) \hat{u}_d(z) & \delta_{t-} u_d^n &\xrightarrow{\mathcal{Z}} \frac{1}{k} (1 - z^{-1}) \hat{u}_d(z) & \delta_t u_d^n &\xrightarrow{\mathcal{Z}} \frac{1}{2k} (z - z^{-1}) \hat{u}_d(z) \\ \mu_{t+} u_d^n &\xrightarrow{\mathcal{Z}} \frac{1}{2} (z + 1) \hat{u}_d(z) & \mu_{t-} u_d^n &\xrightarrow{\mathcal{Z}} \frac{1}{2} (1 + z^{-1}) \hat{u}_d(z) & \mu_t u_d^n &\xrightarrow{\mathcal{Z}} \frac{1}{2} (z + z^{-1}) \hat{u}_d(z) \end{aligned}$$

The second difference operator δ_{tt} transforms according to

$$\delta_{tt} u_d^n \quad \xrightarrow{\mathcal{Z}} \quad \frac{1}{k^2} (z - 2 + z^{-1}) \hat{u}_d(z)$$

Under some conditions, it is useful to look at the behaviour of these discrete time operators when z is constrained to be of unit modulus (in other words, when $z = e^{j\omega k}$). For instance, the operators

δ_{tt} and μ_t . transform according to

$$\delta_{tt}u_d^n \implies -\frac{4}{k^2} \sin^2(\omega k/2) \hat{u}_d(e^{j\omega k}) \quad \mu_t.u_d^n \implies \cos(\omega k) \hat{u}_d(e^{j\omega k})$$

where the \implies is now interpreted as referring to a discrete-time Fourier transform.

The Taylor series analysis of the accuracy of difference and averaging operators may be viewed simply in the frequency domain. Considering, for example, the operator δ_{t+} , one may write, by expanding $z = e^{sk}$ in powers of s about $s = 0$,

$$\delta_{t+}u_d^n \xrightarrow{z} \frac{1}{k} (z - 1) \hat{u}_d(z) = \frac{1}{k} (e^{sk} - 1) \hat{u}_d(e^{sk}) = (s + O(k)) \hat{u}_d(e^{sk})$$

In other words, the difference operator δ_{t+} behaves, in the frequency domain, as a multiplication by a factor $s + O(k)$, corresponding to first derivative plus a correction of the order of the time step k , and is thus first order accurate.

2.3.3 Recursions and Polynomials in z

Finite difference schemes are recursions. Though concrete examples of such recursions, as derived from ordinary differential equations will appear at many instances in the following two chapters, a typical example is the following:

$$u_d^n = - \sum_{n'=1}^N a^{(n')} u_d^{n-n'} \quad (2.10)$$

The value of the time series u_d^n is calculated from a weighted combination of the last N values, and $a^{(n')}$ are the weighting coefficients. Such a recursion corresponds to an ODE without a forcing term, and for those with a signal processing background, is no more than an all-pole digital filter operating under zero-input (transient) conditions. If the coefficients $a^{(n')}$ are constants (i.e., the underlying problem is LTI), the usual analysis technique, here as in standard filter design, is to take a z transform of (2.10), to get

$$P(z)\hat{u}_d = 0 \quad \text{with} \quad P(z) = \sum_{n'=0}^N a^{(n')} z^{N-n'} \quad (2.11)$$

Frequency domain stability analysis of finite difference schemes is concerned with finding the roots of the polynomial $P(z)$, i.e.,

$$P(z) = 0 \quad (2.12)$$

where $a^{(0)} = 1$. These roots correspond to the natural frequencies of oscillation of the recursion (2.10). Polynomials such as the above arise when analyzing finite difference schemes for linear and shift-invariant PDEs as well, in which case the set of values $a^{(n')}$ may themselves be constant coefficient functions of spatial frequency variables. It is worth introducing some useful conditions on root locations here.

For numerical stability, it is usually the case that, just as for digital filters, one needs to bound any solution z to (2.11) by

$$|z| \leq 1 \quad \rightarrow \quad \sigma \leq 0$$

In other words, the roots must have magnitude less than or equal to unity, corresponding to damped

complex frequencies². One way of proceeding is to find the roots z to (2.11) explicitly, but if one is interested in merely finding out the conditions under which the roots are bounded, simpler tests exist, such as the Schur-Cohn recursive procedure [244], which is analogous to the Routh-Hurwitz stability test [275] for checking root locations of polynomials associated with continuous time systems. In many cases in physical modeling sound synthesis, however, the polynomial (2.11) is of second order, i.e.,

$$z^2 + a^{(1)}z + a^{(2)} = 0 \quad (2.13)$$

It is possible to show, either through an explicit calculation of the roots, or through the use of the Schur-Cohn recursion, that the roots of this quadratic will be bounded by unity if

$$|a^{(1)}| - 1 \leq a^{(2)} \leq 1 \quad (2.14)$$

See Problem 2.10. These conditions are plotted as a region in the $(a^{(1)}, a^{(2)})$ plane in Figure 2.2. This relatively simple pair of conditions will be used repeatedly throughout this book. It is also

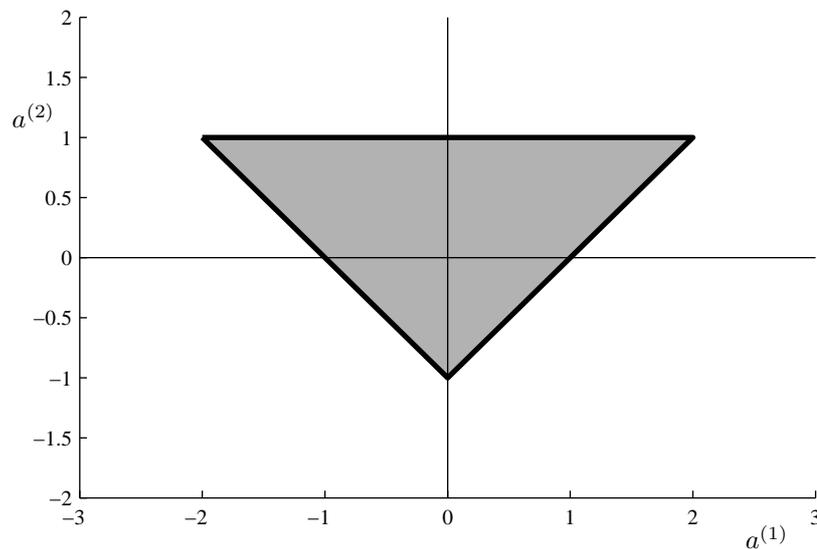


Figure 2.2: *Region of the $(a^{(1)}, a^{(2)})$ plane (in grey) for which the roots of Eq. (2.13) are bounded by unity in magnitude.*

worth noting that under certain conditions (namely losslessness of the underlying model problem), polynomials appear in which $a^{(2)} = 1$, implying the simpler condition

$$|a^{(1)}| \leq 2 \quad (2.15)$$

²This condition is, in fact, a little too simple to catch many special cases which may occur in practice. For example, through analysis of finite difference schemes, one can arrive at polynomials possessing double roots on the unit circle, leading, potentially, to linear growth of solutions; sometimes this is a natural consequence of the underlying model equations, sometimes not. Also, in some cases one may need to beware if the coefficients themselves are functions of parameters such as k , the time step, in order to ensure that roots are bounded in the limit as k becomes small.

2.3.4 Difference Operators and Digital Filters

As discussed in Chapter 1, the most prominent current physical modeling sound synthesis techniques, such as digital waveguides and scattering methods, are based, traditionally, around the use of digital filters, and the accompanying frequency domain machinery. It should be somewhat comforting then, that, at least in the LTI case, many of the primary results in the numerical simulation field (especially von Neumann analysis, to be introduced in Chapter 5) are described in a nearly identical language, employing Fourier, Laplace and z transforms, though spatial Fourier transforms and filters will be familiar only to those engineers with a background in image processing or optics. In this short section, some connections between the difference operators described in the preceding sections, and very simple digital filter designs are indicated.

Digital filters of high order play a central role in signal processing, including that of audio signals. Yet in physical modeling applications, due to the more strict adherence to an underlying model problem, higher order difference operators, especially in time, are much more difficult to use. One reason for this has to do with numerical stability—though moderately higher order schemes for ODEs, such as those of the Runge-Kutta variety are commonly seen in the literature, for distributed problems, the coupling with spatial differentiation can lead to severe difficulties in terms of analysis, even in the linear case. In general, complex behaviour in musical instrument simulations results from coupled low-order time-differences—in digital filtering terminology, one might of such a configuration in terms of “parallel banks of oscillators.” Though a conventional analysis technique, particularly for finite element methods [86, 66] involves reducing a spatially distributed problem to a system of ODEs (i.e., through so-called semi-discretization), in the nonlinear case, any stability results obtained in this way are generally not sufficient, and can be interpreted only as rules of thumb. Another reason is that in conventional audio filtering applications, the emphasis is generally on the steady state response of a given filter. But in physical modeling applications, in some cases, one is solely interested in the transient response of a system (percussive sound synthesis is a particular example). The higher the order of the time differentiation employed, the higher the number of initial conditions which must be supplied; as most systems in musical acoustics are of second order in time differentiation, time difference operators of degree higher than two will necessarily require the setting of extra “artificial” initial conditions.

It is useful to examine the difference operators defined in §2.2 in this light. Consider any LTI operator p (such as $\frac{d}{dt}$, the identity operator 1, or $\frac{d^2}{dt^2}$), applied to a function $x(t)$, and yielding a function $y(t)$. When viewed in an input output sense, one arrives, after Laplace transformation, at a transfer function form, i.e.,

$$y(t) = px(t) \quad \xrightarrow{\mathcal{L}} \quad \hat{y}(s) = h(s)\hat{x}(s)$$

where $h(s)$ is the transfer function corresponding to the operator p . When $s = j\omega$, one may find the magnitude and phase of the transfer function, as a function of ω , i.e.,

$$\text{magnitude} = |h(j\omega)| \quad \text{phase} = \angle h(j\omega)$$

Similarly, for a discrete time operator p_d , applied to a time series x_d^n , yielding a time series y_d^n , one has, after applying a z transformation,

$$y_d^n = p_d x_d^n \quad \xrightarrow{\mathcal{Z}} \quad \hat{y}_d(z) = h_d(z)\hat{x}_d(z)$$

Table 2.1: Comparison between continuous time operators $\frac{d}{dt}$, the identity 1, and $\frac{d^2}{dt^2}$, and various difference approximations, viewed in terms of transfer functions. For each operator, the continuous time transfer function is given as a function of s , and the discrete time transfer function as a function of z . Magnitude and phase are given for $s = j\omega$ or $z = e^{j\omega k}$.

C.T. op.	$h(s)$	$ h(j\omega) $	$\angle h(j\omega)$	D.T. op.	$h_d(z)$	$ h_d(e^{j\omega k}) $	$\angle h_d(e^{j\omega k})$
$\frac{d}{dt}$	s	$ \omega $	$-\frac{\pi}{2}, \quad \omega > 0$ $\frac{\pi}{2}, \quad \omega < 0$	δ_{t+}	$\frac{1}{k}(z-1)$	$\frac{2}{k} \sin(\frac{\omega k}{2}) $	$\frac{\pi}{2} + \frac{\omega k}{2}, \quad \omega > 0$ $-\frac{\pi}{2} + \frac{\omega k}{2}, \quad \omega < 0$
				δ_{t-}	$\frac{1}{k}(1-z^{-1})$	$\frac{2}{k} \sin(\frac{\omega k}{2}) $	$\frac{\pi}{2} - \frac{\omega k}{2}, \quad \omega > 0$ $-\frac{\pi}{2} - \frac{\omega k}{2}, \quad \omega < 0$
				δ_t	$\frac{1}{2k}(z-z^{-1})$	$\frac{1}{k} \sin(\omega k) $	$\frac{\pi}{2}, \quad \omega > 0$ $-\frac{\pi}{2}, \quad \omega < 0$
1	1	1	0	μ_{t+}	$\frac{1}{2}(z+1)$	$ \cos(\frac{\omega k}{2}) $	$\frac{\omega k}{2}$
				μ_{t-}	$\frac{1}{2}(1+z^{-1})$	$ \cos(\frac{\omega k}{2}) $	$-\frac{\omega k}{2}$
				μ_t	$\frac{1}{2}(z+z^{-1})$	$ \cos(\omega k) $	$0, \quad \omega < \frac{\pi}{2k}$ $\pi, \quad \omega > \frac{\pi}{2k}$
$\frac{d^2}{dt^2}$	s^2	ω^2	π	δ_{tt}	$\frac{1}{k^2}(z-2+z^{-1})$	$\frac{4}{k^2}\sin^2(\frac{\omega k}{2})$	π

where $h_d(z)$ is the transfer function corresponding to the operator p_d . Again, one can find the magnitude and phase by restricting z to $z = e^{j\omega k}$, i.e.,

$$\text{magnitude} = |h_d(e^{j\omega k})| \quad \text{phase} = \angle h_d(e^{j\omega k})$$

In Table 2.1, transfer functions for various differential and difference operators are given, as well as their magnitudes and phases. The discrete time transfer functions are readily interpreted in terms of well-known filter structures. Leaving issues of causality aside, the operators δ_{t+} and δ_{t-} behave as high-pass filters with a single zero at the DC frequency $z = 1$, and δ_t is a two zero filter with zeros at DC and the Nyquist frequency $z = -1$. The averaging operators μ_{t+} and μ_{t-} behave similarly as low pass filters, each with a single zero at the Nyquist frequency $z = -1$, and μ_t is a two zero filter with zeros at $z = \pm j$, or at one quarter the sample rate. δ_{tt} behaves as a two zero filter with a double zero at DC.

Notice that for all the difference operators presented in the table, the approximation to the magnitude response of the associated continuous time operator is second order accurate about $\omega = 0$ —that is, if one expands the magnitude of $|h(e^{j\omega k})|$, for $\omega \geq 0$, in powers of ω , one has

$$|h_d(e^{j\omega k})| = |h(j\omega)| + O(k^2)$$

But the same is not true of the phase response; for centered operators, such as δ_t , μ_t and δ_{tt} , the phase response is exact, at least in some neighbourhood around $\omega = 0$. For the other non-centered operators, it is not, and differs from the phase response of the associated continuous time operator by a term of order k . Thus the “first order accuracy” of such non-centered operators manifests itself in the phase behaviour, which is to be expected. Such behaviour may be directly related to the discussion of indeterminacy of accuracy of difference operators in isolation, from §2.2.3.

This behaviour is perhaps more easily viewed in frequency domain plots of the magnitude and phase of these operators, as given in Figure 2.3, as are certain other interesting features. For example, notice that, in terms of the magnitude response alone, the first-order accurate operators δ_{t+} and δ_{t-} better approximate the first derivative near $\omega = 0$ than the second-order accurate operator δ_t , which might seem somewhat surprising. This is due to the fact that the simple forward and backward differences employ values of the time series which are only one time step apart, rather

than two, in the case of the centered difference operator. Also note the “doubling” of the magnitude response in the case of δ_t , about one-quarter the sampling rate; this is also surprising, but then, the centered difference operator, which operates on values of the time series separated by two time steps, can be viewed as operating at a downsampled (by a factor of two) rate. This interesting doubling effect appears in finite difference schemes under certain conditions for various systems of interest in musical sound synthesis, such as the wave equation in 1D or 2D, which are discussed in Chapters 6 and 12, respectively.

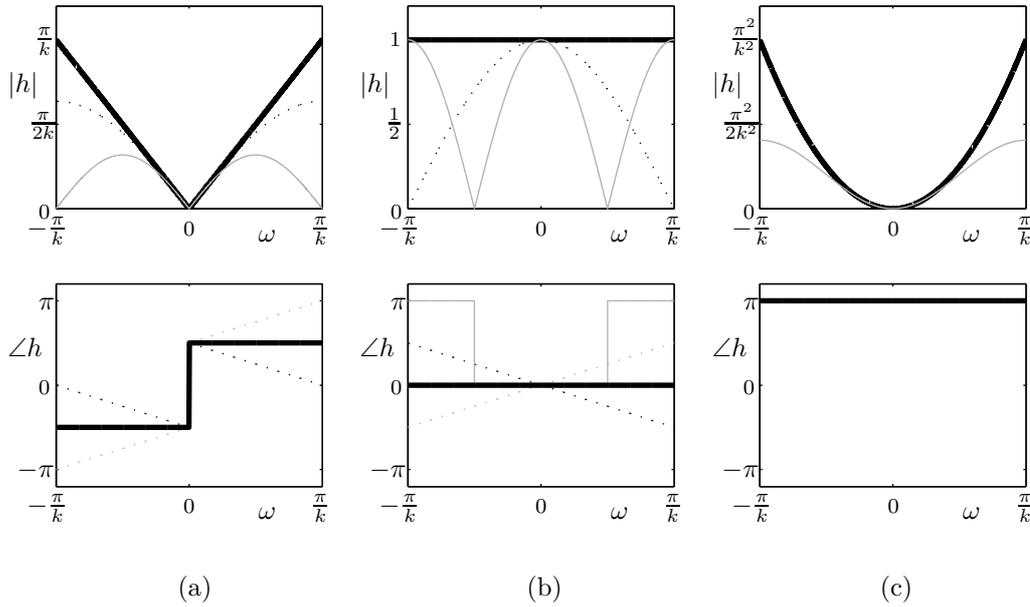


Figure 2.3: Magnitude (top) and phase (bottom) of difference approximations to (a) $\frac{d}{dt}$, (b) the identity operator 1 , and (c) $\frac{d^2}{dt^2}$. In all cases, the exact response of the continuous operator is plotted as a thick black line, over the interval $\omega \in [-\pi/k, \pi/k]$. The responses of the centered operators δ_t , μ_t , and δ_{tt} , when distinct from those of the continuous time operator, appear as solid grey lines. In the case of magnitude plots, the responses of the non-centered operators δ_{t+} , δ_{t-} , μ_{t+} and μ_{t-} appear as dotted black lines, and in the case of phase plots, as distinct dotted grey (forward operators) and black (backward operators) lines.

In the case of all the operators discussed here except for μ_{t-} and δ_{t-} , the resulting recursion appears to imply a non-causal relationship between an input sequence x_d^n and an output sequence y_d^n , but it is important to keep in mind that in numerical applications, this is not quite the correct interpretation. Generally, in a physical modeling sound synthesis application, there is not an input sequence as such (at least not at the audio rate), and no risk of the need for “looking into the future.” But the use of such “non-causal” operations can lead to implicit behaviour [244] in finite difference schemes in some cases—see §6.3.

The Trapezoid Rule

One difference approximation, of special importance in sound synthesis applications (and in particular, those built around the use of scattering based methods—see §1.2.4), is the trapezoid rule. In operator form, it looks like

$$(\mu_{t+})^{-1} \delta_{t+} \quad (2.16)$$

where the $(\mu_{t+})^{-1}$ is to be understood here as an operator inverse. As $(\mu_{t+})^{-1}$ remains an approximation to the identity operator, the operator as a whole still behaves as an approximation to a first time derivative. Like the operators δ_{t+} and δ_{t-} , it can be viewed as second-order accurate about a midpoint between adjacent values of the time series. Operationally, the best way to examine it is in the input output sense, for an input sequence x_d^n and output sequence y_d^n :

$$y_d^n = (\mu_{t+})^{-1} \delta_{t+} x_d^n \quad \text{or} \quad \frac{1}{2} (y_d^{n+1} + y_d^n) = \frac{1}{k} (x_d^{n+1} - x_d^n) \quad (2.17)$$

The trapezoid rule transforms as

$$y_d^n = (\mu_{t+})^{-1} \delta_{t+} x_d^n \quad \xrightarrow{z} \quad \hat{y}_d(z) = \frac{2z-1}{kz+1} \hat{x}_d(z) \quad (2.18)$$

In the frequency domain form, the trapezoid rule is often referred to as a bilinear transformation.

As mentioned above, the trapezoid rule figures prominently in scattering based approaches to synthesis, and is one of the cornerstones of wave digital filters [88]. An example of a wave digital structure employing the trapezoid rule, as well as a discussion of the relationship with other finite difference methods, appears in §3.3.3.

2.4 Energetic Manipulations and Identities

The frequency domain techniques presented in the previous section are of great utility in the analysis of linear and time-invariant (LTI) systems and difference schemes, and extend naturally to the distributed setting, in which case they are often referred to as von Neumann analysis [244]; such techniques can yield a great deal of important information regarding numerical stability, as well as the perceptual effects of discretization, in the form of so-called numerical phase velocity and dispersion. LTI systems, however, are only the starting point in musical sound synthesis based on physical models, and are not sufficiently complex to capture many of the more subtle and perceptually salient qualities of real musical sounds.

Frequency-domain techniques do not apply directly to nonlinear systems, nor to finite difference schemes which approximate them. It is often tempting to view nonlinear systems in musical acoustics as perturbations of linear systems and to apply frequency domain analysis in a loose sense. If the perturbations are small, this approach is justifiable, and can yield additional information regarding, say, the evolution of natural frequencies of oscillation. In many cases, however, these perturbations are not small; the vibration of percussion instruments such as gongs serve as an excellent example of the perceptual importance of highly nonlinear effects. Frequency domain analysis applied in such cases can be dangerous in the case of analysis, in that important transient effects are not well-modelled, and potentially disastrous in the case of sound synthesis based on finite difference schemes, in that numerical stability is impossible to rigorously ensure. Energetic techniques, which are based on direct time-domain analysis of finite difference schemes, yield less information than frequency

domain methods, but may be extended to deal with nonlinear systems and difference schemes quite easily, and deal with many issues, including numerical stability analysis and the proper choice of numerical boundary conditions, in a straightforward way. In fact, though these techniques are, as a rule, far less familiar than frequency domain methods, they are not much more difficult to employ. In this section, some basic manipulations are introduced.

2.4.1 Time Derivatives of Products of Functions or Time Series

Though, in this chapter, no systems have been defined as yet, and it is thus impossible to discuss quantities such as energy, or the Hamiltonian, some of the algebraic manipulations may be introduced. Energetic quantities are always written in terms of products of functions or, in the discrete case, time series. Consider, for example, the following products:

$$\frac{du}{dt} \frac{d^2u}{dt^2} \quad \frac{du}{dt} u \quad (2.19)$$

In energetic analysis, whenever possible, it is useful to rewrite terms such as these as time derivatives of a single quantity (in this case, some function of u or its time derivatives). For instance, the terms above may be simply rewritten as

$$\frac{d}{dt} \left(\frac{1}{2} \left(\frac{du}{dt} \right)^2 \right) \quad \frac{d}{dt} \left(\frac{1}{2} u^2 \right)$$

These are time derivatives of quadratic forms; in the context of the simple harmonic oscillator, which will be discussed in detail in Chapter 3, the quadratic forms above may be identified with the kinetic and potential energies of the oscillator, when u is taken as the dependent variable. Notice in particular that both quantities are squared quantities, and thus non-negative, regardless of the values taken by u or du/dt . It is useful to be able to isolate these energetic quantities, combinations of which are often conserved or dissipated, because from them, one may derive bounds on the size of the solution itself. Arriving at such bounds in the discrete case is, in fact, a numerical stability guarantee.

For linear systems, the energetic quantities are always quadratic forms. For nonlinear systems, they will generally not be, but manipulations similar to the above may still be performed. For instance, it is also true that

$$\frac{du}{dt} u^3 = \frac{d}{dt} \left(\frac{1}{4} u^4 \right) \quad (2.20)$$

This signals the utility of such analysis tools for nonlinear systems, where Fourier or Laplace based techniques become extremely unwieldy.

There are analogous manipulations in the case of products of time series under difference operators; the number is considerably greater, though, because of the multiplicity of ways of approximating differential operators, as was seen in §2.2. Consider, for instance, the products

$$(\delta_t \cdot u_d) \delta_{tt} u_d \quad u_d \delta_t \cdot u_d$$

where now, $u_d = u_d^n$ is a time series; these are clearly approximation to the expressions given in Eq.

(2.19). Expanding the first of these at time step n gives

$$\begin{aligned} (\delta_t u_d^n) \delta_{tt} u_d^n &= \frac{1}{2k} (u_d^{n+1} - u_d^{n-1}) \frac{1}{k^2} (u_d^{n+1} - 2u_d^n + u_d^{n-1}) \\ &= \frac{1}{2k} \left(\left(\frac{u_d^{n+1} - u_d^n}{k} \right)^2 - \left(\frac{u_d^n - u_d^{n-1}}{k} \right)^2 \right) \\ &= \delta_{t+} \left(\frac{1}{2} (\delta_{t-} u_d^n)^2 \right) \end{aligned}$$

Expanding the second gives

$$(\delta_t u_d^n) u_d^n = \frac{1}{2k} (u_d^{n+1} - u_d^{n-1}) u_d^n = \frac{1}{2k} (u_d^{n+1} u_d^n - u_d^n u_d^{n-1}) = \delta_{t+} \left(\frac{1}{2} u_d^n e_{t-} u_d^n \right)$$

These particular instances of products of time series under difference operators can thus be reduced to total differences of quadratic forms; but when one moves beyond quadratic forms to the general case, it is not true that every such approximation will behave in this way. As an illustration, consider two approximations to the quantity given on the left of Eq. (2.20) above:

$$(\delta_t u_d^n) (u_d^n)^3 \qquad (\delta_t u_d^n) (\mu_t u_d^n) (u_d^n)^2$$

The first expression above cannot be interpreted as the total difference of a quartic form, as per the right side of Eq. (2.20) in continuous time. But the second can—one may write:

$$(\delta_t u_d^n) (\mu_t u_d^n) (u_d^n)^2 = \frac{1}{2k} (u_d^{n+1} - u_d^{n-1}) \frac{1}{2} (u_d^{n+1} + u_d^{n-1}) (u_d^n)^2 \quad (2.21a)$$

$$= \frac{1}{4k} ((u_d^{n+1})^2 (u_d^n)^2 - (u_d^n)^2 (u_d^{n-1})^2) \quad (2.21b)$$

$$= \delta_{t+} \left(\frac{1}{4} (u_d^n)^2 (e_{t-} u_d^n)^2 \right) \quad (2.21c)$$

These distinctions between methods of approximation turn out to be crucial in the stability analysis of finite difference schemes through conservation or energy-based methods.

2.4.2 Product Identities

For the sake of reference, presented here are various identities which are of use in the energetic analysis of finite difference schemes. For a time series u_d^n , it is always true that

$$(\delta_t \cdot u_d) (\delta_{tt} u_d) = \delta_{t+} \left(\frac{1}{2} (\delta_{t-} u_d)^2 \right) \quad (2.22a)$$

$$(\delta_t \cdot u_d) u_d = \delta_{t+} \left(\frac{1}{2} u_d e_{t-} u_d \right) \quad (2.22b)$$

$$(\delta_{t+} u_d) \mu_{t+} u_d = \delta_{t+} \left(\frac{1}{2} u_d^2 \right) \quad (2.22c)$$

$$(\mu_t \cdot u_d) u_d = \mu_{t+} (u_d e_{t-} u_d) \quad (2.22d)$$

$$(\mu_t \cdot u_d) (\delta_t \cdot u_d) = \delta_t \cdot \left(\frac{1}{2} u_d^2 \right) \quad (2.22e)$$

$$u_d e_{t-} u_d = (\mu_{t-} u_d)^2 - \frac{k^2}{4} (\delta_{t-} u_d)^2 \quad (2.22f)$$

For two grid functions, u_d^n and w_d^n , the following identity (which corresponds to the product rule of differentiation) is also useful:

$$\delta_{t+} (uw) = (\delta_{t+} u) (\mu_{t+} w) + (\mu_{t+} u) (\delta_{t+} w) \quad (2.23)$$

Proofs of these identities are direct; see Problem 2.8. All these identities generalize in an obvious way to the distributed case—see the comment on page 110 for more on this.

An inequality of great utility, especially in bounding the response of systems subject to external excitations, is the following bound on the product of two numbers u and w :

$$|uw| \leq \frac{u^2}{2\alpha^2} + \frac{\alpha w^2}{2} \quad (2.24)$$

which holds instantaneously for any real number $\alpha \neq 0$, including values of any two time series u^n and w^n , or functions $u(t)$ and $w(t)$.

2.4.3 Quadratic Forms

In the energetic analysis of finite difference schemes for both linear and nonlinear systems, quadratic forms play a fundamental role—this is because the energy function for a linear system is always a quadratic function of the state, usually positive definite or semi-definite. Many nonlinear systems, some examples of which will be discussed in this book, may be written as linear systems incorporating extra nonlinear perturbation terms, and as a result, the energy for such a system can be written as a quadratic form plus an additional perturbation, which may be non-negative.

Consider a particular such form in two real variables, x and y :

$$\mathfrak{H}(x, y) = x^2 + y^2 + 2axy \quad (2.25)$$

where a is a real constant. It is simple enough to show that for $|a| < 1$, $\mathfrak{H}(x, y)$ is a paraboloid, and is positive definite (it is non-negative for all values of x and y , and possesses a unique minimum of

zero at $x = y = 0$). For $a = \pm 1$, the form $\mathfrak{H}(x, y)$ is still non-negative, but not positive definite, i.e., it takes the value zero over the family of points given by $x = \mp y$. Considering the case $|a| < 1$, consider level curves of \mathfrak{H} , at some value $\mathfrak{H} = \mathfrak{H}_0$ which are ellipses oriented at 45 degrees with respect to the x or y axis. It should be clear, by visual inspection of Figure 2.4(a), that for a given value of \mathfrak{H}_0 , the magnitudes of x and y are bounded, and in fact by

$$|x|, |y| \leq \sqrt{\frac{\mathfrak{H}_0}{1-a^2}} \quad (2.26)$$

See Problem 2.11. If $|a| > 1$, the level curves at $\mathfrak{H}(x, y) = \mathfrak{H}_0$ are hyperbolas, and it is simple to

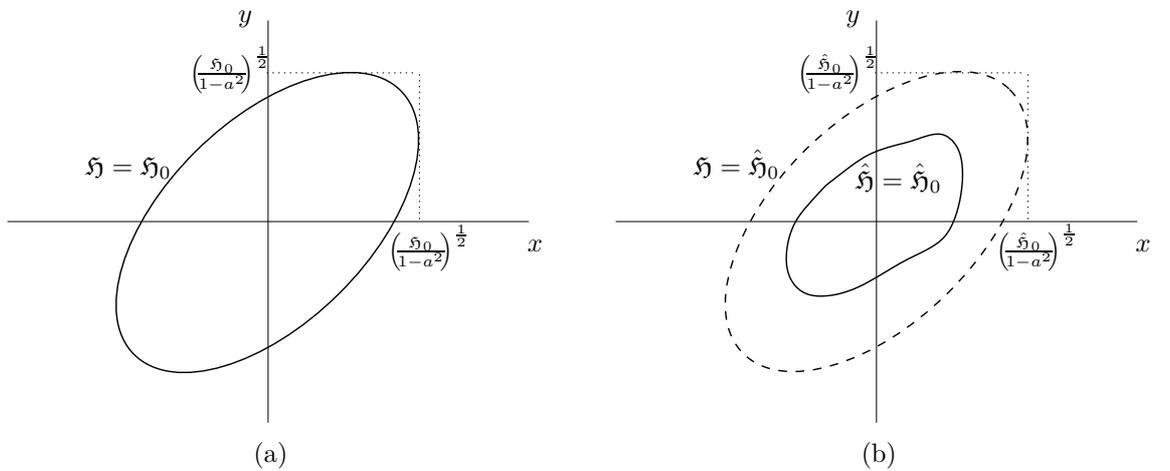


Figure 2.4: (a) A level curve of the quadratic form (2.25), for $\mathfrak{H} = \mathfrak{H}_0$. (b) A level curve of the nonlinear form (2.27) (solid line), and an associated level curve for the linear part (dotted line).

show that it is not possible to bound x or y in terms of \mathfrak{H}_0 . The same is true of the borderline case of $|a| = 1$.

Quadratic forms such as $\mathfrak{H}(x, y)$, as mentioned above, appear naturally as energy functions of finite difference schemes for linear systems. For nonlinear systems, one almost always³ has energy functions of the form

$$\hat{\mathfrak{H}}(x, y) = \mathfrak{H}(x, y) + \mathfrak{H}'(x, y) \quad (2.27)$$

where $\mathfrak{H}'(x, y)$ is another function of x and y , assumed non-negative, but not necessarily a quadratic form. Consider now a level curve of the function $\hat{\mathfrak{H}}(x, y)$, at $\hat{\mathfrak{H}} = \hat{\mathfrak{H}}_0$. Because $\mathfrak{H} \leq \hat{\mathfrak{H}}$, one may

³While it is true that LTI systems always possess energy functions which are quadratic forms, the converse is not necessarily true—that is, there do exist nonlinear systems for which the energy is still a quadratic form. An interesting (and rare) example is the so-called simplified von Karman model of plate vibration, discussed in depth in §13.2, which serves as an excellent model of strongly nonlinear behaviour in percussion instruments such as cymbals and gongs, for which the Hamiltonian is indeed a quadratic form. Though this may be counterintuitive, it is worth recalling certain nonlinear circuit components which are incapable of storing energy (such as an ideal transformer with a nonlinear winding ratio). A closed circuit network, otherwise linear except for such elements, will also possess a stored energy expressible as a quadratic function in the state variables.

deduce that

$$x^2 + y^2 + 2axy = \mathfrak{H}(x, y) \leq \hat{\mathfrak{H}}(x, y) = \hat{\mathfrak{H}}_0$$

and thus, using the same reasoning as in the case of the pure quadratic form, one may obtain the bounds

$$|x|, |y| \leq \sqrt{\frac{\hat{\mathfrak{H}}_0}{1 - a^2}}$$

This bound is illustrated in Figure 2.4(b). Thus even for extremely complex nonlinear systems, one may determine bounds on x and y provided that the additional perturbing energy term is non-negative, which is, in essence, no more than Lyapunov-type stability analysis [162]. This generality is what distinguishes energy-based methods from frequency domain techniques. Notice, however, that the bound is now not necessarily tight—it may be overly conservative, but through further analysis, it might well be possible to determine more strict bounds on x and y . In general, in the analysis of numerical schemes, these simple techniques allow one to deduce conditions for numerical stability for nonlinear systems, using no more than linear system techniques, but the price paid for such simplicity of analysis is that conditions derived may be only sufficient, and perhaps quite far from necessary.

In dynamical systems terminology, the level curve shown in Figure 2.4 may represent the path that a lossless system's state traces in the so-called phase plane; such a curve represents the constraint of constant energy (i.e., \mathfrak{H}_0) in such a system. Though phase plane analysis is not used in this book, most of the lossless systems (and associated numerical methods) can, and should be imagined in terms of such curves or surfaces of constant energy. A certain familiarity with phase plane analysis in the analysis of nonlinear systems is essential to an understanding of the various phenomena which arise [178], but in this book, only those tools that will be useful for practical robust algorithm design for sound synthesis will be developed. Symplectic numerical methods, very much related to energy techniques, are based directly on the analysis of the time evolution of numerical solutions in phase space [41, 222]

2.5 Problems

Problem 2.1 Consider the difference operator δ_{t-} . When applied to a continuous function $u(t)$, it will require access to the values $u(t)$ and $u(t - k)$. Show that it is a second-order accurate approximation to the derivative du/dt by expanding in Taylor series about the time $t - \frac{k}{2}$.

Problem 2.2 Show that the operators

$$(a) \quad \delta_{t+}\delta_{tt} \qquad (b) \quad \delta_{t-}\delta_{tt} \qquad (c) \quad \delta_t.\delta_{tt}$$

are all approximations to a third time derivative. For each, explicitly write the expression which results when applied to a time series u_d^n . What is the accuracy of such operators acting in isolation, when the expansion point is taken to be the instant $t = nk$? About which time instants should one expand in Taylor series in order to obtain a maximal order of accuracy in each case? What is the temporal width (see §2.2.1) for each operator?

Problem 2.3 Describe, in operator notation, the complete family of second-order accurate approximations to a second time derivative which operates over a width of five levels. Using Taylor series

expansions, find the unique member of this family which is fourth order accurate, when the expansion point is taken to be the central point of the five-point set of values.

Problem 2.4 Show that the operator defined by

$$\delta_t - \frac{k^2}{6}\delta_{t+}\delta_{tt}$$

approximates a first time derivative to third order accuracy, and find the appropriate time instant about which to perform a Taylor expansion. Show that this operator is of width four, and it is the only such operator of third order accuracy.

Problem 2.5 For all the difference operators described in Problems 2.2, 2.3 and 2.4 above, find the transfer function description $h_d(z)$. Where do the zeros lie in each case? What is the multiplicity of each zero? Plot the magnitude and phase response in each case.

Problem 2.6 Prove that the composition of any number of operators of the form μ_{t+} , μ_{t-} or μ_t is an approximation to the identity operator.

Problem 2.7 Prove that the composition of any number of operators of the form μ_{t+} , μ_{t-} or μ_t with a single operator of type δ_{t+} , δ_{t-} or δ_t is an approximation to a first time derivative.

Problem 2.8 Prove the identities given in Eqs. (2.22) and (2.23).

Problem 2.9 Consider the quadratic polynomial equation (2.13) in the variable z when $a_2 = 1$, namely

$$z^2 + a^{(1)}z + 1 = 0$$

where $a^{(1)}$ is a real constant. Show that, depending on the value of $a^{(1)}$, the roots z_+ and z_- of the above equation will be either a) complex conjugates, or b) real, and find the condition on $a^{(1)}$ which distinguishes these two cases. Also show that the product z_+z_- is equal to one in either case, without using the explicit forms of z_+ and z_- , and deduce that in case a) above, z_+ and z_- are of unit modulus, and in case b), if the roots are distinct, one must be of magnitude greater than unity.

Problem 2.10 Prove condition (2.14) for the quadratic 2.13. You should use the method described in the previous problem as a starting point.

Problem 2.11 Consider the quadratic form (2.25), and show that if $|a| \geq 1$, it is possible to find arbitrarily large values of x and y which solve the equation $\mathfrak{H} = \mathfrak{H}_0$, for any value of \mathfrak{H}_0 .

Problem 2.12 Consider the function \mathfrak{H} defined as

$$\mathfrak{H} = \frac{1}{2}(\delta_t u_d)^2 + \frac{\omega_0^2}{2}u_d e_{t-u_d}$$

where ω_0 is a real constant, and $u_d = u_d^n$ is a time series, with time step k . Evaluating \mathfrak{H} at time step n , show that it is a quadratic form as given in Eq. 2.25, in the variables u_d^n and u_d^{n-1} , except for a constant scaling. Find a condition on k , the time step, such that \mathfrak{H} is positive definite. Under this condition, determine a bound on u_d^n in terms of \mathfrak{H} , ω_0 and k . (\mathfrak{H} as defined here is an energy function for a particular finite difference scheme for the simple harmonic oscillator, which is discussed in §3.2. The analysis performed in this problem leads, essentially, to a numerical stability condition for the scheme.)

Problem 2.13 Consider the function \mathfrak{H} defined as

$$\mathfrak{H} = \frac{1}{2}(\delta_{t-}u_d)^2 + \frac{\omega_0^2}{2}u_d e_{t-}u_d + \frac{b^2}{4}u_d^2 e_{t-}(u_d^2)$$

where ω_0 and b are real constants, and $u_d = u_d^n$ is a time series, with time step k . Evaluating \mathfrak{H} at time step n , show that it is a positive definite form as given in Eq. (2.27), in the variables u_d^n and u_d^{n-1} , except for a constant scaling. Perform an analysis similar to that of the previous problem, determining a bound on the size of u_d^n in terms of \mathfrak{H} , ω_0 and k . (\mathfrak{H} as defined here is an energy function for a particular finite difference scheme for a nonlinear oscillator, which is discussed in §4.2.1. The analysis performed in this problem leads, again, to a numerical stability condition for the scheme, now in the nonlinear case.)

Chapter 3

The Oscillator

An oscillator of some kind is the sound producing mechanism in nearly every music-making device, including most acoustic Western orchestral instruments, many from outside the Western tradition, electromechanical instruments, analogue synthesizers, and digital synthesis algorithms. In the context of the latter case, which is the subject in this book, the oscillator was one of the first digital sound synthesis modules to appear (see §1.1.1), and continues to play a key role in physical modelling algorithms—indeed, every system described in this book is, at its most basic level, an oscillator.

It is interesting, however, that the reasons underlying the predominance of the oscillator in digital sound synthesis are distinct in the case of early, or abstract sound synthesis algorithms, discussed in §1.1 and physical modelling algorithms, described in §1.2. For abstract methods, sinusoids were seized upon at an early stage, apparently because of their particular perceptual significance in human audition: a sinusoid is perceived as a unit, characterized by a bare minimum of parameters, namely amplitude and frequency, which correlate well with perceived loudness and pitch. This is particularly true of additive and FM synthesis, but also of subtractive synthesis methods, in which case a non-sinusoidal oscillator is often used as the excitation, as well as wavetable methods, for which values are often read periodically from the table. In physical modelling synthesis, which is based on descriptions of objects which are subject to the laws of physics, sinusoids (more properly complex exponentials, or damped sinusoids) appear naturally as eigenfunctions of linear and time invariant systems, and many musical instruments can be well-described to a first approximation as such. That sinusoids play such a dual perceptual/physical role is certainly not a coincidence—if the world is, to a first approximation, a linear system, then humans will be well-adapted to perceive collections of sinusoids. Thus the distinction made above is not as clear as it might first appear to be. Regardless of one's point of view, the linear oscillator is clearly the starting point for any description of sound synthesis, through physical modeling or any other method.

It is interesting to note that in many parts of the numerical simulation literature, the second-order simple harmonic oscillator is *not* the point of departure for descriptions of numerical methods—rather, a first-order linear ODE is usually taken to be the basic equation. The reason for this is that much research into numerical simulation is geared towards dealing with highly nonlinear fluid dynamics problems, which are often most easily expressed as systems of first order equations. It would be simple enough to begin with such a first order system here as well, and then progress to the second order oscillator by writing it as a first order system in two variables (see §3.1.3), but in fact, most systems of interest in musical acoustics (and in particular those involving the vibration of solids) can be expressed directly as second-order in time ODEs or PDEs; the finite element literature, also concerned first and foremost with the vibration of solids, shares this point of view.

This chapter is concerned, generally, with outlining the properties of the second order linear oscillator, introduced in §3.1, as well as some simple, and practically useful finite difference schemes, in §3.2 and §3.3. There is an emphasis here on obtaining bounds on solution size; though this is of course a simple matter in the case of a linear oscillator, the tools and techniques introduced here will play a role in determining numerical stability conditions for simulations of musical systems which are far more complex. Both frequency domain analysis, which is familiar to most students of signal processing, and energetic methods, which will be much less so, are presented here. The former set of methods is useful for linear problems, but the latter is more general, and extends well to interesting nonlinear musical systems, many examples of which will appear subsequently in this book. Such methods are applied to lumped systems of masses and springs in §3.4. Finally, the simple harmonic oscillator with loss, and associated finite difference schemes, are discussed in §3.5.

3.1 The Simple Harmonic Oscillator

The simple harmonic oscillator, perhaps the single most important ordinary differential equation (ODE) in physics, and of central importance to musical sound synthesis, is defined as

$$\frac{d^2u}{dt^2} = -\omega_0^2u \quad (3.1)$$

It is a second order ODE, and depends on the single parameter ω_0 , also known as the angular frequency of oscillation. The frequency f_0 , in Hertz, is given by $f_0 = \omega_0/2\pi$. The harmonic oscillator, as it is second order, requires the specification of two initial conditions, normally

$$u(0) = u_0 \quad \left. \frac{du}{dt} \right|_{t=0} = v_0$$

Equation (3.1) above may be arrived at in a variety of different contexts. In mechanics and acoustics, the canonical example is that of the mass/spring system, illustrated in Figure 3.1(a). A mass M is connected, via a linear spring of spring constant K to a rigid support, and $u(t)$ represents the variation of the displacement of the mass about an equilibrium spring extension. Under these conditions, and if initial conditions such as u_0 and v_0 above are given, $u(t)$ satisfies (3.1) with $\omega_0 = \sqrt{K/M}$. The oscillator also appears naturally in electrical circuit theory, as illustrated in Figure 3.1(b)—here, $u(t)$ represents the voltage across a linear capacitor, of capacitance C in series with an linear inductor of inductance L . Again, if appropriate initial conditions are supplied, the voltage will evolve according to (3.1), with $\omega_0 = 1/\sqrt{LC}$. Though electrical circuit theory would appear to be a poor match for problems involving musical instrument modelling, and in fact will appear only fleetingly in the present treatment, it is worth keeping in mind that scattering methods such as digital waveguides [242] and wave digital filters [88], which are heavily used in physical modelling sound synthesis, were all first developed using concepts borrowed from electrical network theory. See §1.2.4 for some general remarks on the use of scattering methods in synthesis, and §3.3.3 for a brief technical discussion of wave digital filters. There is also, of course, a long tradition in acoustics of modeling lumped systems in terms of equivalent circuit elements, primarily for simplicity of representation [174].

3.1.1 Sinusoidal Solution

It is well-known that the solution to (3.1), if $u(t)$ is constrained to be real-valued, and if $\omega_0 \neq 0$, has the form

$$u(t) = A \cos(\omega_0 t) + B \sin(\omega_0 t) \quad (3.2)$$

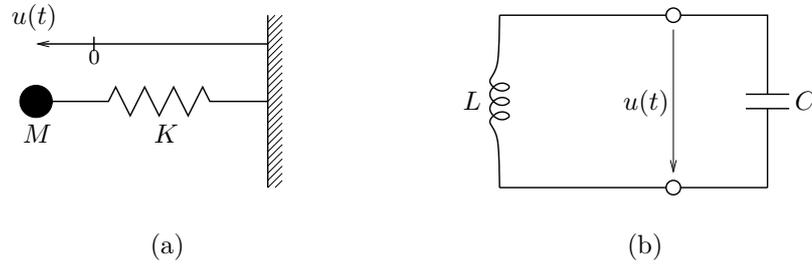


Figure 3.1: (a) Mass-spring system, with a mass M and a spring of stiffness K , for which the displacement $u(t)$, measured about an equilibrium distance (marked as 0) solves (3.1). (b) A series connection of an inductor, of inductance L , and a capacitor of capacitance C , for which the voltage $u(t)$ across the capacitor solves (3.1).

where clearly, one must have

$$A = u_0 \quad B = v_0/\omega_0$$

Another way of writing (3.2) is as

$$u(t) = C_0 \cos(\omega_0 t + \phi_0) \quad (3.3)$$

where

$$C_0 = \sqrt{A^2 + B^2} \quad \phi_0 = \tan^{-1}(-B/A) \quad (3.4)$$

C_0 is the amplitude of the sinusoid, and ϕ_0 is the initial phase. See Figure 3.2.

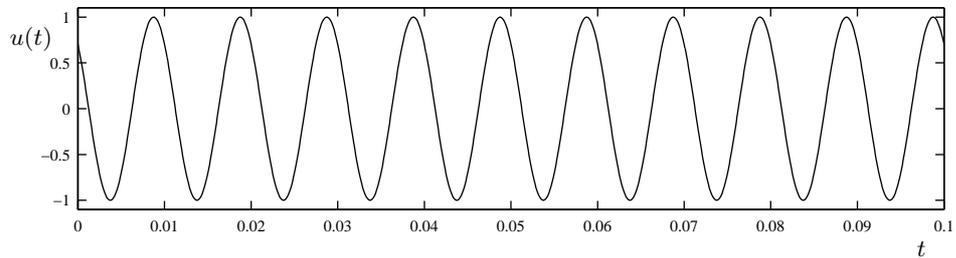


Figure 3.2: Sinusoid, of the form given in (3.3), with $C_0 = 1$, $f_0 = \omega_0/2\pi = 100$, and $\phi_0 = \pi/4$.

From the sinusoidal form (3.3) above, it is easy enough to deduce that

$$|u(t)| = |C_0 \cos(\omega_0 t + \phi_0)| \leq C_0 \quad \text{for } t \geq 0 \quad (3.5)$$

which serves as a convenient bound on the size of the solution for all t , purely in terms of the initial conditions and the system parameter ω_0 . Such bounds, in a numerical sound synthesis setting, are extremely useful, especially if a signal such as u is to be represented in a limited precision audio format. Note however, that the bound above is obtained only through a priori knowledge of the form of the solution itself (i.e., it is a sinusoid). As will be shown in the next section, it is not really necessary to make such an assumption.

The use of sinusoidal, or more generally complex exponential solutions to a given system in order to derive bounds on the size of the solution was developed, in the discrete setting, into a framework for determining numerical stability of a simulation algorithm, also known as von Neumann analysis

[244, 113]. Much more will be said about this from Chapter 5 onwards.

3.1.2 Energy

It is straightforward to derive an expression for the energy of the simple harmonic oscillator. Recalling the discussion at the beginning of §2.4.1, after multiplying (3.1) by $\frac{du}{dt}$, one arrives at

$$\frac{du}{dt} \frac{d^2u}{dt^2} + \omega_0^2 \frac{du}{dt} u = 0 \quad \rightarrow \quad \frac{d}{dt} \left(\frac{1}{2} \left(\frac{du}{dt} \right)^2 + \frac{\omega_0^2}{2} u^2 \right) = 0 \quad (3.6)$$

Writing

$$\mathfrak{H} = \mathfrak{T} + \mathfrak{V} \quad \text{with} \quad \mathfrak{T} = \frac{1}{2} \left(\frac{du}{dt} \right)^2 \quad \text{and} \quad \mathfrak{V} = \frac{\omega_0^2}{2} u^2 \quad (3.7)$$

one has

$$\frac{d\mathfrak{H}}{dt} = 0 \quad \rightarrow \quad \mathfrak{H}(t) = \text{constant} \geq 0$$

When scaled by a constant with dimensions of mass, \mathfrak{T} , \mathfrak{V} and \mathfrak{H} are the kinetic energy, potential energy, and total energy (or Hamiltonian) for the simple harmonic oscillator.

In order to arrive at a bound on solution size in this case, one may first observe that

$$\mathfrak{H}(t) = \mathfrak{H}(0) = \frac{1}{2} v_0^2 + \frac{1}{2} \omega_0^2 u_0^2 = \frac{1}{2} \omega_0^2 C_0^2$$

where C_0 is defined in (3.4). Then, noting the non-negativity of \mathfrak{T} and \mathfrak{V} , as defined in (3.7), one has

$$0 \leq \mathfrak{V}(t) \leq \mathfrak{H}(t) = \mathfrak{H}(0) = \frac{1}{2} \omega_0^2 C_0^2$$

and, using the form of $\mathfrak{V}(t)$, finally,

$$|u(t)| \leq C_0 \quad \text{for} \quad t \geq 0 \quad (3.8)$$

which is the same as the bound obtained in the previous section. In this case, however, one has made no a priori assumptions about the form of the solution. A bound on du/dt can clearly also be obtained; though not so useful for the SHO given the strength of the above condition on u itself, such bounds do come in handy when dealing with distributed systems under free boundary conditions. See Problem 3.1.

This type of analysis, in which frequency domain analysis is avoided, has also been applied to the analysis of numerical simulation techniques, and is sometimes referred to as the energy method [201]. It is worth noting that even though it was stated at the beginning of this section that an energy would be “derived” from system (3.1), one could equally well begin from the expressions for kinetic and potential energy given in (3.7), and, through the application of variational principles, arrive at (3.1). This variational point of view dominates in the modern finite element simulation community.

3.1.3 As First-Order System

In the literature, many numerical time-integration techniques are introduced with regard to first order ODE systems. It is simple enough to expand the definition of the simple harmonic oscillator, from (3.1), to a first order system in two variables, i.e., in matrix form,

$$\frac{d}{dt} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

All frequency domain and energy analysis of course applied equally to this equivalent system. Though first-order systems will only rarely appear in this book, it is worth being aware of such systems, as many time-integration techniques, including the ubiquitous Runge-Kutta family of methods and wave digital filters [88], are indeed usually presented with reference to first order systems. In the distributed case, the important “finite-difference time domain” family of methods [249], is also usually applied to first order systems, such as the defining equations of electromagnetics.

3.1.4 Coupled Systems of Oscillators

Coupled second order ODE systems appear frequently in finite element analysis, and, in musical sound synthesis, directly as descriptions of the dynamics of lumped networks, as per §1.2.1; what often remains, after spatial discretization of an LTI distributed system, is a system of the form:

$$\mathbf{M} \frac{d^2}{dt^2} \mathbf{u} = -\mathbf{K} \mathbf{u} \quad (3.9)$$

where here, \mathbf{u} is an $N \times 1$ column vector, and \mathbf{M} and \mathbf{K} are known as, respectively, the $N \times N$ mass and stiffness matrices, which are constants. \mathbf{M} is normally diagonal, and \mathbf{K} is usually symmetric, reflecting reciprocity of forces acting within the system defined by (3.9). If the product $\mathbf{M}^{-1}\mathbf{K}$ exists and is diagonalizable, with $\mathbf{M}^{-1}\mathbf{K} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$, then the system may be immediately decoupled as

$$\frac{d^2}{dt^2} \mathbf{v} = -\mathbf{\Lambda} \mathbf{v} \quad (3.10)$$

where $\mathbf{u} = \mathbf{U}\mathbf{v}$, and $\mathbf{\Lambda}$ is the diagonal matrix containing the eigenvalues of $\mathbf{M}^{-1}\mathbf{K}$, which are normally real and non-negative. The system (3.9) will then have N distinct frequencies ω , given by

$$\omega_p = \sqrt{\mathbf{\Lambda}_{p,p}} \quad \text{for } p = 1, \dots, N \quad (3.11)$$

Energy analysis also extends to systems such as (3.9). Left-multiplying by the row vector $d\mathbf{u}^T/dt$, where T indicates a transposition operation gives

$$\frac{d\mathbf{u}^T}{dt} \mathbf{M} \frac{d^2 \mathbf{u}}{dt^2} + \frac{d\mathbf{u}^T}{dt} \mathbf{K} \mathbf{u} = 0$$

or, using the symmetry of \mathbf{M} and \mathbf{K} ,

$$\frac{d\mathfrak{H}}{dt} = 0 \quad \text{with} \quad \mathfrak{H} = \mathfrak{T} + \mathfrak{V} \quad \text{and} \quad \mathfrak{T} = \frac{1}{2} \frac{d\mathbf{u}^T}{dt} \mathbf{M} \frac{d\mathbf{u}}{dt} \quad \mathfrak{V} = \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u}$$

If \mathbf{M} and \mathbf{K} are positive definite, then the energy \mathfrak{H} will also be non-negative, and bounds on solution size may be derived as in the case of a single oscillator.

Lumped network approaches to sound synthesis, mentioned in §1.2.1, are built, essentially, around such coupled oscillator systems. The numerical treatment of such systems will appear later in §3.4. See Problems 3.2 and 3.3 for some simple concrete examples of mass spring networks.

3.2 A Simple Scheme

The simplest possible finite difference scheme for the simple harmonic oscillator (3.1) is obtained by introducing a time step k , a time series u^n intended to approximate the solution at time $t = nk$, and by replacing the second time derivative by a second time difference, as defined in (2.4). One has, in operator form,

$$\delta_{tt} u = -\omega_0^2 u \quad (3.12)$$

Note the compactness of the above representation, and in particular the absence of the time index, which is assumed to be n for each occurrence of the variable u . The finite difference scheme (3.12)

is a second order accurate approximation to (3.1)—see §3.2.5. For reference, a code example in the Matlab programming language is provided in §A.1.

3.2.1 As Recursion

By expanding out the behaviour of the operator δ_{tt} , and reintroducing the time index n , one loses compactness of representation, but has a clearer picture of how to program such a scheme on a computer. One has

$$\frac{1}{k^2} (u^{n+1} - 2u^n + u^{n-1}) = -\omega_0^2 u^n \quad (3.13)$$

which relates values of the time series at three levels $n + 1$, n , and $n - 1$. Rewriting (3.13) so that u^{n+1} is isolated gives

$$u^{n+1} = (2 - \omega_0^2 k^2) u^n - u^{n-1} \quad (3.14)$$

which is a two-step recursion in the time series u^n . This difference equation is identical in form to that of a two-pole digital filter under zero-input conditions [183].

3.2.2 Initialization

The difference equation (3.14) must be initialized with two values, typically u^0 and u^1 . This pair of values is slightly different from what one would normally use to initialize the continuous time harmonic oscillator, namely the values $u(0)$ and $\frac{du}{dt}|_{t=0}$. In the distributed case, especially under striking conditions, it is often the initial velocity which is of interest. Supposing that one has, instead of the two values u^0 and u^1 , an initial value u_0 and an initial velocity condition (call it v_0), a very simple way of proceeding is to write

$$u^0 = u_0 \quad \text{and} \quad \delta_{t+} u^0 = v_0 \quad \rightarrow \quad u^1 = u_0 + kv_0$$

This approximation is only first-order accurate. More accurate settings may be derived—see Problem 3.4—but, in general, there is not much point in setting initial conditions to an accuracy greater than that of the scheme itself. Since many of the schemes used in this book will be of relatively low accuracy, the above condition is wholly sufficient for most sound synthesis applications, especially if one is operating at an elevated audio sampling rate (i.e., for a small value of k), which is usually the case. In many cases, in fact, an initial condition is not employed, as the physical model is excited by a continuously varying source (such as a bow, or reed mechanism).

3.2.3 Numerical Instability

It is easy enough to arrive at a numerical scheme such as (3.12) approximating the SHO, and, indeed, under some conditions it does generate reasonable results, such as the sinusoidal output shown in Figure 3.3(a). But this is not necessarily the case—sometimes, the solution exhibits explosive growth, as shown in Figure 3.3(b). Such explosive growth clearly does not correspond, in any sense, to an approximation to the SHO, and is referred to as unstable.

Stability analysis is a key aspect of all numerical methods, and is given an involved treatment throughout the rest of this book. It is particular importance in the case of physical models of musical instruments, which almost always operate under nearly lossless conditions, and which are thus especially prone to such instability. There are various ways of analyzing this behaviour—in the present case of the SHO, frequency domain methods, described in the following section are a natural choice. Energy methods, which extend well to more complex (nonlinear) problems appear in §3.2.6.

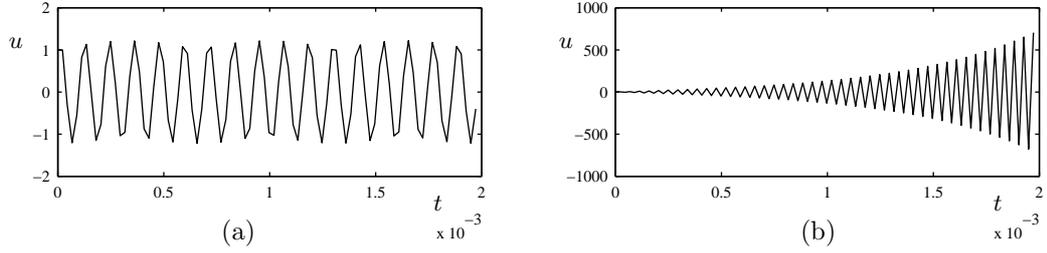


Figure 3.3: Output of scheme (3.12), running at a sample rate $f_s = 44100$ Hz. In (a), sinusoidal output when $f_0 = \omega_0/2\pi = 8000$ Hz, and in (b) unstable, explosive numerical behavior when $f_0 = \omega_0/2\pi = 14040$ Hz.

3.2.4 Frequency Domain Analysis

The frequency domain analysis of scheme (3.12) is very similar to that of the simple harmonic oscillator, as outlined in §3.1.1. Applying a z -transformation, or, simply inserting a test solution of the form $u^n = z^n$, where $z = e^{sk}$ (see comments on the frequency domain ansatz in §2.3), leads to the characteristic equation

$$z + (k^2\omega_0^2 - 2) + z^{-1} = 0$$

which has the solution

$$z_{\pm} = \frac{2 - k^2\omega_0^2 \pm \sqrt{(2 - k^2\omega_0^2)^2 - 4}}{2} \quad (3.15)$$

When z_{\pm} are distinct, the solution to (3.12) will evolve according to

$$u^n = A_+ z_+^n + A_- z_-^n \quad (3.16)$$

Under the condition

$$k < \frac{2}{\omega_0} \quad (3.17)$$

the two roots z_{\pm} will be complex conjugates of magnitude unity (see §2.3.3), in which case they may be written as

$$z_{\pm} = e^{\pm j\omega k}$$

where $\omega \neq \omega_0$, and the solution (3.16) may be rewritten as

$$u^n = A \cos(\omega nk) + B \sin(\omega nk) = C_0 \cos(\omega nk + \phi_0) \quad (3.18)$$

where

$$A = u^0 \quad B = \frac{u^1 - u^0 \cos(\omega k)}{\sin(\omega k)} \quad C_0 = \sqrt{A^2 + B^2} \quad \phi_0 = \tan^{-1}(-B/A)$$

This solution is clearly well-behaved, and resembles the solution to the continuous time SHO, from (3.2). From the form of the solution above, the following bound on the numerical solution size, in terms of the initial conditions and ω_0 may be deduced:

$$|u^n| \leq C_0 \quad (3.19)$$

Condition (3.17) may be violated in two different ways. If $k > \frac{2}{\omega_0}$, then the two roots of (3.15) will be real, and one will be of magnitude greater than unity. Thus solution (3.16) will grow exponentially. If $k = \frac{2}{\omega_0}$, then the two roots of (3.15) coincide, at $z_{\pm} = -1$. In this case, the solution (3.16) does not hold, and there will be a term which grows linearly. In neither of the above cases

does the solution behave in accordance with (3.2), and its size cannot be bounded in terms of initial conditions alone. Such growth is called numerically unstable (and in the latter case marginally unstable), and condition (3.17) serves as a stability condition.

Stability Conditions and Sampling Theory

Condition (3.17) may be rewritten, using sample rate $f_s = 1/k$ and the reference oscillator frequency $f_0 = \omega_0/2\pi$ as

$$f_s > \pi f_0$$

which, from the point of view of sampling theory, is counterintuitive—recall the general comments on this subject at the beginning of §2.1. One might expect that the sampling rate necessary to simulate a sinusoid at frequency f_0 should satisfy $f_s > 2f_0$, not the above condition, which is more restrictive. The reason for this is that the numerical solution does not in fact oscillate at frequency ω_0 , but at frequency ω given by

$$\omega = \frac{1}{k} \cos^{-1} (1 - k^2 \omega_0^2 / 2)$$

This frequency warping effect is due to approximation error in the finite difference scheme (3.12) itself; such an effect will be generalized to numerical dispersion in distributed problems seen later in this book, and constitutes what is perhaps the largest single disadvantage of using time-domain methods for sound synthesis. Note in particular that $\omega > \omega_0$. Perceptually, such an effect will lead to mistuning of “modes” in a simulation of a musical instrument. There are many means of combating this unwanted effect, perhaps the most crude being to use a higher sampling rate (or smaller value of k)—the numerical frequency ω approaches ω_0 in the limit as k becomes small. This approach, however, though somewhat standard in the more mainstream simulation community, is not so useful in audio, as the sample rate is generally fixed¹, though downsampling is a possibility. Another approach, which will be outlined in §3.3.2, involves different types of approximations to (3.1), and will be elaborated upon extensively throughout the rest of this book.

3.2.5 Accuracy

Scheme (3.12) employs only one difference operator, a second-order accurate approximation δ_{tt} to the operator $\frac{d^2}{dt^2}$. While it might be tempting to conclude that the scheme itself will generate a solution which converges to the solution of (3.1) with an error that depends on k^2 (which is in fact true in this case), the analysis of accuracy of an entire scheme is slightly more subtle than that applied to a single operator in isolation. It is again useful to consider the action of the operator δ_{tt} on a continuous function $u(t)$. In this case, scheme (3.12) may be rewritten as

$$(\delta_{tt} + \omega_0^2) u(t) = 0 \tag{3.20}$$

or, using (2.6),

$$\left(\frac{d^2}{dt^2} + \omega_0^2 \right) u(t) = O(k^2)$$

The left hand side of the above equation would equal 0 for $u(t)$ solving (3.1), but for approximation (3.12), there is a residual error on the order of the square of the time step k .

In general, the accuracy of a given scheme will be at least that of the constituent operators, and,

¹It is also worth mentioning that in a distributed setting, involving approximations in both space and time, the spatial temporal warping errors often tend to cancel one another to a certain degree (indeed, perfectly, in the case of the 1D wave equation).

in some rare cases, higher; two interesting case studies of great relevance to sound synthesis are presented in §3.3.4 and §6.2.4.

3.2.6 Energy Analysis

Beginning from the compact representation of the difference scheme given in (3.12), one may derive a discrete conserved energy similar to that obtained for the continuous problem in §3.1.2. Multiplying (3.12) by a discrete approximation to the velocity, $\delta_t u$ gives

$$(\delta_t u) (\delta_{tt} u) + \omega_0^2 (\delta_t u) u = 0$$

Employing the product identities given in §2.4.2 in the last chapter, one may immediately arrive at

$$\delta_{t+} \left(\frac{1}{2} (\delta_{t-} u)^2 + \frac{\omega_0^2}{2} u e_{t-} u \right) = 0$$

or

$$\delta_{t+} \mathfrak{h} = 0 \tag{3.21}$$

with

$$\mathfrak{h} = \mathfrak{t} + \mathfrak{v} \quad \text{and} \quad \mathfrak{t} = \frac{1}{2} (\delta_{t-} u)^2 \quad \mathfrak{v} = \frac{\omega_0^2}{2} u e_{t-} u \tag{3.22}$$

here, \mathfrak{t} , \mathfrak{v} and \mathfrak{h} are clearly approximations to the kinetic, potential, and total energies, \mathfrak{T} , \mathfrak{V} and \mathfrak{H} , respectively, for the continuous problem, as defined in (3.7). All are time series, and could be indexed as \mathfrak{t}^n , \mathfrak{v}^n and \mathfrak{h}^n , though it should be kept in mind that the above expressions are centered about time instants $t = (n - 1/2)k$.

Eq. (3.21) implies that the discrete time series \mathfrak{h}^n , which approximates the total energy \mathfrak{H} for the continuous time problem, is constant. Thus the scheme (3.12) is energy conserving, in a discrete sense. Thus,

$$\mathfrak{h}^n = \text{constant} = \mathfrak{h}^0 \tag{3.23}$$

Though one might think that the existence of a discrete conserved energy would immediately imply good behaviour of the associated finite difference scheme, it is important to note that the discrete approximation to potential energy, given above in (3.22), is of indeterminate sign, and thus, possibly, so is \mathfrak{h} . The determination of non-negativity conditions on \mathfrak{h} leads directly to a numerical stability condition. Expanding out the operator notation, one may write the function \mathfrak{h}^n at time step n as

$$\mathfrak{h}^n = \frac{1}{2k^2} ((u^n)^2 + (u^{n-1})^2) + \left(\frac{\omega_0^2}{2} - \frac{1}{k^2} \right) u^n u^{n-1}$$

which is no more than a quadratic form in the variables u^n and u^{n-1} . Applying the results given in §2.4.3, one may show that the quadratic form above is positive definite when

$$k < \frac{2}{\omega_0}$$

which is exactly the condition (3.17) arrived at through frequency domain analysis. See Problem 2.12. Under this condition, one further has that

$$|u^n| \leq \sqrt{\frac{2k^2 \mathfrak{h}^n}{1 - (1 - \omega_0^2 k^2 / 2)^2}} = \sqrt{\frac{2k^2 \mathfrak{h}^0}{1 - (1 - \omega_0^2 k^2 / 2)^2}} \tag{3.24}$$

Just as in the case of frequency domain analysis, the solution may be bounded in terms of the initial conditions (or the initial energy) and the system parameters. In fact, the bound above is identical to that given in (3.19). See Problem 3.5.

It is important to make some comments about the nature of this numerical energy conservation

property. In infinite precision arithmetic, the quantity \mathfrak{h}^n remains exactly constant, according to (3.23). That is, the discrete kinetic energy \mathfrak{t}^n and potential energy \mathfrak{v}^n sum to the same constant, at any time step n . See Figure 3.4(a). But in a finite precision computer implementation, there will be fluctuations in the total energy at the level of the least significant bit; such a fluctuation is shown, in the case of floating point arithmetic, in Figure 3.4(b). In this case, the fluctuations are on the order of 10^{-15} of the total energy, and the bit-quantization of these variations is clearly visible. Numerical energy is conserved to machine accuracy.

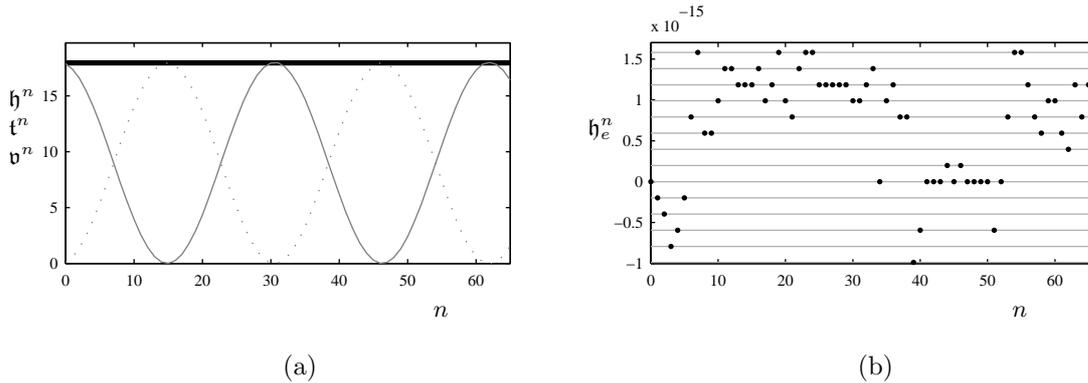


Figure 3.4: (a) Variation of the discrete potential energy \mathfrak{v}^n (solid grey line), discrete kinetic energy \mathfrak{t}^n (dotted grey line) and total discrete energy \mathfrak{h}^n (solid black line), plotted against time step n , for the output of scheme (3.12). In this case, the values $\omega_0 = 1$ and $k = 1/10$ were used, and scheme (3.12) was initialized with the values $u^0 = 0.2$ and $u^1 = -0.4$. (b) Variation of the error in energy \mathfrak{h}_e^n , defined, at time step n , as $\mathfrak{h}_e^n = (\mathfrak{h}^n - \mathfrak{h}^0) / \mathfrak{h}^0$, plotted as black points. Multiples of single bit variation are plotted as grey lines.

The analysis of conservative properties of finite difference schemes is younger than frequency domain analysis, and, as a result, much less well-known. It is, in some respects, far more powerful than the body of frequency domain analysis techniques which forms the workhorse of scheme analysis. The energy method [113] is the name given to a class of analysis techniques which do not employ frequency domain concepts. The study of finite difference schemes which possess conservation properties (of energy, and other invariants) experienced a good deal of growth in the 1980s [112, 221] and 1990s [159, 282, 102], and is related to more modern symplectic integration techniques [222, 232].

3.2.7 Computational Considerations

The computational requirements of scheme (3.12) are most easily seen through an examination of the Matlab code in §A.1. In each updating cycle, the algorithm requires one multiplication, and one addition. Interestingly, this is nearly the same amount of computation required to generate values of a sinusoid through table reading, as is common practice in sound synthesis applications [170], as discussed in §1.1.3, but with the advantage of a much reduced cost in terms of memory, as mentioned below. The use of recursive structures as a generator of sinusoids has been discussed by Smith and Cook [243]. In fact, with a small amount of additional work, a scheme can be developed which calculates exact values of the sinusoid—see §3.3.4.

Though, for the sake of clarity of presentation, the algorithm in §A.1 appears to make use of three memory locations, one for the current value of the computed solution, and two for the previous

two values, this may be reduced to two locations through overwriting. See Programming Exercise 3.1.

3.3 Other Schemes

Difference scheme (3.12) is but the simplest form of approximation to (3.1), and, though it is indeed extremely useful, it does suffer from frequency warping error—the sinusoid generated by the scheme is of a frequency ω which may be different from ω_0 , the frequency of the exact solution. The degree of warping becomes more pronounced for high values of ω_0 relative to $2\pi f_s$, where f_s is the sample rate. In this section, various techniques for the construction of improved finite difference schemes are presented.

3.3.1 Using Time-averaging Operators

Another way to approximate (3.1) involves the use of temporal averaging operators, as introduced in §2.2. For example,

$$\delta_{tt}u = -\omega_0^2 \mu_t u \quad (3.25)$$

is also a second order accurate approximation to (3.1), where it is to be recalled that the averaging operator μ_t approximates the identity operation. Second order accuracy is easy enough to determine by inspection of the centered operators δ_{tt} and μ_t , which are second order accurate approximations to $\frac{d^2}{dt^2}$ and the identity operation, respectively. See §2.2.3.

This scheme, when the action of the operators δ_{tt} and μ_t is expanded out, leads to

$$\frac{1}{k^2} (u^{n+1} - 2u^n + u^{n-1}) = -\frac{\omega_0^2}{2} (u^{n+1} + u^{n-1}) \quad (3.26)$$

which again involves values of the time series at three levels $n+1$, n , and $n-1$, and which can again be solved for u^{n+1} , as

$$u^{n+1} = \frac{2}{1 + \omega_0^2 k^2 / 2} u^n - u^{n-1} \quad (3.27)$$

The characteristic polynomial is now

$$z - \frac{2}{1 + \omega_0^2 k^2 / 2} + z^{-1} = 0$$

which has roots

$$z_{\pm} = \frac{1 \pm \sqrt{1 - (1 + \omega_0^2 k^2 / 2)^2}}{1 + \omega_0^2 k^2 / 2}$$

The solution will again evolve according to (3.16) when z_+ and z_- are distinct.

In this case, however, it is not difficult to show that the roots z_{\pm} will be complex conjugates of unit magnitude for any choice of k . Thus $z_{\pm} = e^{\pm j\omega k}$, where ω , the frequency of the sinusoid generated by the scheme, is given by

$$\omega = \frac{1}{k} \cos^{-1} \left(\frac{1}{1 + \omega_0^2 k^2 / 2} \right)$$

Again, $\omega \neq \omega_0$, but, in contrast to the case of scheme (3.12), one now has $\omega < \omega_0$. The result is a well-behaved sinusoidal solution of the form of (3.18) for any value of k ; there is no stability condition of the form of (3.17). (This is typical of some types of implicit schemes for partial differential equations—see §6.3.2.)

The energetic analysis mirrors this behaviour. Multiplying (3.25) by $\delta_t u$, and using identities

(2.22a) and (2.22e) gives

$$(\delta_t u)(\delta_{tt} u) + \omega_0^2 (\delta_t u)(\mu_t u) = \delta_{t+} \left(\frac{1}{2} (\delta_{t-} u)^2 + \frac{\omega_0^2}{2} \mu_{t-} (u^2) \right) = 0$$

or

$$\delta_{t+} \mathfrak{h} = 0 \quad \text{with} \quad \mathfrak{h} = \mathfrak{t} + \mathfrak{v} \quad \text{and} \quad \mathfrak{t} = \frac{1}{2} (\delta_{t-} u)^2 \quad \mathfrak{v} = \frac{\omega_0^2}{2} \mu_{t-} u^2 \quad (3.28)$$

Now, \mathfrak{t} and \mathfrak{v} , and as a result \mathfrak{h} are non-negative for any choice of k . A bound on solution size follows immediately. See Problem 3.6.

3.3.2 A Second-order Family of Schemes

Though the stability condition inherent in scheme (3.12) has been circumvented in scheme (3.25), the problem of warped frequency persists. Using combination properties of the averaging operators, discussed in §2.2.2, it is not difficult to see that any scheme of the form

$$\delta_{tt} u = -\omega_0^2 (\alpha + (1 - \alpha) \mu_t) u \quad (3.29)$$

will also be a second-order accurate approximation to (3.1), for any choice of the real parameter α . In fact, it would be better to say “at least second-order accurate”—see §3.3.4 for more discussion. There is thus a one-parameter family of schemes for (3.1), all of which operate as two step recursions: written out in full, the recursion has the form

$$u^{n+1} = \frac{2 - \alpha \omega_0^2 k^2}{1 + (1 - \alpha) \omega_0^2 k^2 / 2} u^n - u^{n-1}$$

Schemes (3.12) and (3.25) are members of this family with $\alpha = 1$ and $\alpha = 0$, respectively.

The stability condition for the family of schemes (3.29) is now dependent on the free parameter α . Using frequency domain techniques (see Problem 3.7), it may be shown that this condition is

$$k < \frac{2}{\omega_0 \sqrt{2\alpha - 1}} \quad \text{for} \quad \alpha \geq \frac{1}{2}, \quad \text{otherwise stable} \quad (3.30)$$

The scheme (3.29) also possesses a conserved energy, i.e.,

$$\delta_{t+} \mathfrak{h} = 0 \quad \text{with} \quad \mathfrak{h} = \mathfrak{t} + \mathfrak{v} \quad \text{and} \quad \mathfrak{t} = \frac{1}{2} (\delta_{t-} u)^2 \quad \mathfrak{v} = \frac{\omega_0^2}{2} (\alpha u e_{t-} u + (1 - \alpha) \mu_{t-} (u^2)) \quad (3.31)$$

Finding a condition such that this energy is non-negative is again equivalent to the stability bound above obtained using frequency domain techniques. See Problem 3.7.

Frequency Warping

It is interesting to examine the frequency warping characteristics of various members of this family, as function of ω_0 , the reference frequency for the simple harmonic oscillator. In Figure 3.5(a), the frequency ω of the scheme (3.29) is plotted as a function of ω_0 , for different choices of α . Notice that for small values of α , lower than about $\alpha = 0.7$, the difference scheme frequency is artificially low, and for values closer to $\alpha = 1$, artificially high. In fact, for high values of α , there is a certain “cutoff” frequency ω_0 , for which the difference scheme becomes unstable—this is readily visible in the figure. In the middle range of values of α (between about $\alpha = 0.7$ and $\alpha = 0.8$), the difference scheme yields a good approximation over nearly the entire range of frequencies ω_0 up to the Nyquist limit. It is useful, from a musical point of view, to plot the frequency deviation in terms of cents, defined as

$$\text{deviation in cents} = 1200 \log_2 \left(\frac{\omega}{\omega_0} \right)$$

A deviation of 100 cents corresponds to a musical interval of a semitone. See Figure 3.5(b). For $\alpha = 0.7$, the maximum detuning is approximately two semitones, and, if the sample rate is chosen sufficiently high, occurs in a region of the spectrum for which frequency discrimination in humans may not be of great importance [291]. See Programming Exercise 3.3. Note also the limited range of frequencies which can be simulated, for larger values of α , reflecting increased strictness of the stability condition for the family of schemes, from (3.30).

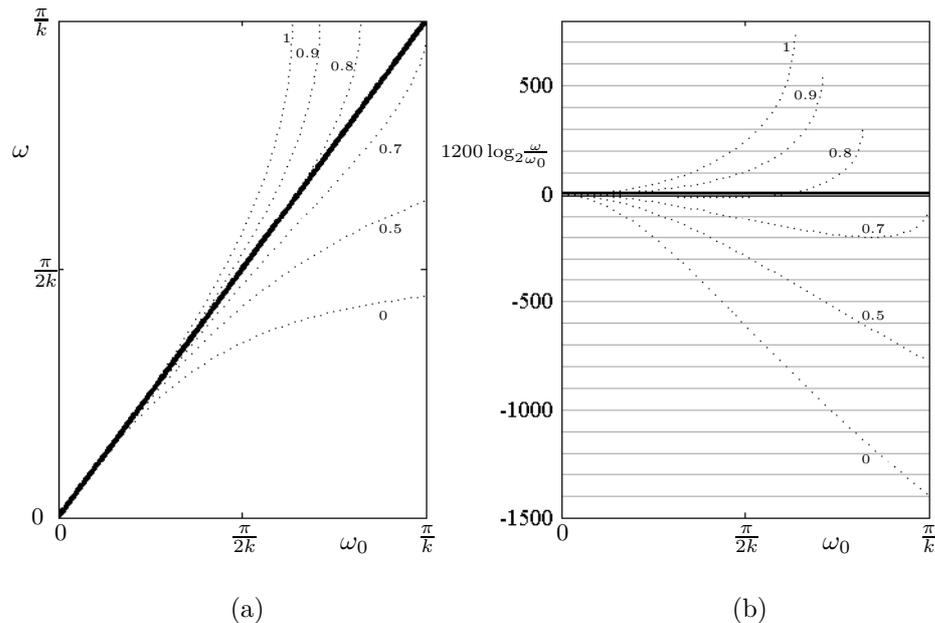


Figure 3.5: (a), Frequency ω of the parameterized difference scheme (3.29) for the simple harmonic oscillator, plotted against $\omega_0 \in [0, \pi/k]$ as dotted lines, for various values of the free parameter α (indicated in the figure). The reference frequency is plotted as a solid black line. (b), deviation in cents of ω from ω_0 for the same members of the family, as a function of $\omega_0 \in [0, \frac{\pi}{k}]$. Zero deviation is plotted as a solid black line. Successive deviations of a musical semitone are indicated by grey horizontal lines.

As mentioned earlier, the effect of frequency warping in discrete time simulations of lumped systems generalizes to the issue of numerical dispersion, and thus inharmonicity in schemes for partial differential equations. Though, as will be seen shortly, in §3.3.4, the problem may be dealt with neatly in the case of the oscillator in isolation, in general, the only cure is through the use of more carefully designed schemes, such as those presented in this section.

3.3.3 Wave Digital Filters

Because scattering methods have played such a dominant role in physical modeling sound synthesis for the past twenty years, it is worth making a short detour at this point to see how they fit into the standard framework of numerical methods. Wave digital filters [88, 87] were the first such method to appear, though not, admittedly, in the context of musical sound synthesis—see §1.2.4 for some general comments on the history of such methods. Though based on ideas from circuit and network theory, in the end, they can be rather simply viewed as finite difference schemes.

Consider a parallel combination of an inductor and a capacitor, as shown in Figure 3.6(a). An electrical network theorist would describe this system in the following way: supposing that the voltage across and current through the inductor (of inductance L) are u_1 and i_1 , and the voltage across and current through the capacitor (of capacitance C) are u_2 and i_2 , then the following four equations describe the dynamics of the parallel combination:

$$u_1 = L \frac{di_1}{dt} \quad i_2 = C \frac{du_2}{dt} \quad u_1 = u_2 \quad i_1 + i_2 = 0 \quad (3.32)$$

The first two equations describe the internal behaviour of the circuit elements themselves, and the latter two are Kirchhoff's connection rules for a parallel combination of two circuit elements. The above system (3.32) of four equations in four unknowns can be reduced to the simple harmonic oscillator (3.1) with $\omega_0 = 1/\sqrt{LC}$, with a dependent variable which is any of u_1 , u_2 , i_1 or i_2 . (The analogy of the electrical network with an acoustical system such as that which might arise in a musical instrument model is direct: replacing currents with velocities, and voltages with forces, and the inductance and capacitance with mass and the inverse of stiffness respectively, one arrives immediately at a simple mass-spring system.)

The wave digital discretization procedure relies on essentially two manipulations. The first is the discrete approximation of the first two of (3.32) through the trapezoid rule (see page 41); in terms of the difference operators that have been defined in this book, one may write these discretized equations as

$$\delta_{t+} u_1 = L \mu_{t+} i_1 \quad \delta_{t+} i_2 = C \mu_{t+} u_2 \quad (3.33)$$

or, expanding out this notation and introducing a time index n ,

$$u_1^{n+1} - u_1^n = \frac{kL}{2} (i_1^{n+1} + i_1^n) \quad i_2^{n+1} - i_2^n = \frac{kC}{2} (u_2^{n+1} + u_2^n) \quad (3.34)$$

The connection rules are assumed to hold instantaneously in discrete time. The second manipulation is the introduction of wave variables:

$$a_1 = \frac{u_1 + R_1 i_1}{2\sqrt{R_1}} \quad b_1 = \frac{u_1 - R_1 i_1}{2\sqrt{R_1}} \quad a_2 = \frac{u_2 + R_2 i_2}{2\sqrt{R_2}} \quad b_2 = \frac{u_2 - R_2 i_2}{2\sqrt{R_2}}$$

Here, two positive constants, R_1 and R_2 , called port resistances have been assigned to the inductor and capacitor, respectively. Under the special choices of $R_1 = 2L/k$ and $R_2 = k/(2C)$, the recursions (3.34) reduce to

$$b_1^n = -a_1^{n-1} \quad b_2^n = a_2^{n-1} \quad (3.35)$$

These are the defining equations of the wave digital inductor and capacitor, respectively—notice in

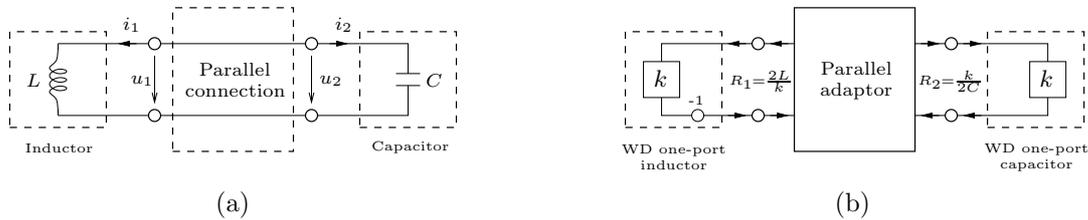


Figure 3.6: *The LC harmonic oscillator—(a) a parallel connection of an inductor, of inductance L and a capacitor of capacitance C , and (b) the corresponding wave digital network.*

particular that they involve only shifting operations (with sign inversion in the case of the inductor), and are thus strictly causal. They do, however, act in isolation, and must be connected through the equations which result from the substitution of wave variables into the connection rules:

$$\begin{bmatrix} a_1^n \\ a_2^n \end{bmatrix} = \begin{bmatrix} -1 + \frac{2R_2}{R_1+R_2} & \frac{2\sqrt{R_1R_2}}{R_1+R_2} \\ \frac{2\sqrt{R_1R_2}}{R_1+R_2} & -1 + \frac{2R_1}{R_1+R_2} \end{bmatrix} \begin{bmatrix} b_1^n \\ b_2^n \end{bmatrix} \quad (3.36)$$

This operation, called scattering, is at the heart of wave digital filtering, and is viewed as a two-port element in its own right, called an adaptor. Notice in particular that the matrix operation above is orthogonal, preserving the “energy,” or l_2 norm of wave variables through the scattering operation². The scattering formulation may be extended to connection of an many elements as desired, all the while maintaining this property of orthogonality, or losslessness. The wave digital discretization of the parallel inductor/capacitor combination is complete, and is illustrated in Figure 3.6(b); in each step in the recursion that the structure implies, there will occur a scattering step (at a wave digital adaptor), and a shifting of data. Such a mode of operation is also characteristic of digital waveguide networks.

One of the nice features of wave digital networks is that the energetic properties are fundamentally built in to the scattering and shifting operations. For example, using the orthogonality of the scattering matrix, in (3.36) and the simple shifting operations above in (3.35), it is simple to show, in this case, that

$$(a_1^n)^2 + (a_2^n)^2 = (b_1^n)^2 + (b_2^n)^2 = (a_1^{n-1})^2 + (a_2^{n-1})^2$$

or,

$$(a_1^n)^2 + (a_2^n)^2 = \text{constant} \quad (3.37)$$

implying losslessness, using wave variables. This should come as no surprise—combining the two recursions in (3.33) leads immediately to the difference scheme (in, say, $u = u_1 = u_2$)

$$\delta_{tt}u = -\omega_0^2\mu_{tt}u$$

which is a member of the family (3.29) of schemes for the simple harmonic oscillator, with $\alpha = 1/2$ and $\omega_0^2 = 1/(LC)$. As was shown earlier, this scheme possesses a conserved energy, given in (3.31); this is in fact identical to the conserved quantity for the wave digital network, from (3.37). (Notice, however, that scheme (3.29) with $\alpha = 0.5$ behaves quite poorly with regard to frequency warping, as illustrated in Figure 3.5.)

Energy conservation, and the good stability properties that follow from it have been a primary selling point of wave digital filtering methods; another is the manner in which a complex system may be broken down into modular components, which behave strictly causally, leading to fully explicit methods. But, as has been seen throughout this chapter, at least in the case of the simple harmonic oscillator, energy conservation follows from the simplest imaginable difference schemes; in fact, one need not restrict oneself to the use of a particular numerical integration method (such as the trapezoid rule) in order to obtain this property. Indeed, the entire apparatus of network theory and scattering is not necessary to show such conservation properties, or numerical stability. The modularity argument is also somewhat misleading—though for certain circuit networks, it is indeed possible to arrive at fully explicit methods using wave digital discretization principles,

²Though power-normalized waves, which give rise to orthogonal scattering matrices, have been used here, it is by no means necessary to use such wave variables. Simple “voltage waves” are in fact the norm in most applications of wave digital filters, including sound synthesis. In this case, scattering matrices are no longer generally orthogonal, but rather orthogonal with respect to a weighting of the port resistances—energy conservation holds in this case as well. For nonlinear networks, however, power-normalized waves are necessary in order to ensure passivity. See, e.g., [89, 90, 24].

there are many (in particular those involving nonlinearities) for which this is rather strenuous—a delicate juggling of port-resistances, through the use of so-called reflection-free ports [91] will be necessary. This approach has been taken by some in the sound synthesis applications—see §1.2.4 on hybrid methods. Often, however, simple finite difference schemes are explicit, such as those for the simple harmonic oscillator, as discussed in this chapter. In short, wave digital filters are intuitively appealing, but somewhat restrictive, and perhaps overly complicated, and do not lead to any real gains in terms of efficiency or ease of analysis, at least for simulation purposes. On the other hand, if one is making use of digital waveguides, which are genuinely more efficient than finite difference schemes for a certain range of applications, then wave digital filters are an excellent match when used in order to model, say waveguide terminations, or connections with lumped elements.

3.3.4 An Exact Solution

For this very special case of the simple harmonic oscillator, there is in fact a two-step recursion which generates the exact solution to (3.1), at times $t = nk$. It is simply given by

$$u^{n+1} - 2 \cos(\omega_0 k) u^n + u^{n-1} = 0 \quad (3.38)$$

This recursion is perhaps more familiar to electrical and audio engineers as a two-pole filter operating under transient conditions. The z transformation analysis reveals:

$$z - 2 \cos(\omega_0 k) + z^{-1} = 0$$

which has solutions

$$z_{\pm} = \cos(\omega_0 k) \pm j \sin(\omega_0 k) = e^{\pm j \omega_0 k} \quad \rightarrow \quad \omega = \omega_0$$

Thus the oscillation frequency ω of recursion (3.38) is exactly ω_0 , the frequency of the continuous time oscillator (3.1).

In a sense, then, all the preceding analysis of difference schemes is pointless, at least in the case of (3.1). On the other hand, the simple harmonic oscillator is a very special case; more complex systems, especially when nonlinear, rarely allow for exact numerical solutions. One other case of interest, however, and the main reason for dwelling on this point here, is the 1D wave equation, to be discussed in Chapter 6, which is of extreme practical importance in models of musical instruments which are essentially one-dimensional, such as strings and acoustic tubes. Numerical methods which are exact also exist in this case, and have been exploited with great success as digital waveguides [242].

Another question which arises here is that of accuracy. Considering again the one-parameter family of two-step difference schemes given by (3.29). Using identity (2.7a) relating μ_t to δ_{tt} , it may be rewritten as

$$\delta_{tt} u = \frac{-\omega_0^2}{1 + \frac{\omega_0^2(1-\alpha)k^2}{2}} u$$

Under the special choice of $\alpha = \frac{2}{\omega_0^2} - \frac{\cos(\omega_0 k)}{1 - \cos(\omega_0 k)}$, the difference scheme becomes exactly (3.38), or

$$\underbrace{\left(\delta_{tt} + \frac{2}{k^2} (1 - \cos(\omega_0 k)) \right)}_P u = 0 \quad (3.39)$$

Consider the action of the operator P as defined above on a continuous function. Expanding the

operator δ_{tt} and the function $\cos(\omega_0 k)$ in Taylor series leads to

$$P = \sum_{l=1}^{\infty} \frac{2k^{2(l-1)}}{(2l)!} \frac{d^{2l}}{dt^{2l}} + (-1)^{l-1} \frac{2\omega_0^{2l} k^{2(l-1)}}{(2l)!} = \sum_{l=1}^{\infty} \frac{2k^{2(l-1)}}{(2l)!} \left(\frac{d^{2l}}{dt^{2l}} + (-1)^{l-1} \omega_0^{2l} \right)$$

The various terms of the form $\left(\frac{d^{2l}}{dt^{2l}} + (-1)^{l-1} \omega_0^{2l} \right)$ all possess a factor of $\frac{d^2}{dt^2} + \omega_0^2$. Thus (3.39) may be rewritten as

$$(1 + O(k^2)) \left(\frac{d^2}{dt^2} + \omega_0^2 \right) u = 0$$

which shows that the solution u indeed solves the equation of the simple harmonic oscillator exactly.

This property of accuracy of a difference scheme beyond that of the constituent operators, under very special choices of the scheme parameters, is indeed a very delicate one. In the distributed setting, it has been exploited in the construction of so-called modified equation methods [126, 231, 181, 182, 69] and compact spectral-like schemes [157, 289, 143, 161].

3.3.5 Further Methods

In this book, relatively simple difference strategies will be employed whenever possible. On the other hand, sometimes more accurate methods are necessary, especially when one is interested in more delicate modeling of musical instrument physics. There is a large variety of time differencing methods which are in use for the solutions of ODEs; normally, these are framed in terms of a first-order system, and so do not fit well into the main development here. Some of the better known are Adams-Bashforth, Adams-Moulton and Runge-Kutta methods, and will be discussed in the context of spectral methods in §15.5.

There do exist, however, some families of methods which have been developed directly for use with second order systems, generally for application to problems in structural dynamics, such as those which occur in musical acoustics—among these are the Newmark family of schemes [179], as well as those proposed by Hilber, Hughes and Taylor [118].

3.4 Lumped Mass-Spring Networks

Similar principles of difference approximation may be applied to coupled systems of oscillators, introduced in §3.1.4. For an N mass system defined by (3.9), one explicit approximation is

$$\mathbf{M}\delta_{tt}\mathbf{u} = -\mathbf{K}\mathbf{u} \quad (3.40)$$

where now, $\mathbf{u} = \mathbf{u}^n$ is an N -vector time series. The resulting recursion is

$$\mathbf{u}^{n+1} = (2\mathbf{I} - k^2\mathbf{M}^{-1}\mathbf{K})\mathbf{u}^n - \mathbf{u}^{n-1}$$

where \mathbf{I} is the identity matrix. Stability analysis is similar to the scalar case; employing the ansatz $\mathbf{u}^n = \boldsymbol{\phi}z^n$ gives the following eigenvalue equation:

$$k^2\mathbf{M}^{-1}\mathbf{K}\boldsymbol{\phi} = -(z - 2 + z^{-1})\boldsymbol{\phi}$$

This equation possesses N solutions z ; the condition that the scheme above is stable is similar to that of the scheme (3.12) for the simple harmonic oscillator—in order to ensure that all roots z are bounded by 1 in magnitude, the condition

$$k \leq \frac{2}{\sqrt{\max(\text{eig}(\mathbf{M}^{-1}\mathbf{K}))}} \quad (3.41)$$

must be satisfied. Energy principles may be applied to scheme (3.40) to yield the same result. See Problem 3.9.

Though scheme (3.40) is explicit, and thus simple to implement, one difficulty is that of stiffness—if the natural frequencies of the coupled system are widely separated, then it will be necessary to use a relatively small time step (or high sampling frequency) in order to compute a solution. The flip side of this difficulty, in the realm of sound synthesis, where the sampling frequency is fixed, is that one can expect a good degree of frequency warping if the natural frequencies are in the upper range of the audio spectrum. One remedy is to apply a parameterized scheme such as that discussed in §3.3.2. For the coupled system (3.9), this will have the form

$$\mathbf{M}\delta_{tt}\mathbf{u} = -\mathbf{K}(\alpha + (1 - \alpha)\mu_t)\mathbf{u} \quad (3.42)$$

and will yield a much lower degree of frequency warping over the entire audio spectrum than scheme (3.40) for a properly chosen value of the parameter α . In the case of the single oscillator, there is no computational disadvantage to using such a scheme, but in the case of a coupled system, when \mathbf{K} is not diagonal (this is the case for any interesting problem), linear system solution techniques will be necessary in order to perform an update. Though, for sparse \mathbf{K} , many fast iterative techniques are available for linear system solution, computational cost is certainly heavier than for an explicit scheme, and programming complexity is undeniably greater. The deeper issue is that inherent to all implicit schemes: for most reasonably complex systems in musical acoustics, better accuracy usually comes with a price. Implicit methods (i.e., those involving a coupling among unknowns in a difference scheme update) will be revisited in more detail in the distributed setting—see §6.3.2.

Another possibility is to diagonalize system (3.9) before discretization, or, for the coupled system (3.9), work with (3.10). In this case, the system consists of an uncoupled set of scalar oscillators, and one may employ a distinct exact scheme (see §3.3.4) to each component of the solution. This is an example of a modal method, and has the virtue of generating an exact solution with an explicit method. This is the approach taken by many in the finite element community, and also by

3.5 Loss

Damping intervenes in any musical system, and, in the simplest case, may be modelled through the addition of a “linear loss” term to a given system. In the case of the simple harmonic oscillator, one may add damping as

$$\frac{d^2u}{dt^2} = -\omega_0^2u - 2\sigma_0\frac{du}{dt} \quad (3.43)$$

where $\sigma_0 \geq 0$ is the damping parameter for the system. In terms of the mass-spring system and LC oscillator shown in Figure 3.1, the damping term corresponds to the addition of a linear dashpot and resistor, respectively.

The characteristic equation for (3.43), obtained through Laplace transformation, or through the insertion of a test solution $u = e^{st}$, is

$$s^2 + 2\sigma_0s + \omega_0^2 = 0$$

which has solutions

$$s_{\pm} = -\sigma_0 \pm \sqrt{\sigma_0^2 - \omega_0^2}$$

If damping is small, i.e., if $\sigma_0 < \omega_0$, the roots above simplify to

$$s_{\pm} = -\sigma_0 \pm j\omega_1 \quad \text{with} \quad \omega_1 = \sqrt{\omega_0^2 - \sigma_0^2}$$

In this case, the solution (3.3) may be generalized to

$$u(t) = e^{-\sigma_0 t} (A \cos(\omega_1 t) + B \sin(\omega_1 t)) = C_1 e^{-\sigma_0 t} \cos(\omega_1 t + \phi_1) \quad (3.44)$$

where A and B are defined by

$$A = u_0 \quad B = \frac{v_0 + \sigma_0 u_0}{\omega_1} \quad C_1 = \sqrt{A^2 + B^2} \quad \phi_1 = \tan^{-1}\left(-\frac{B}{A}\right)$$

See Figure 3.7.

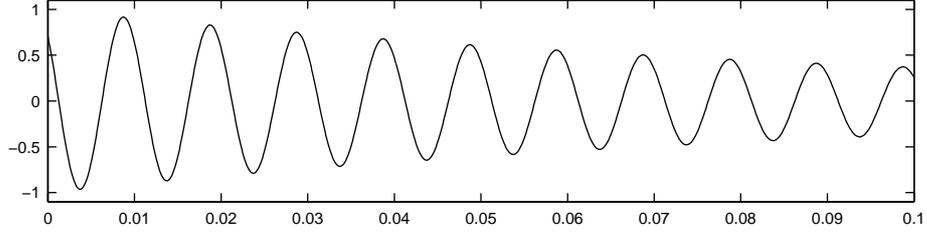


Figure 3.7: *Damped sinusoid, of the form given in (3.44), with $C_1 = 1$, $\sigma_0 = 10$, $\omega_1 = 200\pi$, and $\phi_0 = \pi/4$.*

It should be clear that the bound

$$|u(t)| \leq C_1 e^{-\sigma_0 t} \leq C_1 \quad (3.45)$$

holds for all $t \geq 0$.

The case of large damping, i.e., $\sigma_0 \geq \omega_0$ is of mainly academic interest in musical acoustics. The main point here is that for $\sigma_0 > \omega_0$, the solution will consist of two exponentially damped terms. For $\sigma_0 = \omega_0$, degeneracy of the roots of the characteristic solution leads to some limited solution growth. See Problem 3.11.

3.5.1 Energy

The energetic analysis of (3.43) is a simple extension of that for system (3.1). One now has, after again multiplying through by $\frac{du}{dt}$,

$$\frac{d\mathfrak{H}}{dt} = -\sigma_0 \left(\frac{du}{dt}\right)^2 \leq 0 \quad \rightarrow \quad \mathfrak{H}(t_1) \leq \mathfrak{H}(t_2) \leq \mathfrak{H}(0) \quad \text{for } t_1 \geq t_2 \geq 0 \quad (3.46)$$

where \mathfrak{H} is defined as in (3.7). The energy is thus a positive and monotonically decreasing function of time. This leads to the bound

$$|u(t)| \leq C_0 \quad (3.47)$$

which is identical to that obtained in the lossless case.

Modified Energy Functions

The bounds (3.45) and (3.47) obtained through frequency domain and energetic analysis, respectively, are distinct. Notice, in particular, that bound (3.47) is insensitive to the addition of loss, and, at the same time, is more general than bound (3.45), which requires the assumption of small damping (i.e., $\sigma_0 < \omega_0$). It is possible to reconcile this difference by extending the energetic analysis in the following way.

First, note that although $\mathfrak{H}(t)$ will be monotonically decreasing in the case of linear loss, by (3.46), the specifics of its rate of decay are unclear. To this end, define the function $\bar{\mathfrak{H}}$ by

$$\bar{\mathfrak{H}} = \mathfrak{H} + \sigma_0 u \frac{du}{dt} = \frac{1}{2} \left(\frac{du}{dt} \right)^2 + \frac{\omega_0^2}{2} u^2 + \sigma_0 u \frac{du}{dt} \quad (3.48)$$

Note that $\bar{\mathfrak{H}}$ approaches \mathfrak{H} in the limit as $\sigma_0 \rightarrow 0$. It is simple to show that

$$\frac{d\bar{\mathfrak{H}}}{dt} = -\sigma_0 \bar{\mathfrak{H}} \quad \rightarrow \quad \bar{\mathfrak{H}}(t) = e^{-\sigma_0 t} \bar{\mathfrak{H}}(0) \quad (3.49)$$

Thus $\bar{\mathfrak{H}}$ decays exponentially, at decay rate σ_0 , for any value of σ_0 . See Problem 3.12.

On the other hand, $\bar{\mathfrak{H}}$, in contrast to \mathfrak{H} , is not necessarily a non-negative function of the state u and du/dt . This non-negativity property is crucial in obtaining bounds such as (3.8). To this end, it is worth determining the conditions under which $\bar{\mathfrak{H}}$ is non-negative. First note that

$$u \frac{du}{dt} \geq -\frac{1}{2} \left(\alpha u^2 + \frac{1}{\alpha} \left(\frac{du}{dt} \right)^2 \right)$$

for any real constant $\alpha > 0$. Choosing $\alpha = \sigma_0$ leads to

$$\bar{\mathfrak{H}} \geq \frac{\omega_0^2 - \sigma_0^2}{2} u^2$$

which is clearly non-negative under the condition of low loss $\sigma_0 < \omega_0$ which was assumed in the frequency domain analysis earlier in this section. Under this condition, one has, from the above inequality, that

$$|u(t)| \leq \frac{1}{\omega_1} \sqrt{2\bar{\mathfrak{H}}(t)} \leq \frac{1}{\omega_1} \sqrt{2\bar{\mathfrak{H}}(0)} \quad \text{for } t \geq 0$$

The bounding quantity on the right of the above inequality is in fact identical to C_1 , and thus the bound above is identical to (3.45) obtained using frequency domain analysis.

In general, true energy functions such as \mathfrak{H} for physical systems are non-negative, and the introduction of modified energetic quantities such as $\bar{\mathfrak{H}}$, for which non-negativity conditions must be determined, might seem to be a needless complication. On the other hand, for numerical approximations, as will be seen shortly, the discrete equivalent of the system energy is not necessarily non-negative, and the determination of non-negativity conditions leads to numerical stability conditions. Thus the simple analysis above serves as a useful example, in miniature, of what will follow in this book. Even in the discrete case, it will sometimes be useful to introduce modified energetic quantities such as $\bar{\mathfrak{H}}$, especially when dealing with certain types of non-centered finite difference schemes.

3.5.2 Finite Difference Scheme

The damped oscillator (3.43) may be dealt with in a variety of ways. A simple scheme is the following:

$$\delta_{tt}u = -\omega_0^2 u - 2\sigma_0 \delta_t u \quad (3.50)$$

which, when the operator notation is expanded out, leads to the recursion

$$u^{n+1} = \frac{2 - \omega_0^2 k^2}{1 + \sigma_0 k} u^n - \frac{1 - \sigma_0 k}{1 + \sigma_0 k} u^{n-1}$$

This is again a recursion of temporal width three.

The characteristic equation will now be

$$z - \frac{2 - \omega_0^2 k^2}{1 + \sigma_0 k} + \frac{1 - \sigma_0 k}{1 + \sigma_0 k} z^{-1} = 0 \quad (3.51)$$

The roots may be written explicitly as

$$z_{\pm} = \frac{1}{1 + \sigma_0 k} \left(1 - \frac{\omega_0^2 k^2}{2} \pm \sqrt{(1 - \omega_0^2 k^2 / 2)^2 - (1 - \sigma_0^2 k^2)} \right) \quad (3.52)$$

Though a direct analysis of the above expressions for the roots is possible, if one is interested in bounding the magnitudes of the roots by unity, it is much simpler to employ the conditions (2.14), which yield, simply,

$$k \leq \frac{2}{\omega_0} \quad (3.53)$$

The stability condition for scheme (3.50), which involves a centered difference approximation for the loss term, is thus unchanged from that of the undamped scheme (3.12). On the other hand, a more direct examination of the behaviour of the roots given above can be revealing; see Problem 3.13.

Energetic analysis also yields a similar condition. Multiplying (3.50) by $\delta_t u$ gives, instead of (3.21),

$$\delta_{t+} \mathfrak{h} = -2\sigma_0 (\delta_t u)^2 \leq 0 \quad (3.54)$$

where $\mathfrak{h} = \mathfrak{t} + \mathfrak{v}$, with \mathfrak{t} and \mathfrak{v} defined as before in (3.22). From the previous analysis, it remains true that \mathfrak{h} is non-negative under the condition (3.53), in which case one may deduce that

$$\mathfrak{h}^n \leq \mathfrak{h}^{n-1} \leq \mathfrak{h}^0$$

for all $n \geq 1$. The discrete energy function is thus monotonically decreasing. As the expression for the energy is the same as in the lossless case, one may again arrive at a bound on solution size, namely

$$|u^n| \leq \sqrt{\frac{2k^2 \mathfrak{h}^n}{1 - (1 - \omega_0^2 k^2 / 2)^2}} \leq \sqrt{\frac{2k^2 \mathfrak{h}^0}{1 - (1 - \omega_0^2 k^2 / 2)^2}} \quad (3.55)$$

The scheme above makes use of a centered difference approximation to the loss term. Consider the following non-centered scheme:

$$\delta_{tt} u = -\omega_0^2 u - 2\sigma_0 \delta_{t-} u \quad (3.56)$$

The characteristic equation will now be

$$z - (2 - \omega_0^2 k^2 + 2\sigma_0 k) + (1 - 2\sigma_0 k) z^{-1} = 0 \quad (3.57)$$

The condition (2.14) now gives the following condition on k for numerical stability:

$$k \leq \frac{2}{\omega_0} \left(-\sigma_0 + \sqrt{\sigma_0^2 + \omega_0^2} \right) \quad (3.58)$$

See Problem 3.14.

This non-centered scheme is only first order accurate, but, if the loss parameter σ_0 is small, as it normally is for most systems in musical acoustics, the solution accuracy will not be severely degraded. Although in this case, one can arrive at an explicit update using centered or non-centered difference approximations, in the distributed setting, such backward difference approximations can be useful in avoiding implicit schemes, which are generally more costly in terms of implementation—see comments on this topic in §3.4.

3.5.3 Numerical Decay Time

Like frequency, loss is an extremely important perceptual characteristic of a musical system, in that it determines a characteristic decay time; as has been shown in the case of the SHO, frequency can be altered through numerical approximation, and, as one might expect, so can the decay time. It is

thus worth spending a few moments here in determining just how much distortion will be introduced through typical finite difference approximations.

First consider the solution (3.44) to the SHO with loss, under normal (low-loss) conditions. The solution is exponentially damped, and the 60 dB decay time T_{60} (in amplitude) may be defined as

$$T_{60} = \frac{6 \ln 10}{\sigma_0} \quad (3.59)$$

Considering now the simple difference scheme (3.50), if the stability condition (3.53) is respected, and again under low-loss conditions, the roots z_{\pm} of the characteristic polynomial (3.51) will be complex conjugates, of magnitude

$$|z| = e^{-\sigma_d k} = \sqrt{\frac{1 - \sigma_0 k}{1 + \sigma_0 k}}$$

Here, σ_0 is the loss parameter for the model system (3.43), and σ_d is that corresponding to the finite difference scheme. One can then define a numerical 60 dB decay time $T_{d,60}$ by

$$T_{d,60} = \frac{6 \ln 10}{\sigma_d} = \frac{12k \ln 10}{\ln\left(\frac{1 + \sigma_0 k}{1 - \sigma_0 k}\right)} \quad (3.60)$$

For scheme (3.56), for instance, which uses a non-centered approximation to the loss term, the magnitudes of the solutions to the characteristic polynomial (3.57) will be, again under low loss conditions,

$$|z| = e^{-\sigma_d k} = \sqrt{1 - 2\sigma_0 k}$$

implying that

$$T_{d,60} = \frac{6 \ln 10}{\sigma_d} = \frac{12k \ln 10}{-\ln(1 - 2\sigma_0 k)}$$

It is useful to plot the relative decay time $T_{d,60}/T_{60}$ for the schemes mentioned above—see Figure 3.8.

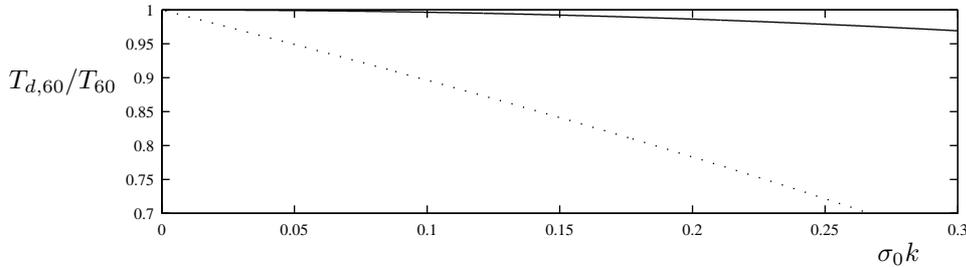


Figure 3.8: The relative 60 dB decay time $T_{d,60}/T_{60}$ for scheme (3.50), as a solid black line, and for scheme (3.56), as a dotted black line, plotted against $\sigma_0 k$.

For scheme (3.50), the numerical decay time is closer than 99% of the true decay time for $\sigma_0 k \leq 0.17$. For a sample rate of 44.1 kHz (implying $k = 1/44100$), this implies that for decay times greater than $T_{60} = 0.0018$ s, the numerical decay time will not be noticeably different than that of the model system. For scheme (3.56), the 99% threshold is met for $T_{60} \geq 0.031$ s. These decay times are quite short by musical standards, and so it may be concluded that numerical distortion of decay time is not of major perceptual relevance, at least at reasonably high sample rates.

Although in the two examples of numerical schemes above, the numerical decay time is independent of ω_0 (as it is for the model system), this is not necessarily always the case. See Problem 3.15.

3.5.4 An Exact Solution

Just as in the case of the SHO, an exact two-step numerical solution is available in the case of the SHO with loss. Considering the recursion

$$u^{n+1} = e^{-\sigma_0 k} \left(e^{\sqrt{\sigma_0^2 - \omega_0^2} k} + e^{-\sqrt{\sigma_0^2 - \omega_0^2} k} \right) u^n - e^{-2\sigma_0 k} u^{n-1} \quad (3.61)$$

the characteristic polynomial is

$$z - e^{-\sigma_0 k} \left(e^{\sqrt{\sigma_0^2 - \omega_0^2} k} + e^{-\sqrt{\sigma_0^2 - \omega_0^2} k} \right) + e^{-2\sigma_0 k} z^{-1}$$

which has roots

$$z_{\pm} = e^{(-\sigma_0 \pm \sqrt{\sigma_0^2 - \omega_0^2}) k}$$

The proof that the difference scheme (3.61) solves the loss SHO exactly may be carried out through the same series expansion methods used in §3.3.4. It is also possible to make a correspondence between scheme (3.61) and a parameterized finite difference scheme—see Problem 3.16.

3.6 Sources

So far in this chapter, only unforced oscillators have been discussed. Though forcing terms almost never appear in the linear case in applications in musical acoustics, they play a rather important role in the case of nonlinear excitation mechanisms involving a continuous supply of energy, such as bows and reed/lip models. See §4.3.1 and §4.3.2.

Considering the case of the oscillator with loss, a force term $F(t)$ may be added as

$$\frac{d^2 u}{dt^2} = -\omega_0^2 u - 2\sigma_0 \frac{du}{dt} + F(t) \quad (3.62)$$

In this case, it will be assumed that the force term $F(t)$ is known, though it should be kept in mind that in certain applications involving connections among objects, the force term may itself be a function of other unknowns in the system. (It should be noted that F , as it appears here and elsewhere in this book does not in fact have dimensions of force, but of acceleration—in the case of a mass spring system, it is the applied force divided by the mass. This, again, is an example of a reduction in the number of parameters necessary to specify a system. f will be used instead of F when a true force is intended.)

Frequency domain analysis of the forced oscillator is carried out in many textbooks—see, e.g., [174]. The main result, when the forcing function is sinusoidal, is that one obtains a solution which is a combination of a steady-state response of the oscillator at the driving frequency, at a magnitude governed by the impedance of the oscillator, and the free or transient response, at the natural frequency of the oscillator—see Figure 3.9.

Energy analysis leads immediately to

$$\frac{d\mathfrak{H}}{dt} = -2\sigma_0 \left(\frac{du}{dt} \right)^2 + \frac{du}{dt} F$$

Using inequality (2.24), with a choice of $\alpha = \frac{1}{\sqrt{4\sigma_0}}$ gives the bound

$$\frac{d\mathfrak{H}}{dt} \leq \frac{F^2}{8\sigma_0} \quad \Rightarrow \quad \mathfrak{H}(t) \leq \mathfrak{H}(0) + \frac{1}{8\sigma_0} \int_0^t F^2 dt'$$

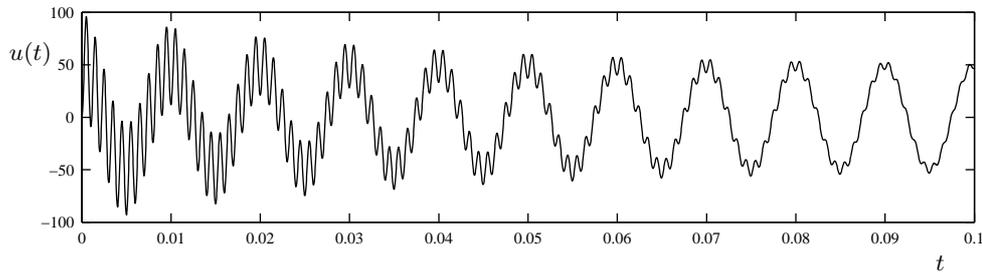


Figure 3.9: Response of the oscillator given in (3.62), with frequency $f_0 = \omega_0/2\pi = 1000$ Hz, and $T_{60} = 0.5$ ss, when the forcing function F is given by $F = \cos(2\pi 100t)$, for $t \geq 0$. The transient response of the system at $f_0 = 1000$ Hz is superimposed onto the steady state response at 100 Hz.

Thus the energy of the system may be bounded in terms of the initial conditions, through $\mathfrak{H}(0)$, and the external forcing function F , both of which are known a priori. This bound is, in most cases, rather conservative, but is useful especially in cases for which the forcing function $F(t)$ is of finite support of short duration. Notice, however, that in order to obtain such a bound, one must have a loss coefficient σ_0 which is strictly positive.

The number of possibilities for finite difference schemes is multiplied considerably by the presence of a forcing term—such is the interest of more involved techniques such as Newmark methods [179]. Keeping with two step methods, one simple set is given by

$$\delta_{tt}u = -\omega_0^2 u - 2\sigma_0 \delta_t u + [F] \quad (3.63)$$

where $[F] = [F]^n$ is some unspecified second-order accurate approximation to $F(t)$, such as F^n , $\mu_t F^n$, $\mu_{tt} F^n$. In the case of the forcing term, there is much less difficulty in making use of wide-stencil approximations, because F is assumed known (though in real-time sound synthesis applications, it may be sampled from data supplied by a performer).

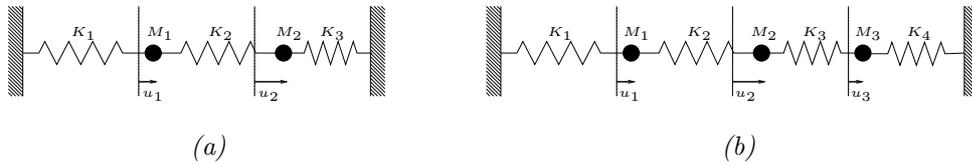
Energy analysis can be applied to a difference scheme in order to find bounds on the size of the solution in terms of the initial conditions, and the time series $[F]$ —see Problem 3.18.

3.7 Problems

Problem 3.1 For the simple harmonic oscillator, find a bound on du/dt in terms of the energy \mathfrak{H} . Use this to find another bound on $u(t)$, namely $|u(t)| \leq a_1 t + a_2$ for $t \geq 0$ and determine the positive constants a_1 and a_2 in terms of initial conditions. See §3.1.2.

Problem 3.2 Illustrated below are (a) a two-mass and (b) a three-mass lumped system. In each case, the masses are constrained to move longitudinally, and the displacement of the l th mass from its rest position is $u_l(t)$, as indicated by distance from the vertical dotted lines. The masses M_1, M_2, \dots , are connected to one another by linear springs, of stiffnesses K_1, K_2, \dots , as indicated in the figure. The series of masses and springs is terminated by rigid supports. Notice in particular that in each of (a) and (b), the masses M_l and spring stiffnesses K_l take on values which are distinct for each l (i.e., they are not all the same).

Given that the forces acting on mass l from the right and left are $K_{l+1}(u_{l+1} - u_l)$ and $-K_l(u_l - u_{l-1})$, respectively, and that the total of all forces acting on mass l must be equal to $M_l d^2 u_l / dt^2$, derive coupled dynamical equations for all the masses. For each of (a) and (b) shown in the figure, write these equations in the vector-matrix form (3.9), with $\mathbf{u} = [u_1, u_2, \dots]^T$, and give the explicit forms of the matrices \mathbf{M} and \mathbf{K} .



Problem 3.3 Consider the two mass and three mass systems of the previous problem, under the simplified conditions that $K_1 = K_2 = \dots = K_0$, and $M_1 = M_2 = \dots = M_0$, for positive constants K_0 and M_0 . For each of cases (a) and (b) shown in the figure, explicitly find the eigenvalues of the matrix $\mathbf{M}^{-1}\mathbf{K}$, and, thus, the frequencies of the system, as per (3.11).

Problem 3.4 For the scheme (3.12), consider the following approach to initialization to second order accuracy. For a continuous time function $u(t)$ solving (3.1), it is true, through a Taylor series expansion, that

$$u(k) = u(0) + k \left. \frac{du}{dt} \right|_{t=0} + \frac{k^2}{2} \left. \frac{d^2u}{dt^2} \right|_{t=0} + O(k^3)$$

Given the values $u(0) = u_0$ and $du/dt|_{t=0} = v_0$, determine, from the above approximation as well as the defining equation of the SHO, $u(k)$ in terms of u_0 and v_0 . This can be used to set the value u^1 in the resulting difference scheme.

Problem 3.5 For scheme (3.12), under the condition (3.17), prove that the bounds (3.19) and (3.24) obtained using frequency domain analysis and energetic techniques, respectively, are identical.

Problem 3.6 For the scheme (3.25), find a bound on the solution size in terms of the initial conditions and ω_0 , through the expressions for the energy given in (3.28). The discussion of quadratic forms in §2.4.3 may be of use here.

Problem 3.7 For the scheme (3.29),

- (a) write the characteristic polynomial in z , and prove the stability condition (3.30), and
- (b) show that the expression given in (3.31) is indeed a conserved energy, and find conditions under which it is non-negative.

Problem 3.8 The deviation in cents of a frequency f (in Hertz) from a reference frequency f_0 is defined as

$$\text{deviation in cents} = 1200 \log_2(f/f_0)$$

Find an explicit expression for this deviation in the case of schemes (3.12), and (3.25). Recall that $f = \omega/2\pi$ is the frequency of oscillation of the scheme, and $f_0 = \omega_0/2\pi$ is the “reference” frequency of the underlying SHO. For a sample rate of $f_s = 44100$ Hz, find the lowest reference frequency f_0 at which the absolute value of this deviation is equal to 100 cents (one semitone) and 10 cents (one tenth of a semitone). For scheme (3.29), with a value of $\alpha = 0.82$, what is the minimum value of this deviation over the band $f_0 \in [0, f_s/4]$?

Problem 3.9 For difference scheme (3.40) for the coupled system of oscillators, left multiply by the vector $\delta_t \mathbf{u}^T$, and show that the following numerical energy is a conserved quantity:

$$\mathfrak{h} = \mathfrak{t} + \mathfrak{v} \quad \text{with} \quad \mathfrak{t} = \frac{1}{2}(\delta_t \mathbf{u}^T) \mathbf{M} (\delta_t \mathbf{u}) \quad \mathfrak{v} = \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{e}_{t-\mathbf{u}}$$

You may assume that \mathbf{M} and \mathbf{K} are symmetric.

Generalize the energy analysis of the simple difference scheme for the harmonic oscillator in §3.2.6. Can you show that the energy \mathfrak{h} above is positive definite under the condition (3.41) obtained through frequency domain analysis?

Problem 3.10 Consider the scheme (3.42) for the coupled system (3.9).

(a) Show that it may be written in the form

$$\mathbf{u}^{n+1} = \mathbf{A}^{-1} \mathbf{B} \mathbf{u}^n - \mathbf{u}^{n-1}$$

and find the matrices \mathbf{A} and \mathbf{B} explicitly in terms of \mathbf{M} , \mathbf{K} , the free parameter α and the time step k .

(b) Show that, for symmetric \mathbf{K} and \mathbf{M} , an energy function of the form $\mathfrak{h} = \mathfrak{t} + \mathfrak{v}$ is conserved, with

$$\mathfrak{t} = \frac{1}{2}(\delta_{t-} \mathbf{u}^T) \mathbf{M} (\delta_{t-} \mathbf{u}) \quad \mathfrak{v} = \frac{\alpha}{2} \mathbf{u}^T \mathbf{K} e_{t-} \mathbf{u} + \frac{1-\alpha}{2} \mu_{t-} (\mathbf{u}^T \mathbf{K} \mathbf{u})$$

Problem 3.11 Find a general real-valued solution to (3.43) under the conditions a) $\sigma_0 = \omega_0$ and b) $\sigma_0 > \omega_0$. Determine the values of any constants in terms of initial conditions. Are these situations that you expect to find in a practical acoustics setting? Discuss.

Problem 3.12 Prove that the differential equation given in (3.49) holds in the case of the linear oscillator with loss (3.43).

Problem 3.13 Consider the expression (3.52) for the roots of the characteristic polynomial for the difference scheme (3.50) for the simple harmonic oscillator with loss.

(a) Supposing that the stability condition (3.53) is respected, find the conditions under which the solutions are either both real, or complex conjugates.

(b) Again supposing that the stability condition (3.53) is respected, what form do the roots take when $\sigma_0 > 1/k$? What kind of solution will be produced, and will it be physically reasonable?

Problem 3.14 Considering scheme (3.56) for the simple harmonic oscillator with loss, derive the stability condition (3.58) by using the condition (2.14). Derive the characteristic polynomial equation and a similar condition for the following scheme, which makes use of a forward difference approximation to the loss term:

$$\delta_{tt} u = -\omega_0^2 u - 2\sigma_0 \delta_{t+} u$$

Problem 3.15 Consider the following approximation to the lossy SHO (3.43):

$$\delta_{tt} u = -\omega_0^2 \mu_{t-} u - 2\sigma_0 \delta_{t-} u$$

(a) Find a stability condition on k .

(b) Calculate the numerical decay time for this scheme, under low loss conditions, and show that it will be dependent on ω_0 .

Problem 3.16 Considering the exact two-step recursion (3.61) for the SHO with loss (3.43), it is reasonable to expect that it may be written in standard operator notation.

(a) Consider first the two-parameter scheme (in α and β),

$$\delta_{tt} u = -\omega_0^2 (\alpha + (1-\alpha)\mu_{t-}) u - 2\sigma_0 (\beta\delta_{t-} + (1-\beta)\delta_{t-}) u$$

which is a natural candidate for such a correspondence. Show that it is not possible, in general, to identify it with recursion (3.61). Is it really a two-parameter scheme?

(b) Show that the two-parameter scheme (in γ and ϵ)

$$\delta_{tt} u = -\omega_0^2 (\gamma\mu_{t-} + \epsilon\mu_{t+} + (1-\gamma-\epsilon)) u - 2\sigma_0 \delta_{t-} u$$

can be identified with (3.61), and find the correct settings for γ and ϵ in terms of σ_0 and ω_0 .

Problem 3.17 Consider the finite difference scheme for the oscillator given by

$$u^{n+M} - 2u^n + u^{n-M} = -\omega_0^2 (Mk)^2 u^n$$

where $M \geq 1$ is an integer, and k , as before, is the time step.

(a) Show that this difference scheme approximates the simple harmonic oscillator to second order accuracy in the time step k .

(b) Find the characteristic polynomial for this scheme, and find the roots, and thus the characteristic frequencies.

(This is an illustration of a digital structure which is capable of producing many frequency components, though the operation count is the same as that of a single oscillator. This is the same principle which underlies the digital waveguide—see §1.2.3.)

Problem 3.18 For scheme (3.63), for the SHO with a source term, perform energy analysis analogous to that of the model equation (3.62), and find a bound on the size of the solution u^n in terms of initial conditions, and the values of the time series $[F]$ at time steps between 0 and n .

3.8 Programming Exercises

Exercise 3.1 Adjust the code provided in §A.1 such that only two memory locations are required in order to represent the state of the scheme (3.12) for the simple harmonic oscillator. This is slightly trickier than it appears to be—a hint is that you may wish to break the time series u^n into two sets, containing values for n odd and even.

Exercise 3.2 Create two matlab scripts which generate output according to the difference scheme family (3.29), and the exact scheme (3.38). The parameters which should be set in the preamble to your code should be: f_s , the sample rate, T_f , the total duration of the simulation, in seconds, f_0 , the reference frequency of the SHO, in Hertz, u_0 , the initial displacement of the oscillator, v_0 . For the script corresponding to the family (3.29), the additional parameter α must be specified. In this case, be sure to perform a test for numerical stability involving f_s , f_0 and α , and create an error message if it is violated. Your code should create plots of: the oscillator output as a function of n , the time step, the potential, kinetic, and total energy v^n , k^n , and h^n as a function of n , as well as the variation in energy, i.e., $(h^n - h^0)/h^0$. You may wish to use the Matlab code provided for the simple scheme (3.12) in §A.1 as a reference.

Exercise 3.3 Adjust the code provided in §A.1 such that, in addition to the output of the difference scheme, the exact solution to the differential equation (3.1) is generated, at the sample rate specified. Your code should play both outputs in succession. At a given sample rate such as f_s , find the smallest value of the reference frequency f_0 such that the frequency of the output of the difference scheme is perceptually distinct from that of the exact solution. Try typical values of the sample rate used in audio applications, such as $f_s = 32000$ Hz, 44100 Hz, or 48000 Hz.

Exercise 3.4 Generalize the script provided in §A.1 for the simple harmonic oscillator by introducing loss, according to scheme (3.50). In your code, you should have T_{60} as a global parameter, from which the parameter σ_0 is then derived.

Exercise 3.5 Program the wave digital version of the simple harmonic oscillator, as discussed in §3.3.3. Show that its behaviour is identical to that of the scheme (3.29), under the choice of $\alpha = 1/2$. In order to do this, you will need to ensure that both methods are initialized in corresponding ways.

Exercise 3.6 Consider the two mass system shown at left in Problem 3.2 above, with $M_1 = 1$, $M_2 = 0.0001$, $K_1 = 1000$, $K_2 = 5000$, and $K_3 = 10000$. Write matlab implementations of the schemes (3.40) and (3.42), running at 44100 Hz, and using a parameter value of $\alpha = 0.82$ in the second case. Your code should calculate 1 s worth of output of the displacements u_1 and u_2 , and plot the spectrum magnitude of each such output. Compare the characteristic frequencies of the output, in each case, to the exact frequencies of the system (3.9), calculated through (3.11).

Chapter 4

The Oscillator in Musical Acoustics

Finite difference schemes for the simple harmonic oscillator were introduced in some detail in the last chapter. Though the SHO is, by itself, not an extremely interesting system, many excitation mechanisms in musical instruments may be modeled as variants of it, generally nonlinear. Some examples which will be discussed in this chapter are the hammer/mallet interaction, reeds (including lip models), and the bow.

When designing a sound synthesis simulation, particularly for nonlinear systems, there are different ways of proceeding. It is sometimes useful, in the first instance, to make use of an ad hoc numerical method, generally efficient and easily programmed, in order to generate sound output quickly, perhaps so as to uncover problems in the formulation of the underlying model system. On the other hand, with some extra work (and generally at some additional computational expense), better behaved numerical methods may be derived. Robustness, or numerical stability under a wide variety of possible playing conditions, is an especially useful property for a synthesis algorithm to possess, particularly if the physical structure it simulates is to be virtually connected to other such structures. Other difficulties appear as well, including potential non-uniqueness of computed solutions, which comes up in the case of the bow mechanism.

Frequency domain analysis may indeed be extended to deal with nonlinear systems such as those discussed in this chapter, through perturbation techniques such as Linstedt-Poincaré methods, or harmonic balance techniques [178]. The conclusions one may reach in this manner are generally extremely complex, and are not particularly useful when it comes to the analysis of numerical stability for associated numerical methods. Energy techniques will thus be employed whenever possible. It is worth noting that through the use of such techniques, though generally applicable to all the nonlinear systems discussed in this chapter, simple results are obtained only when nonlinearities of a smooth type, i.e., those which may be expressed in terms of polynomial terms. See §4.2.1. Though in the case of excitation mechanisms, the nonlinearities here are not of this form, some time will be spent on energy methods, mainly because which the nonlinearities which arise in distributed problems, such as strings and plates, generally are quite smooth (see Chapters 8 and 13).

A simple nonlinear generalization of the simple harmonic oscillator appears in §4.1, followed by various applications of interest in musical acoustics, such as lossless collision mechanisms (hammers), in §4.2, and continuous excitations involving loss, such as the bow and reed mechanisms, in §4.3. All of these systems are presented in a pure lumped form; connections to distributed models of instrument bodies are dealt with in subsequent chapters.

References for this chapter include: [142, 52, 246, 284, 140, 213, 214, 63, 9, 198, 229]

4.1 Nonlinear Oscillators

Nonlinear oscillators of lumped type appear in many instances throughout musical acoustics. Normally, such an oscillator is coupled to a distributed system, but, for the sake of analysis, it is useful to look at the behaviour of such systems when decoupled. The word “oscillator” employed here brings to mind an externally-driven source, but, in this chapter, no distinction is made among driven oscillators, passive nonlinear systems, and nonlinear systems in musical acoustics which do not really oscillate at all (as in, say, the case of the hammer). All can be described in roughly the same way.

A general uncoupled nonlinear oscillator may be written as

$$\frac{d^2u}{dt^2} = -F\left(u, \frac{du}{dt}\right) \quad (4.1)$$

Here, if $u(t)$ represents a position, then, F has the interpretation, at least in a mechanical system, of a force, divided by the mass of the oscillating object—see the comments on page ???. The simple harmonic oscillator (3.1) is a special case of such a form, with $F = \omega_0^2 u$. As in the linear case, this is a second-order ODE, and thus requires the specification of two initial conditions, generally $u(0)$ and $\frac{du}{dt}|_{t=0}$.

4.2 Lossless Oscillators

A special case of great interest in musical acoustics is that for which F depends only on u ,

$$\frac{d^2u}{dt^2} = -F(u) \quad (4.2)$$

which generally describes lossless systems. It is not difficult to see why—multiplying by du/dt gives

$$\frac{du}{dt} \frac{d^2u}{dt^2} + \frac{du}{dt} F(u) = 0 \quad \Rightarrow \quad \frac{d\mathfrak{H}}{dt} = 0$$

with

$$\mathfrak{H} = \mathfrak{T} + \mathfrak{V} \quad \mathfrak{T} = \frac{1}{2} \left(\frac{du}{dt} \right)^2 \quad \mathfrak{V} = \int_0^u F(\eta) d\eta \quad (4.3)$$

Thus, just as in the case of the linear oscillator, the ODE (4.1) possesses a conserved energy. Notice that the limit in the integral used to define \mathfrak{V} could be changed, shifting the total energy by a constant.

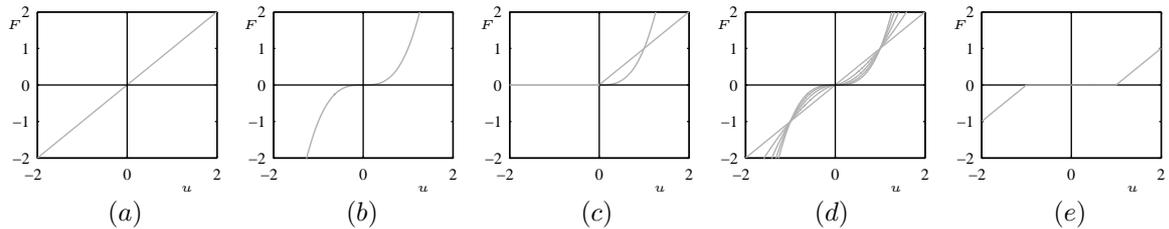


Figure 4.1: *Some lossless characteristics $F(u)$. (a) Linear, (b) cubic, (c) semi-linear and semi-cubic, (d) power law, under various choices of the exponent α , and (e) linear center-limited.*

The most obvious discretization of (4.2) is

$$\delta_{tt}u = -F(u) \quad (4.4)$$

This scheme is explicit, and extremely simple to implement. Unfortunately, however, its stability properties are somewhat obscure—see the following section on the special case of the cubic nonlinear oscillator for more comments on this topic. Still, it is a good first stab if one wants to put together a simulation in a hurry, though one should be alert to the potential for instability.

Given the discussion in the last chapter on energy conservation in the case of the linear oscillator, one might wonder whether a conservative scheme for (4.2) exists. One way of proceeding is to write $F = d\mathfrak{V}/du = (d\mathfrak{V}/dt)/(du/dt)$, and thus obtain the scheme

$$\delta_{tt}u = -\frac{\delta_t \mathfrak{v}}{\delta_t u} \quad (4.5)$$

where \mathfrak{v}^n is a discrete approximation to the potential energy $\mathfrak{V}(t)$ at time $t = nk$. This scheme is indeed conservative (see Problem 4.1), but, unfortunately, uniqueness and existence results are not forthcoming. There is much more to say about conservative schemes for this oscillator—see [112] for an introduction, and commentary. Numerical methods for Hamiltonian systems have undergone a large amount of work in recent years—of particular interest are symplectic methods [222]. A full discussion here would be rather lengthy, and it is perhaps better to examine several particular cases of relevance to musical acoustics and sound synthesis.

4.2.1 The Cubic Nonlinear Oscillator

There is one special case of the nonlinear oscillator which is simply expressed, and of extreme utility in musical acoustics, namely that in which the nonlinearity may be expressed as cubic, i.e.,

$$\frac{d^2u}{dt^2} = -\omega_1^4 u^3 \quad (4.6)$$

This equation goes by various names—Duffing’s equation (without the linear, damping and forcing terms), and also the stiffening or hard spring [166]. Though not entirely realistic, it serves as an excellent model problem—many distributed models of nonlinear string and plate vibration, of crucial importance in musical acoustics applications, can be expressed as third-order (i.e., cubic) systems. The third-order oscillator possesses many nice properties, and, by the standards of nonlinear systems, is comparatively easy to analyze—using perturbation analysis, one can say a great deal about interesting phenomena such as bifurcations, etc. [178]. As this is relatively well-documented, it is perhaps better to spend looking at the properties of numerical simulation methods. As will be seen, the nice properties of this system also carry over to the discrete case.

The energy balance of this system may be found directly. It is:

$$\frac{d\mathfrak{H}}{dt} = 0 \quad \text{with} \quad \mathfrak{H} = \mathfrak{T} + \mathfrak{V} \quad \mathfrak{T} = \frac{1}{2} \left(\frac{du}{dt} \right)^2 \quad \mathfrak{V} = \frac{1}{4} (\omega_1 u)^4 \quad (4.7)$$

Just as in the case of the linear oscillator, it is clear that, due to the non-negativity of \mathfrak{H} , \mathfrak{T} and \mathfrak{V} , one can arrive at the following bounds on the solution size:

$$\left| \frac{du}{dt} \right| \leq \sqrt{2\mathfrak{H}(0)} \quad |u| \leq \frac{\sqrt{2}(\mathfrak{H}(0))^{1/4}}{\omega_1}$$

Again, the solution size may be bounded in terms of the initial conditions, as well as the parameter ω_1 .

The Mixed Linear/Cubic Oscillator and Qualitative Behaviour

Useful, for conceptual purposes, as an illustration of some of features which appear in much more complex vibrating systems, is the combined linear-cubic oscillator, of the form

$$\frac{d^2u}{dt^2} = -\omega_1^2 u - \omega_1^4 u^3 \quad (4.8)$$

This system, like (4.6), also possesses a well-behaved energy—the potential energy is the sum of that of the cubic oscillator, and of the SHO itself. See Figure 4.2, which shows the solution to system (4.8), under initial conditions of increasing magnitude. For small values of the initial condition,

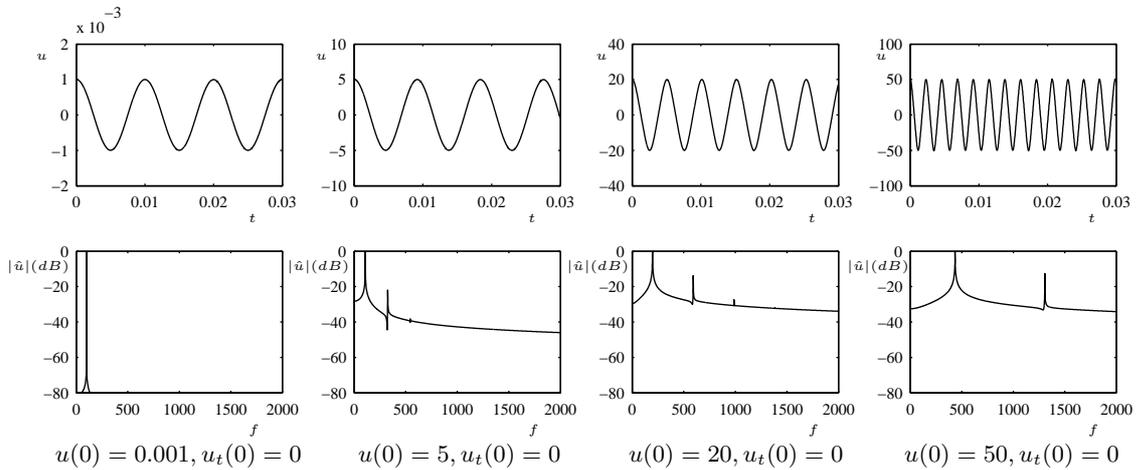


Figure 4.2: *Output waveform (top), and spectrum (bottom), for the mixed linear-cubic nonlinear oscillator (4.8), with $\omega_0 = 100\pi$, and $\omega_1 = \sqrt{20\pi}$, under different initial conditions (given in the figure).*

the solution approaches the solution to the SHO, giving a sinusoid at angular frequency ω_0 . As the initial conditions are increased, two important phenomena are observed. First, under moderate-sized excitation, the solution remains approximately sinusoidal—but at a frequency which is different from that of the solution to the SHO, and, in this case, generally higher. Second, under even stronger excitation, one begins to see other components appearing in the solution, with the most important contribution at (in this case) three times that of the “fundamental.” Both these effects play an important perceptual role in musical acoustics, particularly in the case of string and percussion instruments, leading to phenomena such as pitch glides in strings plucked at high amplitudes, and to the spontaneous generation of high-frequency energy in instruments such as cymbals and gongs.

Finite Difference Schemes

There is a great variety of schemes available in the nonlinear case, even for a basic system such as (4.8). Consider the following difference approximations to (4.6) above:

$$\delta_{tt}u = -\omega_1^4 u^3 \quad \Rightarrow \quad u^{n+1} = (2 - \omega_1^4 k^2 (u^n)^2) u^n - u^{n-1} \quad (4.9a)$$

$$\delta_{tt}u = -\omega_1^4 u^2 \mu_{t \cdot} u \quad \Rightarrow \quad u^{n+1} = \frac{2}{1 + \omega_1^4 k^2 (u^n)^2 / 2} u^n - u^{n-1} \quad (4.9b)$$

$$\delta_{tt}u = -\omega_1^4 \mu_{t \cdot} (u^2) \mu_{t \cdot} u \quad \Rightarrow \quad \text{No Explicit Update Form} \quad (4.9c)$$

All three are centered, and thus second-order accurate schemes. The approximation (4.9a) is perhaps the most natural, and certainly the simplest in implementation. Unfortunately, it is difficult to say much about its stability behaviour, as it does not possess a conserved energy analogous to (4.7). The schemes (4.9b) and (4.9c) are conservative. Through multiplication of these schemes by $\delta_t u$, one may arrive at the energy balances

$$\delta_{t+} \mathfrak{h} = 0 \quad \text{with} \quad \mathfrak{h} = \mathfrak{t} + \mathfrak{v} \quad , \quad \mathfrak{t} = \frac{1}{2} (\delta_{t-} u)^2 \quad \mathfrak{v} = \begin{cases} \frac{\omega_1^4}{4} u^2 e_{t-} u^2 & \text{for scheme (4.9b)} \\ \frac{\omega_1^4}{4} \mu_{t-} u^4 & \text{for scheme (4.9c)} \end{cases} \quad (4.10)$$

In both cases, the kinetic and potential energy terms are non-negative, and one can proceed to find a bound on solution size. For instance, one may immediately write, for either scheme,

$$|\delta_{t-} u| \leq \sqrt{2\mathfrak{h}} \quad (4.11)$$

bounding the rate of growth of the solution. With a bit more work, bounds on the solution size itself may be obtained. See Problem 4.2. In fact, such schemes pose even less difficulty, in terms of stability analysis, than schemes for the linear oscillator—both schemes are unconditionally stable, for any choice of the time step k ! Though one can use schemes such as (4.9a) (and indeed, when the nonlinearity is of a different form, schemes like this are really the best option), there is the danger of instability, which shows itself in a much more obscure way than in the linear case. See Figure 4.3, which illustrates the unexpected instability of scheme (4.9a) after hundreds of thousands of time steps of stable behaviour. A provably stable scheme is thus, for synthesis applications, a safe choice.

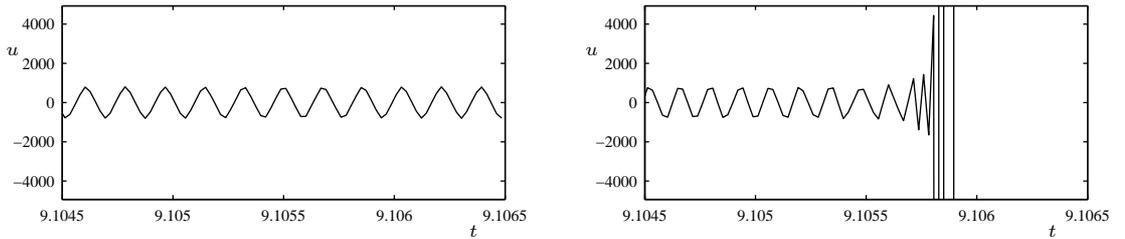


Figure 4.3: Numerical solutions to the cubic nonlinear oscillator, with $\omega_1 = \sqrt{20\pi}$, and with $f_s = 44100$ Hz, for energy-conserving scheme (4.9b), and for scheme (4.9a), which becomes abruptly unstable after more than 9 seconds of run time (at time step $n = 401562$). The initial conditions used, in this case, are $u^0 = u^1 = 704.8165$.

There is, however, a subtle, but extremely significant difference between the two conservative

schemes. Scheme (4.9b) may be written as an explicit recursion, i.e., one may solve directly for u^{n+1} in terms of u^n and u^{n-1} , which are known. This is not the case for scheme (4.9c)—the unknown value u^{n+1} is coupled to the known values through a cubic equation, which must be solved, at each time step. This is not particularly difficult: one may employ a (somewhat formidable) formula for the roots of a cubic, or, better, use an iterative method. The problem is, however, that the cubic possesses, in general, three roots. Thus the issue of uniqueness of the numerical solution rears its head. One can of course go further and look for a condition on k , the time step, such that the cubic only possesses one real solution, but the point is that such difficulties do not arise for scheme (4.9b). The distinction between the two types of stable conservative schemes becomes much more important in the distributed setting, as will be seen in the case of nonlinear string vibration in §8.2. Scheme (4.9a) corresponds to an explicit scheme, scheme (4.9b) to an implicit scheme, where updating may be done through the solution of a linear system, and scheme (4.9c) to an implicit scheme requiring algebraic nonlinear solution techniques.

4.2.2 Power Law Nonlinearities

Many models of nonlinearity in hammers and mallets [40, 52] are based around the use of a power law, of the form

$$F(u) = \omega^{\alpha+1} \text{sign}(u) |u|^\alpha \quad (4.12)$$

where α , the nonlinear exponent is usually determined experimentally (through measurements, say, of a felt compression characteristic), though it is possible to arrive at a theoretical justification, such as Hertz's Contact Law [116]. See Figure 4.1(d). In fact, this nonlinearity most often occurs in musical acoustics in a “one-sided” form—see the next section.

Explicit energy-conserving schemes for such a nonlinearity are not easily arrived at, except in the cases $\alpha = 1$ (linear) and $\alpha = 3$ (cubic, as discussed above). The difficulty is that the nonlinear characteristic F is not an analytic function of u unless α is an odd integer. Thus, the simplest approach is probably to use a straightforward scheme such as (4.4) in such cases. Drawing a lesson from the success of the partially-implicit discretization of the cubic oscillator, one might guess that a scheme such as

$$\delta_{tt}u = -\omega^{\alpha+1} \mu_t u |u|^{\alpha-1} \quad (4.13)$$

might possess better stability characteristics than the simple scheme (4.4)—see Programming Exercise 4.1.

4.2.3 One-sided Nonlinearities and Collisions: Hammers and Mallets

So far, the nonlinearities which have been discussed have been two-sided—the function $F(u)$ is antisymmetric, and thus, when interpreted as a force, acts in the direction opposite to that of u , when interpreted as a displacement. In many settings in musical acoustics involving collisions, a one-sided definition is more useful—the force only acts when the displacement is positive (say), and acts so as to repel the mass. Such characteristics are illustrated in Figure 4.1(c).

The “oscillator” (4.2), in this case, serves as a model of a collision of a mass with a rigid object, such as a wall, and where $F(u)$ describes the stiffness of the mass. It is useful to adopt the following notation:

$$\frac{d^2u}{dt^2} = -[F(u)]^+ \quad \text{where} \quad [F(u)]^+ = \begin{cases} 0, & u \leq 0 \\ F(u), & u > 0 \end{cases} \quad (4.14)$$

As an example, consider a comparison between the case of a collision where, in one case, the stiffening force is linear, i.e., $F(u) = \omega_0^2 u$, and in the other, the stiffening force is cubic, i.e., $F(u) =$

$-\omega_1^4 u^3$. Plots of the resulting behaviour are shown in Figure 4.4. The most important thing to note is that in the case of the linear object, regardless of the velocity with which the object approaches the barrier, the duration of the collision remains constant. For the stiffening object, the contact time becomes shorter as approach velocity is increased. Such effects of duration of contact time obviously have very important perceptual implications in sound synthesis: for a linear object, the resulting dynamics are simply scaled up in amplitude as collision velocity is increased—there will be no variation in timbre. For the nonlinear stiffening object, faster collision velocity leads to a sharper, brighter sound. There is little doubt that most collisions of interest in musical acoustics (such as, e.g., piano hammer excitation) exhibit the latter behaviour. Thus a good nonlinear model is necessary in any percussive instrument, otherwise the resulting sound will be rather uniform, and insensitive to a player’s gesture.

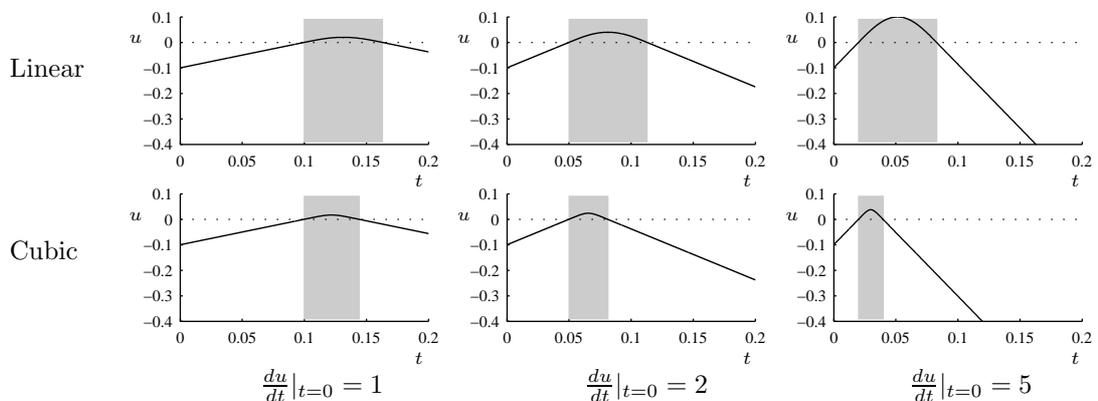


Figure 4.4: Plots of displacement u against time t , for the collision of a mass with a rigid barrier, in operation when u , the center of mass of the object, is positive. In this case, the mass approaches the barrier from below, with initial position $u(0) = -0.1$, and initial velocities as indicated. The top row shows the case of an object with linear stiffness, with $\omega_0 = 50$, and the bottom row that of an object with a cubic stiffening characteristic, with $\omega_1 = \sqrt{5000}$. The contact duration is indicated by a shaded grey area.

Partial Conservation

As mentioned above, in §4.2.2, when the characteristic $f(u)$ is not analytic, it becomes much more difficult to design a conservative numerical method (i.e., one for which numerical stability is easy to guarantee). The same is true of one-sided nonlinearities. One might expect, however, that if the one-sided nonlinearity is of linear or cubic type, conservative behaviour can at least be ensured over intervals for which $u > 0$ or $u \leq 0$. Consider the following finite difference approximations to such one-sided nonlinearities:

$$\delta_{tt}u = -\omega_0^2[u]^+ \qquad \delta_{tt}u = -\omega_1^4([u]^+)^2 \mu_{t,u} \qquad (4.15)$$

where the $[\cdot]^+$ notation is as in (4.14). Such algorithms are not strictly conservative, and thus there is no global expression for energy. Consider, however, the energy-like functions

$$\bar{h} = \frac{1}{2}(\delta_t u)^2 + \frac{\omega_0^2}{2}[u]^+[e_t u]^+ \quad \bar{h} = \frac{1}{2}(\delta_t u)^2 + \frac{\omega_1^4}{4}([u]^+)^2([e_t u]^+)^2$$

Such expressions are at least partially conserved by schemes (4.15) for intervals over which u does not change signs—on the other hand, an energy jump will be observed, as illustrated in Figure 4.5. The jump is larger in the case of the one-sided linear collision than for the one-sided cubic collision, for which the total jump is on the order of machine precision. One effect of such an energy jump is that an object undergoing a theoretically lossless collision will exhibit a difference in exit speed from entry speed. In general, however, such an effect is quite small over normal ranges of parameters. One could go further here, and look at possible ways around this problem (some of which are very involved), but for sound synthesis purposes, schemes such as the above are quite sufficient.

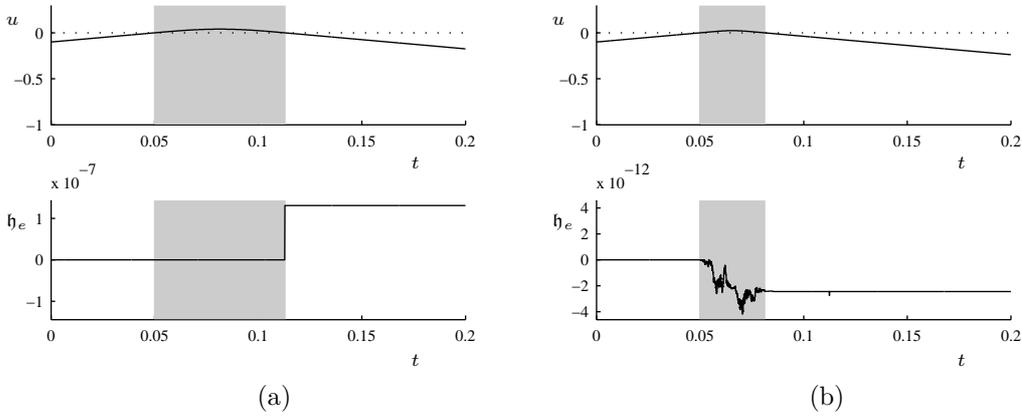


Figure 4.5: Plots of displacement u against time t , for the collision of a mass with a rigid barrier (top) and the variation in numerical energy $h_e = (h - h^0) / h^0$ (bottom), in the case of (a) a linear collision characteristic, and (b) a cubic collision characteristic. In each case, the mass approaches the barrier from below, with initial position $u(0) = -0.1$, with an initial velocity of 5. The stiffness parameters are chosen as $\omega_0 = 50$, for the linear collision, and $\omega_1 = \sqrt{5000}$ for the cubic collision. The algorithms in (4.15) are employed, with a sample rate of $f_s = 44100$ Hz.

Hammer and Lumped Oscillator

With a very slight added degree of complexity, the collision model above becomes very close to that which is often used in models of hammer or mallet interaction with a distributed object. Consider the case of a one-sided nonlinearity acting between a hammer-like object, with an inherent nonlinear stiffening characteristic and a linear lumped mass-spring system, as shown at left in Figure 4.6. Here, it is useful, in a first step, to write the equations of motion of this coupled system as

$$M \frac{d^2 u}{dt^2} = -Ku + f(u - u_H) \quad M_H \frac{d^2 u_H}{dt^2} = -f(u - u_H) \quad f(u - u_H) = K_H ([u_H - u]^+)^{\alpha}$$

where here, u is the displacement of the “target,” a mass-spring system of mass M and stiffness parameter K , and u_H is the displacement of the hammer, of mass M_H , and with a stiffness parameter

K_H . Notice that the interaction force f only acts when the hammer position is greater than that of the target. The nonlinearity has been chosen to be of the form of a power law, a common choice in the musical acoustics of hammers [52].

For the sake of the musician-programmer, it is always useful to reduce the number of parameters which define a system to the bare minimum. In the above case, it is clear that not all of M , K , M_H and K_H need be independently specified, and one may write

$$\frac{d^2u}{dt} = -\omega_0^2u + \mathcal{M}F(u - u_H) \quad \frac{d^2u_H}{dt} = -F(u - u_H) \quad F(u - u_H) = \omega_H^{\alpha+1} ([u_H - u]^+)^{\alpha} \quad (4.16)$$

where $\omega_0 = \sqrt{K/M}$, $\omega_H = (K_H/M_H)^{\frac{1}{\alpha+1}}$, the hammer mass/target mass ratio $\mathcal{M} = M_H/M$. and $F = f/M_H$. System (4.16) may be easily shown to be exactly lossless—see Problem 4.3.

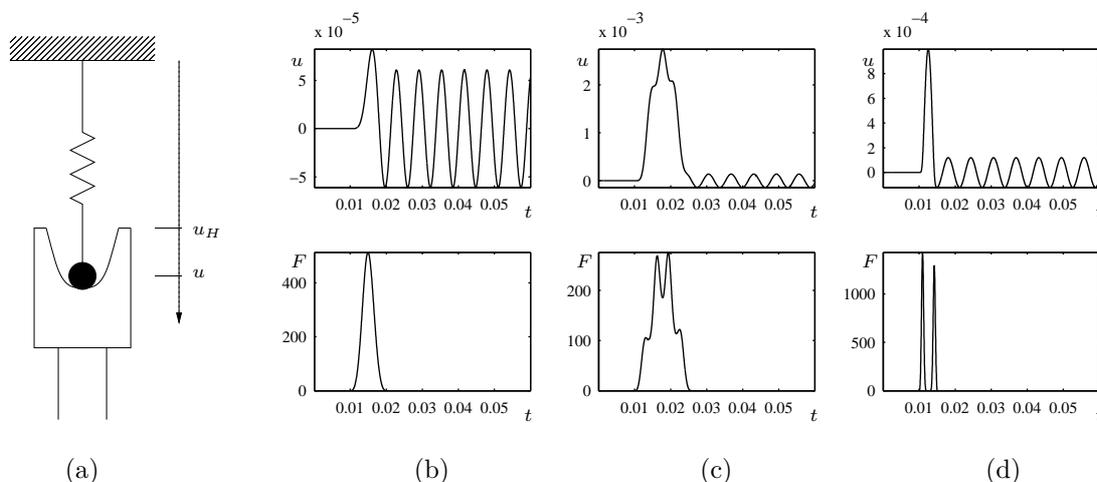


Figure 4.6: Collision between a hammer and a mass-spring system, as described by (4.16), illustrated in (a). Various resulting displacements (top) and interaction forces (bottom) are shown at right—in each case, the initial hammer velocity is 1, its initial displacement is 0.01 m below the target mass, and the mass spring system has angular frequency $\omega_0 = 1000$, and the nonlinear exponent $\alpha = 3$. In (b) is shown a simple interaction, in the case of $\omega_H = \sqrt{100000}$, and $\mathcal{M} = 10$, in (c) a more complex force interaction showing the reaction of the mass spring system on the hammer, for $\omega_H = \sqrt{100000}$ and $\mathcal{M} = 0.01$, and in (d), a contact/recontact phenomenon, when $\omega_H = \sqrt{2000000}$, and $\mathcal{M} = 1$.

As expected, the behaviour of such a system is enormously more complex than that of the collision of a hammer with a rigid barrier—the force experienced by the mass can undergo oscillations due to the reaction of the mass-spring system back on the hammer, and recontact with the hammer is also possible—see Figure 4.6. Such complex behaviour is also very much characteristic of the interaction between a hammer and a distributed object such as a string. See, e.g., [52], and the comprehensive series of articles by Hall [115]. The subject of hammer/string interaction will be taken up again in §7.5. It is not difficult to extend this system to cover a collision between a hammer and multiple mass-spring systems—see Problem 4.5, and Programming Exercise 4.2—such multiple interactions are rather important in the case of, e.g., the piano, for which a hammer strikes several strings at once. See §7.6.

The simplest finite difference scheme for (4.16) is the following:

$$\delta_{tt}u = -\omega_0^2u + \mathcal{M}F(u - u_H) \quad \delta_{tt}u_H = -F(u - u_H) \quad F(u - u_H) = \omega_H^{\alpha+1} ([u_H - u]^+)^{\alpha} \quad (4.17)$$

Such a scheme, which is entirely explicit, is very easy to implement—see the code example in §A.2. The scheme is not conservative, but when $\alpha = 1$ or $\alpha = 3$, there is a modified form which is at least partially conservative—see Problem 4.4.

4.2.4 Center-limited Nonlinearities

A musically interesting extension of the one-sided nonlinearity is the oscillator (4.2) under a center-limited lossless nonlinear characteristic—see Figure 4.1(e). In this case, the nonlinearity is only active when the magnitude of the displacement u is greater than some threshold—as such it serves as a simple model of an element which is able to rattle. The coupling of an element such as this to a string, as a form of preparation, is discussed in §7.7.3.

As an example of such a nonlinear characteristic, consider the following:

$$F(u) = \begin{cases} \omega^{\alpha+1} (u - \epsilon/2)^\alpha, & u \geq \epsilon/2 \\ 0, & |u| < \epsilon/2 \\ -\omega^{\alpha+1} (-u - \epsilon/2)^\alpha, & u \leq -\epsilon/2 \end{cases} \quad (4.18)$$

The object to which this characteristic corresponds can be thought of as a “dumbbell,” of length ϵ (which is an extra design parameter), with a stiffness described by a power-law nonlinearity. See Figure 4.7(a). The system is conservative, with an energy given by (4.3), and one may develop finite difference schemes with a partial conservation property when $\alpha = 1$ or $\alpha = 3$, in a manner very similar to the one-sided collision model of the previous section. See Figure 4.7 for some typical simulation results, generated using a straightforward scheme of the type (4.4), and also Programming Exercise 4.3. Loss and the effect of gravity play a rather important role when one is dealing with rattling elements—see Programming Exercise 4.4 for some exploration of these features.

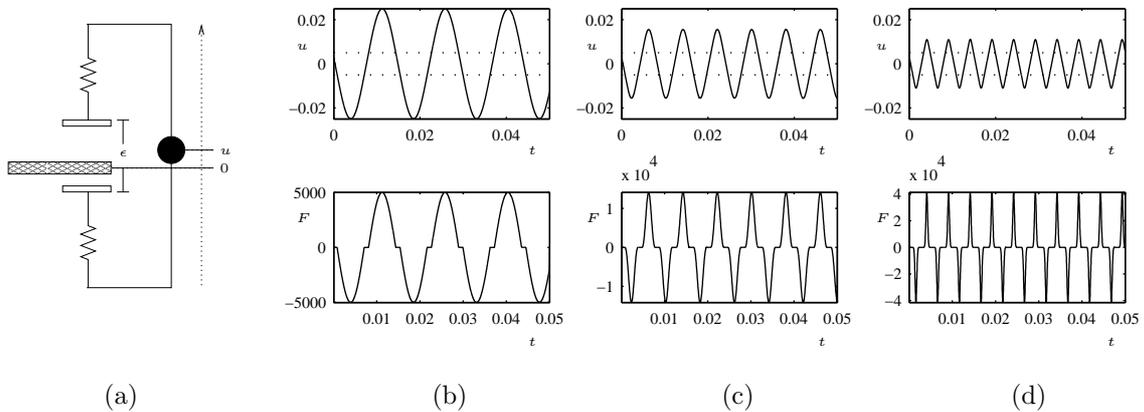


Figure 4.7: (a) Diagram representing a center-limited oscillating object, of length ϵ , and simulations of such an object, of length $\epsilon = 0.01$, and with nonlinear stiffness parameter $\omega = 500$, under different choices of the nonlinear exponent α : in (b), $\alpha = 1$, in (c), $\alpha = 2$ and in (d), $\alpha = 4$. The object displacement u is shown at top, and the force F at bottom. In all cases, the initial position of the object is $u(0) = 0.003$, and the initial velocity is $du/dt|_{t=0} = -10$. Notice that the gross period of oscillation of the element depends strongly on the nonlinear exponent.

4.3 Lossy Oscillators

One could easily introduce a linear loss term into the equation for the lossless oscillator (4.2), but, for collisions of short duration, this will yield results of minor perceptual significance. The real interest in lossy oscillator models in musical acoustics relates to the case of continuous forced excitation, as occurs in bowed string instruments, as well as in woodwind and brass instruments.

4.3.1 The Bow

Before examining more realistic bow models, it is useful to look at an archetypical test problem of the form

$$\frac{d^2 u}{dt^2} = -\alpha \phi \left(\frac{du}{dt} \right) \quad (4.19)$$

Here, $\alpha \geq 0$ is a free parameter, and the function ϕ is a given nonlinear characteristic. Normally, in models of bow friction, the function $\phi(\eta)$ is antisymmetric about $\eta = 0$, and possesses a region of steep positive slope near the origin (sticking regime), outside of which it is of negative slope (sliding regime). This will be discussed further in some examples which follow and in §7.4, but for reference, some typical examples of such bow characteristics appear in Figure 4.8. For most models, the sticking portion of the curve is in fact of infinite slope—two examples are shown in (a) and (b). A continuous curve, as shown in (c), though less physically reasonable, is somewhat easier to work with numerically.

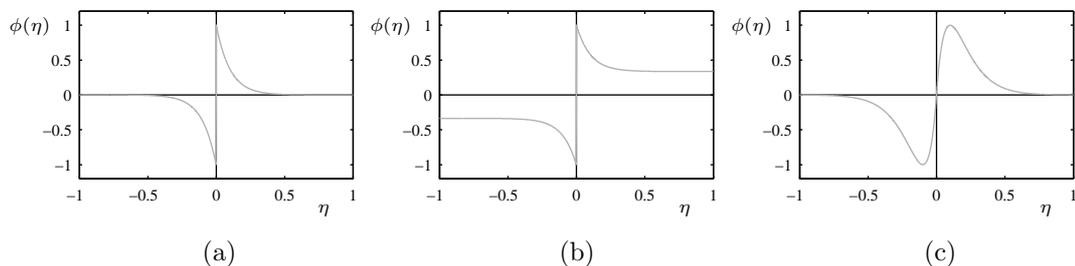


Figure 4.8: *Some bow friction characteristics $\phi(\eta)$. (a) A hard characteristic defined by $\phi(\eta) = \text{sign}(\eta)e^{-\sigma|\eta|}$, for some $\sigma \geq 0$, (b) a hard characteristic with a non-zero limiting sliding friction value, defined by $\phi(\eta) = \text{sign}(\eta) (\epsilon + (1 - \epsilon)e^{-\sigma|\eta|})$, and (c) a soft characteristic of the form $\phi(\eta) = \sqrt{2\sigma}\eta e^{-\sigma\eta^2+1/2}$.*

This equation is of the form of an oscillator, without a stiffness term. The variable u itself, undifferentiated, does not appear in the equation, implying a single mechanism for storing energy (through a kinetic term), and that furthermore, it will not be possible to obtain bounds on u itself. Indeed, one could write the above equation in terms of du/dt alone, though it is simpler to retain the second-order form, especially when the above model of a nonlinear excitation is to be coupled to a distributed model. First order forms equivalent to the above do occur in models of nonlinearities in nonlinear electric circuit components used in analogue synthesizers such as the Moog [117]—see Problem 4.6 for an example. As yet, this remains an unforced problem—the important forcing term will be introduced shortly.

Energy analysis indicates a constraint on ϕ : multiplying by du/dt , one obtains, immediately,

$$\frac{d\mathfrak{H}}{dt} = -\alpha \frac{du}{dt} \phi \left(\frac{du}{dt} \right) \quad \text{with} \quad \mathfrak{H} = \frac{1}{2} \left(\frac{du}{dt} \right)^2$$

Clearly then, if the quantity \mathfrak{H} is to be monotonically decreasing, one must require that the characteristic $\phi(\eta)$ satisfy

$$\phi(\eta)\eta \geq 0 \quad \text{or} \quad \text{sign}(\phi(\eta)) = \text{sign}(\eta) \quad (4.20)$$

which is a requirement for passivity. If such a condition does not hold, it becomes possible for the nonlinearity to behave at certain instants as a source of energy, which is clearly not the case in any musical instrument.

As in the case of the lossless oscillator, various difference schemes are possible. Here are two:

$$\delta_{tt}u = -\alpha\phi(\delta_{t-}u) \quad (4.21a)$$

$$\delta_{tt}u = -\alpha\phi(\delta_t.u) \quad (4.21b)$$

Scheme (4.21a), the simpler of the two, is clearly explicit, due to the use of a backward difference inside the nonlinear characteristic. As one might guess, however, it is rather difficult to say anything conclusive about its behaviour, especially in terms of stability. The scheme (4.21b) permits some such analysis. Multiplying by $\delta_t.u$ yields

$$\delta_{t+}\mathfrak{h} = -\alpha(\delta_t.u)\phi(\delta_t.u) \leq 0 \quad \text{with} \quad \mathfrak{h} = \frac{1}{2}(\delta_{t-}u)^2 \quad (4.22)$$

Thus, as in the continuous case, \mathfrak{h} is non-negative, and monotonically decreasing, and the behaviour of scheme (4.21b) is thus stable, as long as the characteristic satisfies (4.20).

On the other hand, scheme (4.21a) is implicit. It is useful to rewrite it in the following form,

$$-\frac{2}{\alpha k}\delta_t.u + \frac{2}{\alpha k}\delta_{t-}u = \phi(\delta_t.u)$$

for which solutions may be examined graphically in terms of the unknown $\delta_t.u$ —see Figure 4.9.

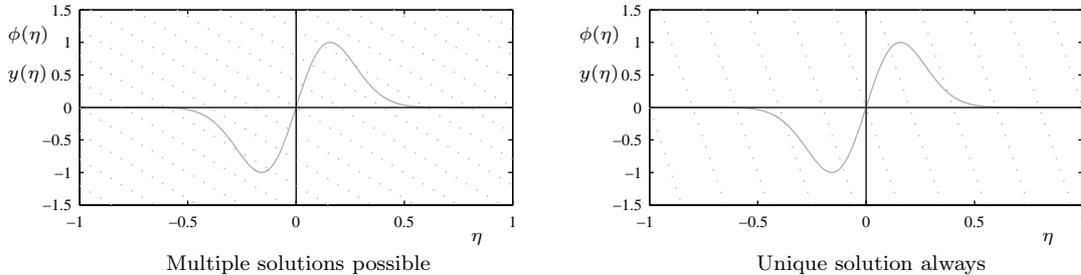


Figure 4.9: Graphical representation of solutions to the equation $\phi(q) = -mq + b$, for a given bow-like characteristic $\phi(q)$ (shown as a solid gray line). The lines $y(q) = -mq + b$, for $m > 0$, and for various values of b , are plotted as dotted grey lines. Left, a choice of m for which multiple solutions may exist, for certain values of b , and right, a choice of m for which the solution is always unique, for any choice of b .

Clearly, there will always be at least one solution to this nonlinear equation, regardless of the form of ϕ . Though existence follows immediately, uniqueness does not—depending on the form of ϕ , there may be multiple solutions to the equation. The issue of multiple solutions in the case of the bow coupled to the string is an interesting one, and the continuous-time case has been discussed

extensively by various authors, beginning with Friedlander [101], and in particular McIntyre and Woodhouse [169]. Here, in the discrete case, the following condition is sufficient:

$$k \leq -\frac{2}{\alpha \min_{\eta} \phi'(\eta)} \quad \text{when} \quad \min_{\eta} \phi'(\eta) \leq 0 \quad (4.23)$$

This condition is a purely numerical one, and applies only to scheme (4.21b), though similar conditions exist for other schemes—see Problem 4.7. It is worth keeping in mind that the above condition *has no bearing on numerical stability*, which is already ensured for scheme (4.21b) under condition (4.20). That is, even when multiple solutions do exist, any choice will lead to strictly passive numerical behaviour (though such a solution may be physically meaningless). It is, however, of the same form as stability conditions which typically arise in the design of explicit schemes for oscillators. See, e.g., §3.2.4, and the following example.

Connection to a Mass-Spring System and Auto-oscillatory Behaviour

To get a better idea of how the bow actually functions, it is better to move to a more concrete setting involving a coupling of a bow model with a single mass-spring system, as shown in Figure 4.10(a). (In fact, it is only a small further step to connect the bow to a fully distributed string model—see §7.4.) Here, the motion of the mass is described by

$$\frac{d^2u}{dt^2} = -\omega_0^2 u - F_B \phi(v_{rel}) \quad \text{where} \quad v_{rel} = \frac{du}{dt} - v_B \quad (4.24)$$

Here, $F_B \geq 0$ is a given bow force (again, it is the bow force divided by the object mass, and has dimensions of acceleration), and v_B is a bow velocity. Notice that it is only the relative velocity v_{rel} of the bow to the mass which appears in the model.

Energy analysis now yields

$$\frac{d\mathfrak{H}}{dt} = -F_B \frac{du}{dt} \phi(v_{rel}) = \underbrace{-F_B v_{rel} \phi(v_{rel})}_{\text{power dissipated by bow}} \quad \underbrace{-F_B v_B \phi(v_{rel})}_{\text{power supplied by bow}}$$

where \mathfrak{H} is the Hamiltonian for a linear oscillator, as given in (3.7), and where the terms on the right hand side of the energy balance may be interpreted as power dissipated and supplied by the bow, as indicated. Because, by (4.20), the dissipated power is negative, one has immediately

$$\frac{d\mathfrak{H}}{dt} \leq -F_B v_B \phi(v_{rel}) \leq |F_B v_B \phi(v_{rel})|$$

For many choices of bow characteristic (such as those pictured in Figure 4.8), F_B represents a maximum bow force—in other words, the characteristic ϕ is bounded such that

$$|\phi| \leq 1 \quad (4.25)$$

In this case, the energy inequality above may be weakened to

$$\frac{d\mathfrak{H}}{dt} \leq |F_B v_B| \quad \Rightarrow \quad \mathfrak{H}(t) \leq \mathfrak{H}(0) + \int_0^t |F_B v_B| dt'$$

This is a useful bound on the growth of the solution, purely in terms of the given numbers F_B and v_B .

The above analysis remains unchanged if the values F_B and v_B are generalized to functions $F_B(t) \geq 0$ and $v_B(t)$, representing the gestural control signals of bow force/object mass and velocity.

The obvious generalization of scheme (4.21b) to the case of coupling to a mass-spring system is

$$\delta_{tt} u = -\omega_0^2 u - F_B \phi(v_{rel}) \quad \text{where} \quad v_{rel} = \delta_t u - v_B \quad (4.26)$$

The energy analysis of this scheme mirrors that of the continuous time system above. Using familiar

techniques, one arrives at the discrete energy balance

$$\delta_{t+} \mathfrak{h} = -F_B v_{rel} \phi(v_{rel}) - F_B v_B \phi(v_{rel}) \leq |F_B v_B|$$

where \mathfrak{h} is the energy of the SHO, as discussed in §3.2.6. Again, energy growth is bounded in terms of the input values (or sequences) F_B and v_B . Two conditions on k , the time step appear:

$$k < \frac{2}{\omega_0} \quad , \quad k \leq \frac{2}{-\max(F_B) \min(\phi')} \quad \text{when} \quad \min(\phi') \leq 0$$

The first condition is the familiar stability condition (3.17) for scheme (3.12) for the SHO, which ensures non-negativity of the numerical energy \mathfrak{h} . The second condition is that required for uniqueness of numerically computed solutions. One could go further and employ a more accurate difference strategy to the linear part of the system, along the lines of the scheme presented in §3.3.4, but the principle of stability analysis remains the same. An implementation of scheme (4.26) appears in §A.3, and makes use of a Newton-Raphson iterative root finder (see Problem 4.8 and Programming Exercise 4.6). For simplicity, the continuous friction characteristic shown in Figure 4.8(c) is used—when a discontinuous characteristic is used, such root finders may also be used, though programming complexity increases somewhat.

The behaviour of the bowed mass system is a simple example of an auto-oscillatory system—given a slowly varying input signal (such as that derived from a player’s gesture), the system can reach of state of continuous oscillation. Even in this rudimentary case, there is a great variety of complex behaviour which appears, and many of the key features of bowed string dynamics may be observed. The most crucial phenomenon is the stick-slip motion of the mass, as illustrated in Figure 4.10(b). The mass “sticks” to the bow for an interval, as the spring is being extended or compressed, exhibiting little displacement relative to the bow, until the force of the spring is sufficient to set the mass into motion, in a direction opposing that of the bow motion, after which it then becomes stuck to the bow once again, etc. The resulting displacement waveform is roughly of the form of an asymmetric triangle.

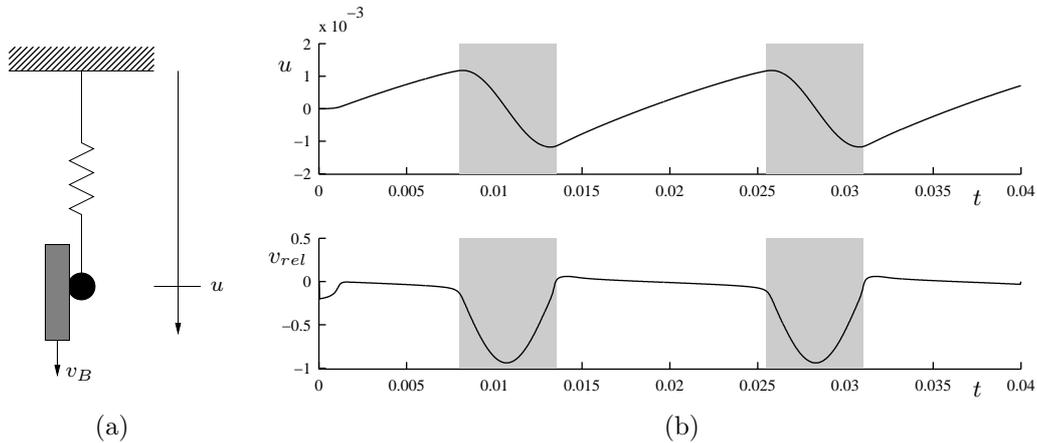


Figure 4.10: (a) A bowed mass, as described by (4.24). (b) Simulation results for the bow mass system, with $f_0 = \omega_0/2\pi = 100$, $F_B = 4000$, and $v_B = 0.2$. The friction characteristic is of the form given in Figure 4.8(c), with $\sigma = 100$. Displacement of mass (top), and relative velocity $v_{rel} = \frac{du}{dt} - v_B$, with shaded regions illustrating intervals during which the mass slips from the bow—otherwise it “sticks.”

Some other phenomena are illustrated in Figure 4.11. In particular, as shown at left, increase in bow force can lead to a characteristic “sharpening” of the displacement waveform, and also an increased period of oscillation—this is a minor effect in bowed string dynamics, and is often referred to as pitch flattening [?]. At right, one may also observe a variation in the length of time necessary to reach steady oscillatory behaviour, also generally decreasing with increased bow force.

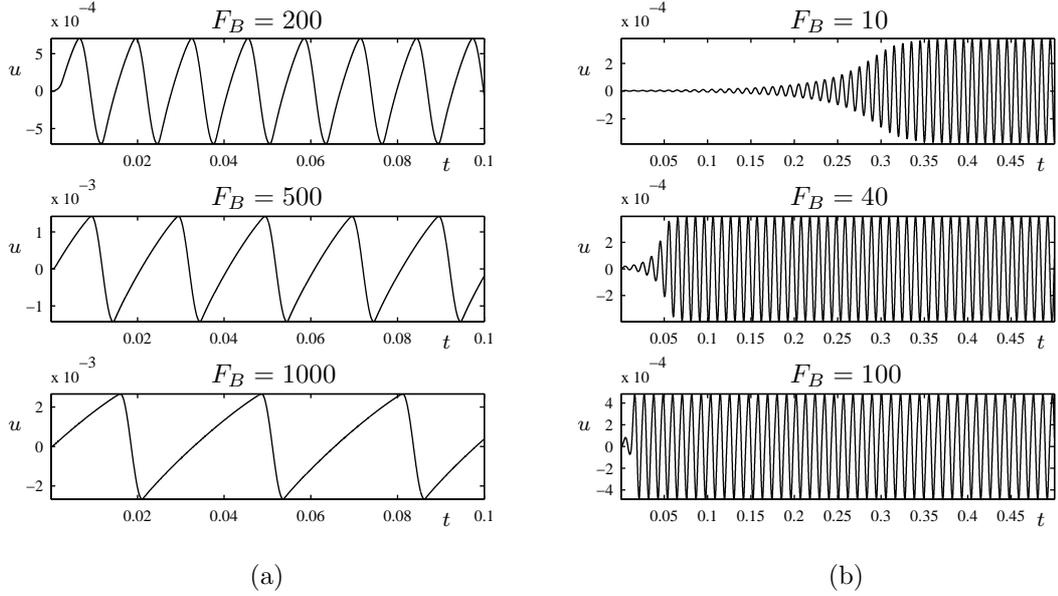


Figure 4.11: Displacement waveforms for the bowed mass system (4.24), under different choices of F_B , as indicated, and assumed constant. In all cases, $v_B = 0.2$, and $f_0 = \omega_0/2\pi = 100\text{Hz}$, and the bow friction characteristic is of the form shown in Figure 4.8(c), with $\sigma = 100$. (a), Pitch flattening and waveform sharpening effects with increased bow force, and (b) shortening of the time of onset of a stable auto-oscillatory regime with increased bow force.

4.3.2 Reed and Lip Models

4.4 Problems

Problem 4.1 Show that the scheme (4.5) for the lossless nonlinear oscillator is indeed conservative, with a conserved energy given by

$$\mathfrak{h} = \frac{1}{2} (\delta_{t-u})^2 + \mu_{t-v}$$

Problem 4.2 Consider scheme (4.9b) for the cubic nonlinear oscillator. From the expression for the potential energy in (4.10), one may deduce that

$$|ue_{t-u}| \leq \frac{2\sqrt{\mathfrak{h}}}{\omega_1^2}$$

A bound on $|u|$ itself does not immediately follow. Use the bound (4.11), along with the above, in order to find a bound on $|u|$. Hint: Square the bound in (4.11), and rewrite δ_{t-u} in terms of u and e_{t-u} .

Problem 4.3 For the hammer/mass-spring collision defined in (4.16), prove that the following quantity,

$$\mathfrak{H} = \frac{1}{2} \left(\frac{du}{dt} \right)^2 + \frac{\omega_0^2}{2} u^2 + \frac{\mathcal{M}}{2} \left(\frac{du_H}{dt} \right)^2 + \frac{\mathcal{M}}{\alpha + 1} (\omega_H [u_H - u]^+)^{\alpha+1}$$

is conserved.

Problem 4.4 For the collision between a hammer and a mass-spring system described by (4.16), consider the following finite difference scheme:

$$\delta_{tt}u = -\omega_0^2 u + \mathcal{M}F(u, u_H) \quad \delta_{tt}u_H = -F(u, u_H) \quad F(u, u_H) = \omega_H^{\alpha+1} \mu_t (u_H - u) ([u_H - u]^+)^{\alpha-1}$$

This is a variation on scheme (4.17), where the nonlinear discretization has a partially implicit character.

a) Show that this scheme exhibits partial energy conservation when $\alpha = 1$ or $\alpha = 3$ (i.e., show that the scheme possesses an energy function which is conserved except at instants when the hammer either comes into contact with, or loses contact with the target mass spring system).

b) Show that the scheme allows for a unique update of u_H and u at each time step.

Problem 4.5 Suppose that, instead of coming into contact with one mass-spring system, a hammer strikes two such systems simultaneously, as illustrated in Figure 4.12. For this system, u_1 and

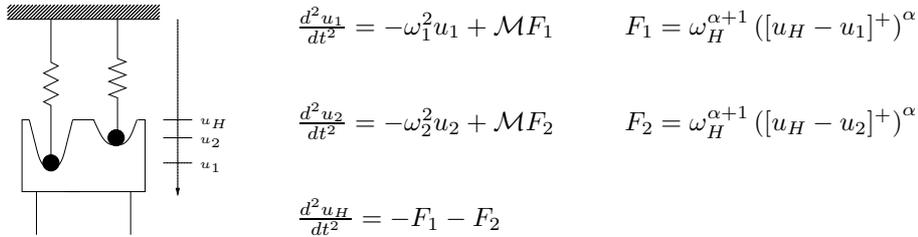


Figure 4.12: Collision between a hammer and two mass-spring systems.

u_2 are the displacements of the two masses, u_H is the hammer displacement, F_1 and F_2 are the forces/mass acting on the two target masses, ω_1 and ω_2 are the angular frequencies of the two mass-spring systems, ω_H the stiffness parameter for the hammer (of nonlinearity exponent α), and \mathcal{M} is the mass ratio of the hammer to either of the two target masses (assumed equal).

(a) Find an expression for the energy of the system as a whole, and show that it is conserved.

(b) Extend the system above to the case of a hammer striking N equal masses.

(c) Extend the system above to the case of a hammer striking two unequal masses, and find a generalized expression for the conserved energy.

Problem 4.6 A single stage in a Moog ladder filter [117] is often modelled using a first-order differential equation similar to the following:

$$\frac{du}{dt} = -\omega \tanh(u)$$

This is the zero-input case—suppose that the system possesses a single initial condition $u(0) = u_0$.

(a) Show that if one defines the energy of this system as $\mathfrak{H} = \frac{1}{2}u^2$, then it must be true that $d\mathfrak{H}/dt \leq 0$, and thus that $|u(t)|$ is a monotonically decreasing function of time, and is bounded by $|u_0|$.

(b) Suppose that the differential equation is discretized as

$$\delta_{t-}u = -\omega \tanh(\mu_{t-}u)$$

where $u = u^n$ is a time series, initialized with a value u^0 . Show that again, if one defines the energy as $\frac{1}{2}u^2$, $|u^n|$ is a monotonically decreasing function of time index n , and is bounded by $|u^0|$.

(c) The difference scheme above is implicit. Using identity (2.7e), it may be written as

$$\mu_{t-u} = -\frac{k\omega}{2} \tanh(\mu_{t-u}) + e_{t-u}$$

Using a diagram, and the fact that e_{t-u} is known, show that the recursion possesses a unique solution at each time step.

Problem 4.7 Consider again the uncoupled bow-like oscillator, as given in (4.19). Instead of scheme (4.21b), examine the scheme given by

$$\delta_{tt}u = -\alpha\mu_{t-}\phi(\delta_{t+}u)$$

(a) Show that the above scheme, which is implicit, may be written as

$$\phi(\delta_{t+}u) = \frac{-2\alpha}{k}\delta_{t+}u + q$$

where q depends only on previously computed values of u . Show that a solution to the above implicit equation for $\delta_{t+}u$ possesses a unique solution under condition (4.23).

(b) Multiply the above scheme by $\delta_{t-}u$, and show that the following energy balance results:

$$\delta_{t+}\mathfrak{h} = \frac{-\alpha}{4k}(\delta_{t+}u + \delta_{t-}u)(\phi(\delta_{t+}u) + \phi(\delta_{t-}u))$$

where \mathfrak{h} is again defined as in (4.22). It is difficult to find simple conditions under which the right side of the above energy balance is non-positive (which is desirable in order to have passive behaviour, and thus numerical stability). Can you find a simple sufficient condition on ϕ such that the scheme is passive for any value of the time step k ? Hint: the characteristics shown in the left and center panels in Figure 4.8 allow such a simple condition, but that shown at right does not.

Problem 4.8 Consider scheme (4.26) for the bowed mass-spring system. Show that it may be written as

$$\frac{2}{k}v_{rel} + F_B\phi(v_{rel}) + b = 0 \quad (4.27)$$

and determine b , which consists of known values (i.e., of previous computed values of the mass displacement, and the values of the time series v_B). For the two friction characteristics

$$\phi(\eta) = \sqrt{2\sigma}e^{-\sigma\eta^2+1/2} \quad \phi(\eta) = \text{sign}(\eta) \left(\alpha + (1-\alpha)e^{-\sigma\eta^2} \right) \quad (4.28)$$

where $0 \leq \alpha \leq 1$, explicitly determine the bounds on k for the uniqueness of the solution to the equation in v_{rel} above.

4.5 Programming Exercises

Exercise 4.1 For the lossless oscillator with a power law characteristic (4.12), compare the performance of difference schemes (4.4) and (4.13). For a given value of the power law exponent $\alpha > 1$ (consider the values $\alpha = 1.5, 2, 2.5, 3, 3.5, 4$), and with initial conditions $u(0) = 1$, $du/dt(0) = 1$, find the value of ω for which each of the schemes become unstable. How does this instability manifest itself in the two cases? Can you conclude that one scheme possesses better stability properties than the other?

Exercise 4.2 Consider the hammer mass-spring interaction, discussed in §4.2.3. Generalize the finite difference approximation given in (4.17) to the case of a hammer striking two mass-spring systems, as described in Problem 4.5. Implement this scheme, using the code given in §A.2 as a starting point. Your code should plot the displacements of the two masses, as well as the combined force $F_1 + F_2$ as a function of time.

Exercise 4.3 Consider a rattling element coupled to a mass-spring system. Not surprisingly, this system and a finite difference scheme can be written in exactly the same way as the hammer/mass-spring interaction, as in (4.16) and (4.17), but where the nonlinear characteristic F is replaced by that which appears in (4.18). Modify the code example in §A.2, so that it simulates such a

rattle/mass-spring interaction. You will need to introduce the rattle length ϵ as an extra input parameter.

Exercise 4.4 Extend the simulation of the rattle/mass-spring interaction in Programming Exercise 4.3 above to include the effects of gravity and loss (in the target mass-spring system). Now, the system must be generalized to

$$\frac{d^2u}{dt^2} = -\omega_0^2u - 2\sigma_0 \frac{du}{dt} + \mathcal{M}F(u - u_R) \quad \frac{d^2u_R}{dt^2} = -F(u - u_R) - g \quad (4.29)$$

where u and u_R are the displacements of the target mass spring system and the rattle, respectively, σ_0 is the usual loss parameter for the SHO (see §3.5), and where $g = 9.8\text{m} \cdot \text{s}^{-2}$ is the acceleration due to gravity.

Develop and program a finite difference scheme for the above system. How would you introduce a loss mechanism into the rattling element itself?

Exercise 4.5 Modify the code example given in §A.3 such that the bow force F_B and velocity v_B are time series $F_B = F_B^n$ and $v_B = v_B^n$. Experiment with different gestural profiles for both of these control parameter sets. For example, one choice of a control signal for, say, F_B might be, in continuous time and defined over the interval T_f ,

$$F_B(t) = F_{B,\max} \begin{cases} t/\alpha, & 0 \leq t \leq \alpha \\ (T_f - t)/(T_f - \alpha), & \alpha \leq t \leq T_f \end{cases}$$

for some peak bow force $F_{B,\max}$ and a parameter α , with $0 \leq \alpha \leq T_f$ which determines the time to the occurrence of the peak. A similar function could be defined for the velocity. Determine, if possible, the effect on the time taken for the mass-spring system to reach a stable oscillatory regime.

Exercise 4.6 Newton-Raphson Iterative Root Finding In some numerical bow models, such as that given in (4.26) (as well as in a vast array of other non-musical applications), it is necessary to solve an equation of the form

$$f(\eta) = 0$$

at each time step. The Newton-Raphson iterative root-finding procedure works as follows: Given some initial guess at the solution $\eta^{(0)}$, perform the following update,

$$\eta^{(p+1)} = \eta^{(p)} - \frac{f(\eta^{(p)})}{f'(\eta^{(p)})}$$

successively, until an adequately stable solution is found. (You might wish to measure convergence in terms of the error $|\eta^{(p+1)} - \eta^{(p)}|$, and terminate the algorithm when this error is sufficiently small.)

Referring to Problem 4.8, for the bowed mass-spring system, the nonlinear equation to be solved, in the relative velocity v_{rel} , is exactly (4.27). For given values of b , F_B and k , write a Matlab function which returns v_{rel} , for both choices of friction characteristic given in (4.28), using an input parameter to indicate the type of characteristic. In the second case, α appears as an addition parameter, and you will need to take care to divide your algorithm into cases, as the characteristic is not continuous. Your function should also perform a test to ensure that condition (4.23) is satisfied, and produce a warning if it is not. You may wish to refer to the code example in §A.3, in which Newton-Raphson for the case of the continuous friction characteristic is employed.

Exercise 4.7 Modify the code example given in §A.3 so that it uses the Matlab function for the Newton-Raphson iterative algorithm that you created in the previous exercise. Compare the behaviour of the system under the distinct choices of friction characteristic from (4.28).

Chapter 5

Grid Functions and Finite Difference Operators in One Dimension

In this short chapter, the basic operations used in the construction of finite difference schemes for 1D distributed problems are introduced. The treatment will be somewhat abbreviated, partly because much of this material appears elsewhere, and also because some of the underlying ideas have been introduced in Chapter 2. As was the case in this earlier chapter, the presentation will again encompass both frequency domain analysis, and pure time-space (energy) techniques. The full power of energy methods [113] will become evident here, not just because of their capability for arriving at stability conditions even for strongly nonlinear problems (some important musical examples of which will appear in subsequent chapters), but because of the ease with which appropriate numerical boundary conditions may be extracted; the proper setting of boundary conditions is always problematic, and is not at all well covered even in the best finite difference texts. This is not to say that frequency domain techniques are not useful; as was mentioned earlier, they are able to yield much important information regarding numerical dispersion, which, in musical simulations, may lead to perceptually audible deviations from the solution to a model system.

In §5.1, partial differential operators are introduced, followed by a brief discussion of the classification of partial differential equations, transform techniques and the concepts of dispersion and phase velocity, and finally an introduction to inner product spaces and various manipulations of interest in PDE analysis. The extension of these techniques to the discrete case appears in §5.2, which introduces grid functions and difference operators, now in a distributed setting. Finally, some material on coordinate changes, useful when the problem under consideration exhibits some spatial variation, appears in §5.3. The extension of these concepts to two spatial dimensions is postponed until Chapter 10.

5.1 Partial Differential Operators

Some elements of musical instruments, in particular strings, bars, and tubes, are well-modelled by partial differential equations (PDEs) in time t and one spatial dimension x ; such systems are often referred to as “1D.” Various quantities, such as displacement, velocity, and pressure are thus described by functions of two variables, such as $u(x, t)$.

As in the case of lumped systems such as the harmonic oscillator (see Chapter 3), time t is usually defined for $t \geq 0$, though one may extend this definition to $t \in [-\infty, \infty]$, for purposes of steady-state

(Laplace transform) analysis. One also normally takes $x \in \mathcal{D}$, where \mathcal{D} represents some subset of the real line. There are essentially three domains \mathcal{D} of interest in problems defined in 1D, the infinite domain $\mathcal{D} = \mathbb{R} = [-\infty, \infty]$, the semi-infinite domain $\mathcal{D} = \mathbb{R}^+ = [0, \infty]$, and the unit interval $\mathcal{D} = \mathbb{U} = [0, 1]$. The first two domains are of use in an analysis setting; in one-dimensional problems in musical acoustics, $\mathcal{D} = \mathbb{U}$ is always chosen. It is worth noting that non-dimensionalization techniques, employed whenever possible in this book (see §6.1.2 for a first example of this in the case of the 1D wave equation) render the use of other finite intervals unnecessary.

PDEs are equations relating partial temporal and spatial derivatives of one or more functions. In operator form, such derivatives are written as

$$\frac{\partial}{\partial t}, \quad \frac{\partial^2}{\partial t^2}, \quad \frac{\partial}{\partial x}, \quad \frac{\partial^2}{\partial x^2}, \quad \text{etc.}$$

When applied to a function such as $u(x, t)$, the abbreviated subscript notation will be frequently employed, i.e.,

$$\frac{\partial u}{\partial t} = u_t, \quad \frac{\partial^2 u}{\partial t^2} = u_{tt}, \quad \frac{\partial u}{\partial x} = u_x, \quad \frac{\partial^2 u}{\partial x^2} = u_{xx}, \quad \text{etc.}$$

For total time derivatives of a quantity without spatial dependence, such as those that occur in energy analysis or a lumped setting, the symbol $\frac{d}{dt}$ is uniformly employed.

A PDE in a single variable $u(x, t)$, under zero-input conditions, may be simply written as

$$P \left(\frac{\partial}{\partial t}, \frac{\partial^2}{\partial t^2}, \frac{\partial}{\partial x}, \frac{\partial^2}{\partial x^2}, \dots \right) u = 0 \tag{5.1}$$

where P is a partial differential operator which is a complete description of the behaviour of the system at a given point—many of the systems under consideration in musical acoustics, both linear and nonlinear, may be written in this form. The notation above is somewhat loose; not indicated is the possibility of coefficients in the PDE which may depend on x , which may indeed occur in describing systems with a degree of variation in material parameters. Coefficients which depend on t (rendering the system time-varying) are possible as well, but occur rarely in musical acoustics. In general, for the systems of interest here, time derivatives of order greater than two appear infrequently, and for many (but not all), spatial derivatives are of even order, reflecting independence of the problem to the direction of propagation of the solution. In some cases, a set of one or more coupled equations of the form (5.1) in several independent variables is necessary in describing a system; nonlinear systems such as strings and plates under high amplitude vibration conditions are two such examples—see Chapters 8 and 13. A description such as (5.1) is not complete until initial and boundary conditions have been supplied. Appropriate settings for such conditions will be dealt with on an as-needed basis, and the first description will occur with reference to the 1D wave equation, in the following chapter.

5.1.1 Classification of PDEs

In an abstract setting, there are established ways of classifying PDEs [103], which are useful in making general statements about the form of the solution. One often speaks of a PDE of being time-dependent or not, or of being hyperbolic or parabolic or elliptic, or of being semi-linear versus quasi-linear. In the more specialized setting of musical acoustics, it is better to abandon this abstract view, and focus on the three very particular types of systems (always time-dependent, and usually hyperbolic) which frequently occur: linear and shift-invariant (LSI) systems, linear and time-invariant (LTI) systems, and nonlinear systems. As an example, consider the following equation:

$$u_{tt} = \gamma^2 u_{xx}$$

where $u(x, t)$ is the dependent variable to be solved for. If γ is a constant, this PDE is known as the 1D wave equation, and is LSI—the medium it describes behaves uniformly at all points in the domain, and at all time instants. If γ depends on x , the system is still LTI, but not generally invariant with respect to the spatial variable, and not LSI. Its properties are different from one spatial location to another. Finally, if γ depends on u itself, the equation is referred to as nonlinear. This classification follows through to any derived numerical simulation method.

The analysis and numerical method construction techniques which are available are governed by the type of system under consideration. LSI systems, which form the majority of those under study in musical acoustics, and employed until recently in physical modeling sound synthesis, are undeniably the easiest to deal with—the full scope of frequency domain and transform techniques is available, and as a result, it is possible to obtain a very comprehensive picture of the resulting dynamics including information about wave propagation speeds and modes, as well as many features of associated numerical methods, such as stability conditions and dispersion. Such systems and the accompanying analysis machinery, as they occur in musical acoustics will be examined in Chapters 6, 7, 11 and 12. Systems which are LTI but not LSI still allow a limited analysis: one may still speak of modes and frequencies, but one loses the convenient notion of a global wave velocity, and the most useful analysis tools (von Neumann) for numerical simulation methods are no longer available. Some such systems will be seen in §7.9, and in Chapter 9. In the nonlinear case, the notion of frequency itself is, strictly speaking, not valid, though if the nonlinearity is a weak one, one may draw some (generally qualitative) conclusions through linearization, or, with much additional effort, through perturbation methods [178]. In all cases, however, it is possible, and usually quite easy to arrive at a numerical simulation routine for synthesis, but depending on the type of the original system, it will be more or less difficult to say or predict anything about the way in which it behaves. See Chapters 8 and 13.

One attribute of a system which persists in being clearly defined for any of the types of system mentioned above is its energetic behaviour. PDEs which occur in musical acoustics are not abstract concoctions—they correspond to real physical systems, and real physical systems always obey laws of conservation or dissipation of energy. Though energy analysis of a PDE system is, as a rule, much less revealing than what can be obtained using frequency domain techniques, it unfailingly points in the right direction as far as simulation design is concerned.

5.1.2 Laplace and Fourier Transforms

As in the lumped case, the Laplace transform of a function such as $u(x, t)$ may be defined as

$$\hat{u}(x, s) = \int_{-\infty}^{\infty} u(x, t)e^{-st} dt \quad (5.2)$$

where $s = j\omega + \sigma$ is again the complex-valued frequency variable. See §2.3. As before, the definition is two-sided, allowing one to ignore initial conditions.

One may also define a spatial Fourier transform, as

$$\tilde{u}(\beta, t) = \int_{-\infty}^{\infty} u(x, t)e^{-j\beta x} dx \quad (5.3)$$

where here, β may be viewed as a real wavenumber; in general, spatial Fourier transforms are only used for problems defined over $\mathcal{D} = \mathbb{R}$, though Fourier series approximations for systems defined over a finite interval play a fundamental role in so-called modal synthesis methods—see, e.g., §6.1.11.

The actions of partial derivative operators applied to a function $u(x, t)$ transform according to

$$\frac{\partial^m u}{\partial t^m} \xrightarrow{\mathcal{L}} s^m \hat{u} \quad \frac{\partial^l u}{\partial x^l} \xrightarrow{\mathcal{F}} (j\beta)^l \tilde{u}$$

where \mathcal{L} and \mathcal{F} accompanied by an arrow indicate a Laplace and Fourier transform, respectively.

Just as in the lumped case, as an alternative to full Laplace/Fourier analysis, it is often simpler, conceptually, to consider the effect of differential operators on a test function of the form

$$u(x, t) = e^{st+j\beta x}$$

which behaves as a single wavelike component, of temporal frequency s , and wavenumber β . The use of this special function is often referred to in the literature as an *ansatz* [113]. See §2.3.

Dispersion Relation

If the PDE under consideration is LSI, then both Laplace and Fourier transforms may be applied in order to arrive at a convenient algebraic description of the PDE in the frequency domain:

$$Pu = 0 \quad \xrightarrow{\mathcal{L}, \mathcal{F}} \quad \tilde{P}\tilde{u} = 0$$

where $\tilde{P} = \tilde{P}(s, j\beta)$, a multinomial in s and $j\beta$, is often referred to as the symbol of the PDE defined by (5.1). Because $\tilde{P}\tilde{u}$ is now a product of two functions of s and $j\beta$, it is easy to see that non-trivial solutions can only occur when

$$\tilde{P}(s, j\beta) = 0$$

This characteristic equation defines various relations between temporal frequency s and spatial frequency β . These will be of the form $s_p = s_p(j\beta)$, for $p = 1, \dots, M$, where M is the order of the highest temporal derivative appearing in P , or the highest power of s in $\tilde{P}(s, j\beta)$. (For most systems of interest in this book, $M = 2$, and the two solutions will be denoted by s_{\pm} , which generally describe left-going and right-going wavelike solutions to the PDE.) These solutions are known as dispersion relations, and allow the extraction of much useful information regarding propagation speeds.

Phase and Group Velocity

In lossless problems, it is typical to arrive at solutions $s_p(j\beta)$ which are purely imaginary, so that one may write the dispersion relations as $\omega_p = \omega_p(\beta)$ for real frequencies ω_p . In this special case, for any such $\omega(\beta)$, the phase velocity v_ϕ and group velocity v_g are defined as

$$v_\phi = \frac{\omega}{\beta} \quad v_g = \frac{d\omega}{d\beta} \quad (5.4)$$

Generally, for a problem of order M in the time variable, there will be M dispersion relations. For second order systems, these usually occur as a complex conjugate pair, leading, from the above definitions, to a pair of velocities of opposite sign, representing wave propagation to the left and right. In this book, the positive root will be taken as the phase velocity. These velocities may be expressed most directly as functions of wavenumber, but more usefully perhaps as functions of frequency ω , (or what is more directly comprehensible in musical acoustics, frequency $f = \omega/2\pi$) through the substitution of the dispersion relation itself; this is always possible in 1D, but extends to higher dimensions only for problems which are isotropic. See §10.1.4 for a discussion of the 2D case.

In general, for a given system, the phase velocity describes the speed of propagation of a single component of the solution, and the group velocity the gross speed of disturbances. For hyperbolic systems, such as the wave equation, it must be true that the group velocity is bounded from above

(i.e., there is a maximum speed at which disturbances may propagate), though this is not necessarily true of the phase velocity—the case of Timoshenko beam theory is an interesting example of this [111]. Most systems that will be examined in this book will be of hyperbolic type, with the exception of thin beams and plates, which do in fact allow infinite group velocities. Such anomalous behaviour is a result of simplifying assumptions, and disappears when more accurate models are employed. If a system exhibits a small degree of loss—this is the majority of systems in musical acoustics—then one may loosely extend this idea of phase velocity by replacing ω in the definitions above by the imaginary part of s .

These concepts carry over to derived numerical methods in a natural way—finite difference schemes, at least for LSI systems, also possess phase and group velocities. Both are of use in coming to conclusions, in the musical sound synthesis context, about perhaps undesirable numerical inharmonicity, as well as stability [258].

5.1.3 Inner Products and Energetic Manipulations

Energy analysis of PDE systems is based, usually, around the definition of various types of spatial inner products [148]. A very useful choice in the continuous case is the L_2 inner product. In 1D, it, along with the accompanying norm, is defined as

$$\langle f, g \rangle_{\mathcal{D}} = \int_{\mathcal{D}} f g dx \quad \|f\|_{\mathcal{D}} = \sqrt{\langle f, f \rangle_{\mathcal{D}}} \quad (5.5)$$

for functions $f(x)$ and $g(x)$ defined over the interval $x \in \mathcal{D}$. For time-dependent problems such as those encountered in this book, such an inner product when applied to two functions $f(x, t)$ and $g(x, t)$ will itself be a function of time alone, i.e., $\langle f, g \rangle = \langle f, g \rangle(t)$.

Identities and Inequalities

The Cauchy-Schwartz and triangle inequalities follow from the above definition as

$$|\langle f, g \rangle_{\mathcal{D}}| \leq \|f\|_{\mathcal{D}} \|g\|_{\mathcal{D}} \quad \text{Cauchy – Schwartz inequality} \quad (5.6a)$$

$$\|f + g\|_{\mathcal{D}} \leq \|f\|_{\mathcal{D}} + \|g\|_{\mathcal{D}} \quad \text{triangle inequality} \quad (5.6b)$$

For any three functions f , g and r , it is also true that

$$\langle f, gr \rangle_{\mathcal{D}} = \langle fg, r \rangle_{\mathcal{D}}$$

If the left and right endpoints of a given domain \mathcal{D} are d_- and d_+ , then the familiar integration by parts rule may be written, in inner product notation, as

$$\langle f, g_x \rangle_{\mathcal{D}} = -\langle f_x, g \rangle_{\mathcal{D}} + fg \Big|_{d_-}^{d_+} \quad (5.7)$$

The identity (5.7) may be extended to second derivatives as

$$\langle f, g_{xx} \rangle_{\mathcal{D}} = \langle f_{xx}, g \rangle_{\mathcal{D}} + (fg_x - f_xg) \Big|_{d_-}^{d_+} \quad (5.8)$$

Generally, if $d_- = -\infty$ or $d_+ = \infty$, then the functions f and g are assumed to vanish when evaluated at such points. Thus, when $\mathcal{D} = \mathbb{R}$, for example, one has simply

$$\langle f, g_x \rangle_{\mathbb{R}} = -\langle f_x, g \rangle_{\mathbb{R}} \quad \langle f, g_{xx} \rangle_{\mathbb{R}} = \langle f_{xx}, g \rangle_{\mathbb{R}} \quad (5.9)$$

The total time derivative of an inner product is assumed to distribute to the constituent functions, i.e.,

$$\frac{d}{dt} \langle f, g \rangle_{\mathcal{D}} = \langle f_t, g \rangle_{\mathcal{D}} + \langle f, g_t \rangle_{\mathcal{D}}$$

Notice the use of the symbol d/dt here to represent the total time derivative of a quantity without spatial dependence.

Vector Form

In some cases, it may be necessary to make use of a vector form of the inner product, namely, for column vectors $\mathbf{f} = [f^{(1)}, \dots, f^{(M)}]^T$ and $\mathbf{g} = [g^{(1)}, \dots, g^{(M)}]^T$,

$$\langle \mathbf{f}, \mathbf{g} \rangle_{\mathcal{D}} = \int_{\mathcal{D}} \mathbf{f}^T \mathbf{g} dx$$

A particular application is to the analysis of nonplanar string vibration—see §8.3.1. The above identities and inequalities generalize directly to the vector case in an obvious way.

5.2 Grid Functions and Difference Operators in One Dimension

A grid function u_l^n , taking on values for integer n (usually $n \geq 0$), and for integer $l \in \mathcal{D}$, is introduced in order to approximate a continuous function $u(x, t)$, at location $x = lh$, and at time $t = nk$. Again, as in the case of time series, the same variable name (in this case u) will be used to represent both a grid function, and the variable it is intended to approximate. See Figure 5.1.

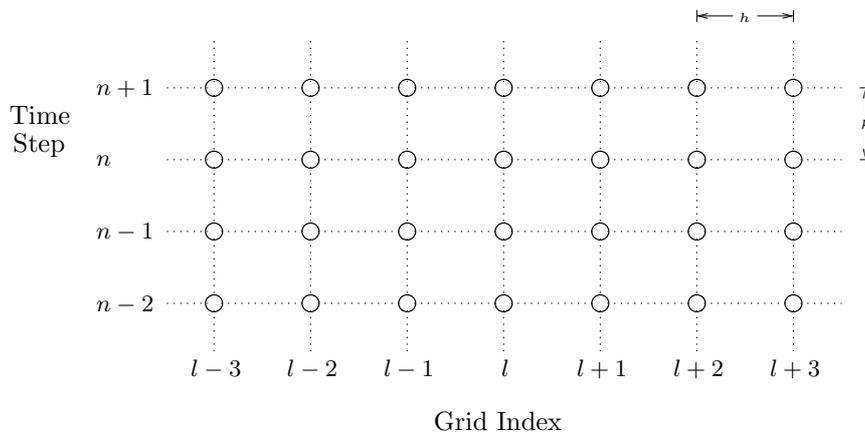


Figure 5.1: Graphical representation of a grid suitable for a 1D problem, with a spacing between adjacent grid points of h (in the horizontal direction) and of k (in the vertical direction).

As in the continuous case, there are essentially three domains \mathcal{D} of interest for problems in one spatial dimension: $\mathcal{D} = \mathbb{Z} = [-\infty, \dots, \infty]$, $\mathcal{D} = \mathbb{Z}^+ = [0, \dots, \infty]$, and $\mathcal{D} = \mathbb{U}_N = [0, \dots, N]$. The finite domain \mathbb{U}_N is the only domain of practical interest, and generally corresponds to the unit interval, provided one has chosen a grid spacing of $h = 1/N$, which one is always free to do through non-dimensionalization techniques. For the moment, for simplicity, it will be assumed that grid functions are infinite in spatial extent, though the bounding of spatial domains will be introduced in §5.2.9.

For reference, graphical representations of the behaviour of the various operators described in this section appear in Figure 5.2.

5.2.1 Time Difference and Averaging Operators

The definitions of time difference operators in the distributed setting are nearly unchanged from those introduced in §2.2 and applied to time series. For a grid function u_l^n , the forward and backward shifts, and the identity operation “1” are defined as

$$e_{t+}u_l^n = u_l^{n+1} \quad e_{t-}u_l^n = u_l^{n-1} \quad 1u_l^n = u_l^n$$

and are to be regarded as applying to the time series u_l^n at all values of the index n and l . Approximations to a partial time derivative are defined as

$$\delta_{t+} \triangleq \frac{1}{k}(e_{t+} - 1) \approx \frac{\partial}{\partial t} \quad \delta_{t-} \triangleq \frac{1}{k}(1 - e_{t-}) \approx \frac{\partial}{\partial t} \quad \delta_t \triangleq \frac{1}{2k}(e_{t+} - e_{t-}) \approx \frac{\partial}{\partial t}$$

and averaging approximations to the identity operation as

$$\mu_{t+} \triangleq \frac{1}{2}(e_{t+} + 1) \approx 1 \quad \mu_{t-} \triangleq \frac{1}{2}(1 + e_{t-}) \approx 1 \quad \mu_t \triangleq \frac{1}{2}(e_{t+} + e_{t-}) \approx 1$$

A simple approximation to a second time derivative is, as before,

$$\delta_{tt} = \delta_{t+}\delta_{t-} \approx \frac{\partial^2}{\partial t^2}$$

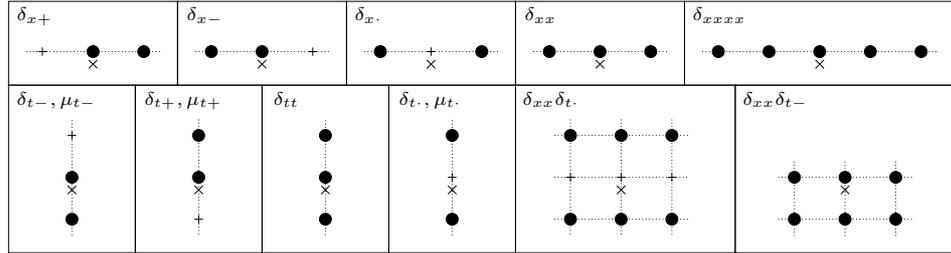


Figure 5.2: Footprints of various simple 1D difference operators, as indicated. Time shifts, of duration k , are shown as vertical displacements, and spatial shifts, of distance h by horizontal displacements. In each case, the grid points required for the approximation are indicated by black circles, and the central point at which the operator is applied is accompanied by a cross symbol.

5.2.2 Spatial Difference Operators

Approximations to spatial derivatives are based on the unit forward and backward spatial shift operations, defined as

$$e_{x+}u_l^n = u_{l+1}^n \quad e_{x-}u_l^n = u_{l-1}^n$$

One has, immediately, the following forward, backward and centered spatial difference approximations,

$$\delta_{x+} \triangleq \frac{1}{h}(e_{x+} - 1) \approx \frac{\partial}{\partial x} \quad \delta_{x-} \triangleq \frac{1}{h}(1 - e_{x-}) \approx \frac{\partial}{\partial x} \quad \delta_{x.} \triangleq \frac{1}{2h}(e_{x+} - e_{x-}) \approx \frac{\partial}{\partial x}$$

and spatial averaging operators similar to their temporal counterparts, can be defined accordingly:

$$\mu_{x+} = \frac{1}{2}(e_{x+} + 1) \quad \mu_{x-} = \frac{1}{2}(1 + e_{x-}) \quad \mu_{x\cdot} = \frac{1}{2}(e_{x+} + e_{x-}) \quad \mu_{xx} = \mu_{x+}\mu_{x-}$$

Simple approximations to second and fourth spatial derivatives are given by

$$\delta_{xx} = \delta_{x+}\delta_{x-} \approx \frac{\partial^2}{\partial x^2} \quad \delta_{xxxx} = \delta_{xx}\delta_{xx} \approx \frac{\partial^4}{\partial x^4}$$

For ease of programming, it is useful to expand out the behaviour of these compact operator representations in order to see directly the coefficients which should be applied to values of a given grid function u_l^n :

$$\delta_{xx}u_l^n = \frac{1}{h^2}(u_{l+1}^n - 2u_l^n + u_{l-1}^n) \quad (5.10a)$$

$$\delta_{xxxx}u_l^n = \frac{1}{h^4}(u_{l+2}^n - 4u_{l+1}^n + 6u_l^n - 4u_{l-1}^n + u_{l-2}^n) \quad (5.10b)$$

There are various identities which are of use in difference scheme analysis—here are two of particular interest.

$$\delta_{x+}(fg) = (\mu_{x+}f)(\delta_{x+}g) + (\delta_{x+}f)(\mu_{x+}g) \quad (5.11)$$

$$\mu_{x\cdot} = 1 + \frac{h^2}{2}\delta_{xx} \quad (5.12)$$

Just as in the case of temporal operators, spatial difference operators may be combined to yield more complex operators, generally of wider stencil. One particularly useful form, dependent on a free parameter α , is the averaging operator

$$\alpha + (1 - \alpha)\mu_c.$$

which is an approximation to the identity for any value of α . A parameterized form such as the above can lead to a means of reducing the dispersive properties of finite difference schemes—see §6.3.

5.2.3 Mixed Spatial-Temporal Difference Operators

In some cases of interest in musical acoustics, and in particular in models of frequency-dependent loss in strings (see §7.3), mixed derivative terms appear. It is straightforward to generate finite difference approximations to such operators using operator notation, and principles of combining difference operators, as outlined in §2.2.2. Considering, for example, the operator $\frac{\partial^3}{\partial t \partial x^2}$, two possible approximations are

$$\delta_{t-}\delta_{xx} \quad \delta_{t\cdot}\delta_{xx}$$

The first approximation employs a backward time difference, and the second a centered time difference. Again, it is of use to examine the effect of such an operator applied to a grid function u_l^n :

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

$$\delta_{t\cdot}\delta_{xx}u_l^n = \frac{1}{2kh^2}(u_{l+1}^{n+1} - 2u_l^{n+1} + u_{l-1}^{n+1} - u_{l+1}^{n-1} + 2u_l^{n-1} - u_{l-1}^{n-1})$$

In practical terms, there is rather a large distinction between the two approximations. That involving the backward time difference computes an approximation to $\frac{\partial^3}{\partial t \partial x^2}$ at grid location (l, n) using only values of the grid function at time steps n and $n - 1$. That involving the centered time difference, however, requires access to values of the grid function at time step $n + 1$. In a general two-step

finite difference scheme (most of the systems to be discussed in this book may be dealt with in this way), the values of the grid function at time step $n + 1$ constitute the unknowns. The use of such an operator then leads to a linear coupling among the unknown values of the grid function, and thus to the need for linear system solution techniques in order to arrive at a solution, and such schemes are known as “implicit” [244]. Such implicit schemes will be employed at times in this book. Though they may require more effort at the programming stage, the actual computational work of performing (potentially large) linear system inversions is alleviated somewhat by the sparsity of the difference operators when viewed in matrix form—see §5.2.7.

5.2.4 Interpolation and Spreading Operators

Input and output are key attributes of physical modeling synthesis routines. The problem in particular of taking output, though potentially very simple to deal with, is not often covered in mainstream simulation texts.

Output and Interpolation

In some cases in musical sound synthesis, one may be interested in accessing (listening to) a point in a distributed medium which lies between spatial grid points—some form of interpolation is thus necessary. Though if the listening, or observation point is static, it is simple enough to truncate this position to a nearby grid location, if it is moving¹, such truncation will inevitably lead to audible distortion (clicks) in the resulting sound output. See §6.2.9.

Consider some one-dimensional grid function u_l , consisting of values at locations lh for integer l , which represents the state of a simulated object at some given instant. Supposing that the observation point is $x_o \in \mathcal{D}$, leftward truncation leads to an integer observation index $l_o = \text{floor}(x_o/h)$, and one may define an interpolation operator $I(x_o)$, acting on the grid function u_l , as

$$I_0(x_o)u_l = u_{l_o}$$

One could, as in sound synthesis methods employing wavetable interpolation [170], make use of a slightly improved interpolation operator which employs rounding rather than truncation. Better yet, if one also makes use of the fractional remainder of the truncation, defined by $\alpha_o = x_o/h - l_o$, one may define a linear interpolation operator $I_1(x_o)$ by

$$I_1(x_o)u_l = (1 - \alpha_o)u_{l_o} + \alpha_o u_{l_o+1}$$

One may of course go further along these lines, to develop interpolants of even better accuracy. For sound synthesis, linear interpolation is often sufficient, but Lagrange cubic interpolation is another possible choice. For a given observation index l_o and fractional part α_o , the cubic interpolant $I_3(x_o)$ is defined as

$$\begin{aligned} I_3(x_o)u_l = & \frac{\alpha_o(\alpha_o - 1)(\alpha_o - 2)}{-6}u_{l_o-1} + \frac{(\alpha_o - 1)(\alpha_o + 1)(\alpha_o - 2)}{2}u_{l_o} \\ & + \frac{\alpha_o(\alpha_o + 1)(\alpha_o - 2)}{-2}u_{l_o+1} + \frac{\alpha_o(\alpha_o + 1)(\alpha_o - 1)}{6}u_{l_o+2} \end{aligned}$$

This approximation makes use of four values of the grid function u_l neighboring the interpolation point, and requires a good deal more computational work than linear interpolation. It is, on the

¹Moving output locations, though distinctly unphysical, can be employed as a useful and very simple “hack” to render physical modeling sound synthesis output slightly more attractive, due to delicate phasing effects. Such effects do indeed occur in the real world (when, for example, a musical instrument moves slightly with respect to the listener), but are difficult to model directly.

other hand, much more accurate than linear interpolation—see §5.2.5. The action of these interpolation operators is illustrated in Figure 5.3. One programming concern is the behaviour of such an interpolator when the observation point is within one grid spacing of the endpoint of the domain. Though not of major importance, there are several options: 1) restrict the observation point such that this cannot occur, 2) develop an asymmetric interpolant at such points, 3) revert to a simpler interpolant, such as I_1 for such locations, or 4) make use of imaginary grid points whose values are set by boundary conditions. Of these options, 1) is certainly the simplest, and 4) the most general and least likely to introduce artifacts. See Programming Exercise 5.1.

Another approach which has been employed, particularly when the sound synthesis algorithm is based around the use of signal processing constructs such as delay lines, has been all-pass interpolation. See the article by Laakso et al. for an overview [153].

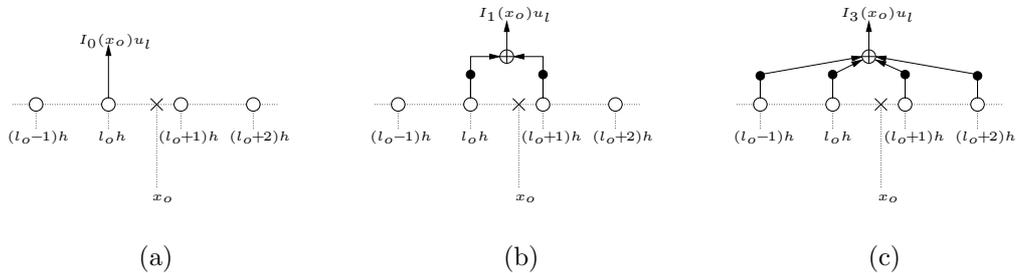


Figure 5.3: Schematic representation of various interpolation functions over a grid: (a) Rounding, (b) linear interpolation, and (c) cubic interpolation. Multiplications by scaling factors, dependent on the interpolation point, are indicated by black dots.

Input and Spreading Operators

The flip side of output is, of course, input; exciting a given distributed object at a given location is a key feature of string instruments in particular, where the excitation may be a strike, pluck, or bow force. In general, the excitation itself is often described as a lumped object, whose effect on a 1D system may be expressed in terms of a distribution of a given form.

The most useful such distribution corresponds to an excitation at a single location $x_i \in [0, 1]$. In the continuous case, this is exactly a Dirac delta function $\delta(x-x_i)$. As in the case of interpolation (see previous section), there are various ways of approximating such a distribution on a grid of spacing h . If $l_i = \text{floor}(x_i/h)$ is the nearest grid point to the left of x_i , then a zeroth order distribution $J_{l_i,0}(x_i)$ is a grid function of the form

$$J_{l_i,0}(x_i) = \frac{1}{h} \quad \text{for} \quad l = l_i, \quad \text{otherwise } 0$$

Notice in particular that the spreading function is scaled by $1/h$. Such a distribution is sufficient for many applications, especially if the grid is dense (i.e., h is small), and if the excitation remains stationary. But this is not always the case, and higher-order distributions may be necessary in some cases. In analogy with the interpolation operators I_1 and I_3 , the linear and cubic spreading distributions, respectively $J_{l_i,1}$ and $J_{l_i,3}$, may be defined, in terms of the fractional address $\alpha_i =$

$x_i/h - l_i$ as

$$J_{l,1}(x_i) = \frac{1}{h} \begin{cases} 0 & l < l_i \\ (1 - \alpha_i) & l = l_i \\ \alpha_i & l = l_i + 1 \\ 0 & l > l_i + 1 \end{cases} \quad J_{l,3}(x_i) = \frac{1}{h} \begin{cases} 0 & l < l_i - 1 \\ \frac{\alpha_i(\alpha_i-1)(\alpha_i-2)}{-6} & l = l_i - 1 \\ \frac{(\alpha_i-1)(\alpha_i+1)(\alpha_i-2)}{2} & l = l_i \\ \frac{\alpha_i(\alpha_i+1)(\alpha_i-2)}{-2} & l = l_i + 1 \\ \frac{\alpha_i(\alpha_i+1)(\alpha_i-1)}{6} & l = l_i + 2 \\ 0 & l > l_i + 2 \end{cases}$$

As in the case of interpolation, higher order distribution functions are a possibility, though the higher the order, the wider the distribution function, and, again one may need to make use of a specialized distribution or programming if x_i is within a very few grid points of an endpoint of the domain.

Some more comments on the relationship between interpolation and spreading operators, and on the gain of spreading operators appear in §5.2.12.

5.2.5 Accuracy of Difference Operators

The concept of the accuracy of time/space difference operators is similar to that described in the case of pure time difference operators in §2.2.3, except that one must now take into account accuracy with respect to both dimensions. Generally, the difference operators described in the previous sections correspond to their continuous counterparts to within an additive factor which is dependent on powers of the time step k and the grid spacing h . Employing the same techniques as in §2.2.3, i.e., applying a difference operator to a continuous function and then making use of a two-dimensional Taylor expansion about the grid point of operation, one may show that, for time difference operators,

$$\begin{aligned} \delta_{t+} &= \frac{\partial}{\partial t} + O(k) & \delta_{t-} &= \frac{\partial}{\partial t} + O(k) & \delta_t &= \frac{\partial}{\partial t} + O(k^2) \\ \mu_{t+} &= 1 + O(k) & \mu_{t-} &= 1 + O(k) & \mu_t &= 1 + O(k^2) \\ & & \delta_{tt} &= \frac{\partial^2}{\partial t^2} + O(k^2) \end{aligned}$$

and for spatial difference operators, that

$$\begin{aligned} \delta_{x+} &= \frac{\partial}{\partial x} + O(h) & \delta_{x-} &= \frac{\partial}{\partial x} + O(h) & \delta_x &= \frac{\partial}{\partial x} + O(h^2) \\ \delta_{xx} &= \frac{\partial^2}{\partial x^2} + O(h^2) & \delta_{xxxx} &= \frac{\partial^4}{\partial x^4} + O(h^2) \end{aligned}$$

For mixed time-space difference operators, such as approximations to $\partial^3/\partial t\partial x^2$, one has

$$\delta_{t-}\delta_{xx} = \frac{\partial^3}{\partial t\partial x^2} + O(k) + O(h^2) \quad \delta_t\delta_{xx} = \frac{\partial^3}{\partial t\partial x^2} + O(k^2) + O(h^2)$$

Notice that in all cases, symmetric difference operators lead, in general, to second order accuracy.

In general, the accuracy of a finite difference scheme may be linked to that of the constituent difference operators employed; one must be somewhat careful in coming to a direct conclusion, because in most cases, the time step and grid spacing must also related in a particular way so as to come to conclusions about numerical stability and convergence. This is a rather subtle point, and the reader is referred to the literature for more information [244].

The accuracy of the interpolation operators discussed in §5.2.4 may also be deduced through the use of Taylor expansions—all are approximations to a Dirac delta function centered at the observation point x_o . When such operators are applied to a continuous function $u(x)$, one has

$$I_0(x_o)u(x) = u(x_o) + O(h) \quad I_1(x_o)u(x) = u(x_o) + O(h^2) \quad I_3(x_o)u(x) = u(x_o) + O(h^4)$$

5.2.6 Frequency Domain Interpretation

As for continuous problems, one may define transforms of discrete grid functions. Assuming for the moment that the grid function u_l^n is defined over $n = [-\infty, \dots, \infty]$, and $l \in \mathcal{D} = \mathbb{Z}$, the discrete time z -transform and discrete spatial Fourier transform are defined by

$$\hat{u} = \sum_{n=-\infty}^{\infty} u_l^n z^{-n} \quad \tilde{u} = \sum_{l=-\infty}^{\infty} u_l^n e^{-jl\beta h}$$

In the case of the z transform, the variable z may be viewed as a complex number of the form e^{sk} , with s a complex frequency $s = \sigma + j\omega$. The factor z , though completely analogous to that used in transform analysis of digital filters is referred to as an amplification factor when employed in the context of finite difference scheme analysis (and is often denoted g instead of z [244]). In the spatial discrete Fourier transform, β is a real wavenumber.

As in the continuous case, it is sufficient to perform frequency domain analysis of difference operators and schemes with respect to the ansatz

$$u_l^n = z^n e^{jl\beta h}$$

For temporal operators, one has, as in the case of difference operators defined for lumped problems,

$$e_{t\pm} u \xrightarrow{\mathcal{Z}} z^{\pm 1} \hat{u}$$

and for the various first differences and averaging operators, one has:

$$\begin{aligned} \delta_{t+} u &\xrightarrow{\mathcal{Z}} \frac{1}{k} (z - 1) \hat{u} & \delta_{t-} u &\xrightarrow{\mathcal{Z}} \frac{1}{k} (1 - z^{-1}) \hat{u} & \delta_t u &\xrightarrow{\mathcal{Z}} \frac{1}{2k} (z - z^{-1}) \hat{u} \\ \mu_{t+} u &\xrightarrow{\mathcal{Z}} \frac{1}{2} (z + 1) \hat{u} & \mu_{t-} u &\xrightarrow{\mathcal{Z}} \frac{1}{2} (1 + z^{-1}) \hat{u} & \mu_t u &\xrightarrow{\mathcal{Z}} \frac{1}{2} (z + z^{-1}) \hat{u} \end{aligned}$$

The second difference δ_{tt} transforms according to

$$\delta_{tt} u \xrightarrow{\mathcal{Z}} \frac{1}{k^2} (z - 2 + z^{-1}) \hat{u}$$

Spatial difference operators behave in a similar way. One has

$$e_{x\pm} u \xrightarrow{\mathcal{F}} e^{\pm j\beta h} \tilde{u}$$

and for approximations to a first derivative

$$\delta_{x+} u \xrightarrow{\mathcal{F}} \frac{1}{h} (e^{j\beta h} - 1) \tilde{u} \quad \delta_{x-} u \xrightarrow{\mathcal{F}} \frac{1}{h} (1 - e^{-j\beta h}) \tilde{u} \quad \delta_{x\cdot} u \xrightarrow{\mathcal{F}} \frac{1}{2h} (e^{j\beta h} - e^{-j\beta h}) \tilde{u} = \frac{j}{h} \sin(\beta h) \tilde{u}$$

Notice in particular that the approximation to the operator $\delta_{x\cdot}$ acts as a pure imaginary multiplicative factor, a consequence of the centered anti-symmetric footprint of $\delta_{x\cdot}$.

The operators δ_{xx} and δ_{xxxx} transform according to

$$\delta_{xx} u \xrightarrow{\mathcal{F}} \frac{1}{h^2} (e^{j\beta h} - 2 + e^{-j\beta h}) \tilde{u} = -\frac{4}{h^2} \sin^2(\beta h/2) \tilde{u} \quad (5.13a)$$

$$\delta_{xxxx} u \xrightarrow{\mathcal{F}} \frac{1}{h^4} (e^{j\beta h} - 2 + e^{-j\beta h})^2 \tilde{u} = \frac{16}{h^4} \sin^4(\beta h/2) \tilde{u} \quad (5.13b)$$

Notice that both operators behave as pure real multiplicative factors, a consequence of the centered symmetric footprint of both—furthermore, δ_{xx} is non-positive and δ_{xxxx} is non-negative for any choice of wavenumber β .

Mixed temporal/spatial difference operators behave as multiplicative factors involving both z

and β . For the approximations to the third derivative $\partial^3/\partial t\partial x^2$, one has

$$\begin{aligned}\delta_t\delta_{xx}u &\xrightarrow{Z,F} \frac{-4}{kh^2}(1-z^{-1})\sin^2(\beta h/2)\hat{u} \\ \delta_t\delta_{xx}u &\xrightarrow{Z,F} \frac{-2}{kh^2}(z-z^{-1})\sin^2(\beta h/2)\hat{u}\end{aligned}$$

Recursions and Amplification Polynomials

The relationship between difference schemes and polynomials in the variable z was discussed earlier in §2.3.3, in the case of ODEs. For finite difference schemes for LSI PDEs, the situation is similar.

Confining attention for the moment to 1D LSI problems defined over $\mathcal{D} = \mathbb{Z}$, finite difference schemes are recursions of the following form:

$$\sum_{l' \in \mathbb{M}} \sum_{n'=0}^N a_{l'}^{(n')} u_{l-l'}^{n-n'} = 0$$

where as before, the scheme operates over $N+1$ adjacent time levels, and over a spatial width given by the set \mathbb{M} . The constants $a_{l'}^{(n')}$ are referred to as the scheme coefficients. One major distinction between the recursion above, operating over an entire grid function, and the recursion (2.10) in a time series is that it is not possible to isolate the unknowns unless all but one of the coefficients $a_{l'}^{(0)}$ are zero. If this is true, the scheme is referred to as explicit, and if not implicit (requiring more involved linear system solution techniques). Much more will be said about explicit and implicit schemes with regard to the 1D wave equation in the following chapter.

Employing the ansatz $u_l^n = z^n e^{jl\beta h}$ in the above recursion leads to a polynomial of the form

$$P(z) = \sum_{n'=0}^N a^{(n')}(\beta) z^{N-n'} = 0 \quad \text{where} \quad a^{(n')}(\beta) = \sum_{l' \in \mathbb{M}} a_{l'}^{(n')} e^{jl'\beta h} \quad (5.14)$$

which is often referred to as an amplification polynomial [244] for the associated scheme. Again, the order $N+1$ of the polynomial corresponds to the number of time levels at which the scheme operates. Now, however, the effect of spatial difference operators is exhibited in the coefficients $a^{(n')}$, which are generally dependent on the wavenumber β . Thus a crude criterion for stability is that the roots of such a polynomial remain bounded in magnitude by unity *for all wavenumbers* β . In particular, when $N=2$, as will be the case for most of the schemes to be discussed here, the simple conditions (2.14) must hold for all β .

Numerical Phase and Group Velocity

The definitions of phase and group velocities, for a scheme with a characteristic polynomial as defined by, say, (5.14), are directly analogous to those of the continuous case, as discussed briefly at the end of §5.1.2. For a lossless problem, behaving in a stable manner, one will have $|z|=1$, or $z = e^{j\omega k}$. The M roots of the characteristic polynomial may then be written as $z_p = z_p(\beta)$, or, using $z = e^{j\omega_p\beta}$, as $\omega_p = \omega_p(\beta)$. The phase and group velocities are again given, for any such root, as

$$v_\phi = \frac{\omega}{\beta} \quad v_g = \frac{d\omega}{d\beta}$$

In general, the extent to which these quantities are different from those of the continuous model system is referred to as numerical dispersion. It is perhaps best to leave this (quite important) topic until concrete examples have been presented in subsequent chapters.

As an example, consider the operator δ_{xx} , applied to a grid function u_l , defined now not over $\mathcal{D} = \mathbb{R}$, but the $N+1$ -point unit interval $\mathcal{D} = \mathbb{U}_N = [0, 1, \dots, N]$. At an interior point in the domain, all is well, and $\delta_{xx}u_l$ may be calculated as

$$\delta_{xx}u_l = \frac{1}{h^2} (u_{l+1} - 2u_l + u_{l-1}) \quad \text{for } l = 1, \dots, N-1$$

But at $l = 0$, and $l = N$, the operator δ_{xx} appears to require values at locations which lie outside the domain of definition of the grid function, namely u_{-1} and u_{N+1} . Such locations are sometimes referred to as “imaginary” grid points in a finite difference scheme—generally, though, boundary conditions come to the rescue, and allow the values at such points to be set in terms of the values in the interior.

For instance, one boundary condition which occurs frequently in musical acoustics, across a wide variety of applications, is the simple fixed condition, sometimes referred to as a Dirichlet type boundary condition for which the values of the grid function at the endpoints are constrained to be zero. In the present case of the operator δ_{xx} , the result is very simple: the values u_0 and u_N are set to zero permanently, and in fact need not be stored at all in an implementation. Slightly trickier is the case of zero spatial derivative conditions, often referred to as Neumann type conditions. Here is one approximation:

$$\delta_{xx}u_l \quad \text{for } l \in \mathbb{U}_N \quad \text{with} \quad \delta_x.u_0 = \delta_x.u_N = 0$$

The two conditions at $l = 0$ and $l = N$ are clearly second-order approximations to a zero derivative condition. Written in terms of values at the image locations, they read as $u_{-1} = u_1$ and $u_{N+1} = u_{N-1}$, and thus the action of the operator δ_{xx} at the endpoints can now be given, purely in terms of values on the domain interior, as

$$\delta_{xx}u_0 = \frac{2}{h^2} (u_1 - u_0) \quad \delta_{xx}u_N = \frac{2}{h^2} (-u_N + u_{N-1})$$

It is worth noting that this is but one type of approximation to a zero-derivative condition—the proper setting of numerical boundary conditions is an issue which will be taken up in greater detail in later chapters. See Problem 5.4 for another numerical variation on the zero-derivative condition, and Problem 5.2 for the more involved case of numerical boundary conditions applied to a fourth order spatial difference operator.

It is also revealing to examine the effect of boundary conditions in the matrix form of the operator. In the zero and zero-derivative cases mentioned above, the matrix operations \mathbf{D}_{xx} look like

$$\mathbf{D}_{xx} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} = \frac{1}{h^2} \underbrace{\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{bmatrix}}_{u_0 = u_N = 0} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} \quad \mathbf{D}_{xx} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix} = \frac{1}{h^2} \underbrace{\begin{bmatrix} -2 & 2 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 & 1 \end{bmatrix}}_{\delta_x.u_0 = \delta_x.u_N = 0} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix} \quad (5.16)$$

As is natural, the boundary conditions affect only those matrix entries in the extreme rows (and, in more involved settings, columns). See Problem 5.5, and Programming Exercise 5.2. For difference operators defined over uniform grids, the bulk of the matrix representation is of sparse Toeplitz form, where the number of non-zero bands reflects the order of the difference operator, and also, often, its accuracy.

5.2.9 Inner Products and Identities

As one might gather, an essential step towards energy-based analysis of finite difference schemes for distributed problems is the introduction of a spatial inner product between two grid functions, which is analogous to the continuous definition given in (5.5). In the discrete setting, there are many ways of doing this, but the simplest is the following: an l_2 spatial inner product of two one-dimensional grid functions, f_l^n and g_l^n , over the interval $l \in \mathcal{D}$, may be defined as

$$\langle f^n, g^n \rangle_{\mathcal{D}} = \sum_{l \in \mathcal{D}} h f_l^n g_l^n \quad (5.17)$$

In fact, this is none other than a rudimentary Riemann sum approximation to the continuous L_2 inner product. The inner product is a scalar time series, dependent on n —note that for grid functions which appear inside an inner product, the spatial index l is suppressed. An l_2 norm follows as

$$\|f^n\|_{\mathcal{D}} = \sqrt{\langle f^n, f^n \rangle_{\mathcal{D}}} \geq 0 \quad (5.18)$$

The Cauchy-Schwartz and triangle inequalities extend directly to such discrete inner products as

$$|\langle f^n, g^n \rangle_{\mathcal{D}}| \leq \|f^n\|_{\mathcal{D}} \|g^n\|_{\mathcal{D}} \quad (5.19a)$$

$$\|f^n + g^n\|_{\mathcal{D}} \leq \|f^n\|_{\mathcal{D}} + \|g^n\|_{\mathcal{D}} \quad (5.19b)$$

It is sometimes useful [113] to define other types of inner products, which vary slightly from the above definition in (5.17); these usually are distinct at the endpoints of the spatial interval over which the inner product is defined. Assuming that the endpoints of the interval \mathcal{D} are given by d_- and d_+ , one of particular interest is given by

$$\langle f^n, g^n \rangle'_{\mathcal{D}} = \sum_{l=d_-+1}^{d_+-1} h f_l^n g_l^n + \frac{h}{2} f_{d_-}^n g_{d_-}^n + \frac{h}{2} f_{d_+}^n g_{d_+}^n \quad (5.20)$$

A norm can again be defined in terms of this inner product, as per (5.18), and the Cauchy-Schwartz and triangle inequalities, from (5.19) above also hold. This inner product may be interpreted as an approximation to the continuous inner product using trapezoids, rather than a simple Riemann sum, hence the factor of $1/2$ which scales the values at the endpoints. As might be imagined, the use of such an inner product leads to variations in the way numerical boundary conditions are posed and implemented. Other, even more general inner products are available—see Problem 5.7. Some more comments on this appear in the next section.

Before looking at some of the manipulations which may be employed in the case of spatial difference operators, it is important to note that the identities involving only time difference or averaging operators applied to products of time series in the lumped context in §2.4.2, extend directly to the case of grid functions. For instance, identity (2.22a) for time series, may be generalized, for a grid function u , as

$$\langle \delta_t u, \delta_{tt} u \rangle_{\mathcal{D}} = \delta_{t+} \left(\frac{1}{2} \|\delta_{t-} u\|_{\mathcal{D}}^2 \right) \quad (5.21)$$

(Notice that in this case, the time difference δ_{t+} operates on a scalar time series, and not a grid function!) Such manipulations are of great use in the analysis in the following chapters, and generalizations of the identities (2.22) are readily proved—see Problem 5.6.

5.2.10 Summation by Parts

Of extreme utility in energy analysis are various manipulations which correspond to integration by parts. For example, consider the inner product $\langle f, \delta_{x+}g \rangle_{\mathcal{D}}$, where, for brevity, the time index n has been dropped from the grid functions. Assuming again that the endpoints of the interval \mathcal{D} are given by d_- and d_+ , an inner product representation of summation by parts may be derived as follows:

$$\begin{aligned} \langle f, \delta_{x+}g \rangle_{\mathcal{D}} &= \sum_{l=d_-}^{d_+} hf_l \frac{1}{h} (g_{l+1} - g_l) = - \sum_{l=d_-}^{d_+} h \frac{1}{h} (f_l - f_{l-1}) g_l + f_{d_+} g_{d_++1} - f_{d_-} g_{d_-} \quad (5.22) \\ &= -\langle \delta_{x-}f, g \rangle_{\mathcal{D}} + f_{d_+} g_{d_++1} - f_{d_-} g_{d_-} \end{aligned}$$

Notice in particular that the boundary terms which appear involve evaluations of the grid functions at points $d_+ + 1$ and $d_- - 1$, which are outside the domain of definition of the problem itself, at least when d_+ and d_- are finite. These are again instances of “imaginary grid points,” as discussed in §5.2.8. As was noted earlier, the values of grid functions at such imaginary locations will always be set, through boundary conditions, in terms of values of the grid function over the domain interior. As such, in an implementation, it is unnecessary to store such values, though in some cases, the explicit representation of such values can lead to algorithmic simplifications. Notice also that in the boundary terms, the grid functions f and g are evaluated at different (adjacent) locations. As in the continuous case, if $d_- = -\infty$ or $d_+ = \infty$, then the values of grid functions at these locations are assumed to vanish. For example, if $\mathcal{D} = \mathbb{Z}$, one has, simply, $\langle f, \delta_{x+}g \rangle_{\mathbb{Z}} = -\langle \delta_{x-}f, g \rangle_{\mathbb{Z}}$.

It is sometimes of use to rewrite identities such as summation by parts in terms of slightly different domains; this is a point of contrast with such identities in the continuous case. For example, for a given domain $\mathcal{D} = [d_-, \dots, d_+]$, one may define related domains lacking one of the boundary points, such as $\underline{\mathcal{D}} = [d_-, \dots, d_+ - 1]$, and $\overline{\mathcal{D}} = [d_- + 1, \dots, d_+]$. One variant of summation by parts then looks like:

$$\langle f, \delta_{x+}g \rangle_{\underline{\mathcal{D}}} = -\langle \delta_{x-}f, g \rangle_{\overline{\mathcal{D}}} + f_{d_+} g_{d_+} - f_{d_-} g_{d_-} \quad (5.23)$$

In this case, the boundary terms involve evaluations of the grid functions f and g at the same point.

It is of course possible to extend summation by parts to cover double spatial differences. Using the operator $\delta_{xx} = \delta_{x+}\delta_{x-}$, and summation by parts (5.22) twice, one has the following identities:

$$\langle f, \delta_{xx}g \rangle_{\mathcal{D}} = \langle \delta_{xx}f, g \rangle_{\mathcal{D}} - f_{d_-} \delta_{x-} g_{d_-} + g_{d_-} \delta_{x-} f_{d_-} + f_{d_+} \delta_{x+} g_{d_+} - g_{d_+} \delta_{x+} f_{d_+} \quad (5.24)$$

$$\langle f, \delta_{xx}g \rangle_{\mathcal{D}} = \langle \delta_{xx}f, g \rangle_{\underline{\mathcal{D}}} - f_{d_-} \delta_{x-} g_{d_-} + g_{d_-} \delta_{x+} f_{d_-} + f_{d_+} \delta_{x+} g_{d_+} - g_{d_+} \delta_{x-} f_{d_+} \quad (5.25)$$

Again, if $\mathcal{D} = \mathbb{Z}$, the above identities simplify to

$$\langle f, \delta_{xx}g \rangle_{\mathbb{Z}} = \langle \delta_{xx}f, g \rangle_{\mathbb{Z}} \quad (5.26)$$

For the identities (5.24) and (5.25), notice that the boundary terms are not centered about the endpoints; this may be remedied with recourse to the primed inner product defined in (5.23) above, and can be useful in developing numerical boundary conditions which possess a higher degree of accuracy. See Problem 5.8.

5.2.11 Some Bounds

It is also helpful to be able to relate norms of grid functions under spatial difference operations to norms of the grid functions themselves. For instance, consider the grid function $\delta_{x+}u$, defined over

domain \mathcal{D} . One may write

$$\begin{aligned} \|\delta_{x+}u\|_{\underline{\mathcal{D}}}^2 &= \sum_{l=d_-}^{d_+-1} h(\delta_{x+}u_l)^2 = \sum_{l=d_-}^{d_+-1} \frac{1}{h}(u_{l+1} - u_l)^2 = \sum_{l=d_-}^{d_+-1} \frac{1}{h}(u_{l+1}^2 + u_l^2 - 2u_{l+1}u_l) \\ &\leq \sum_{l=d_-}^{d_+-1} \frac{2}{h}(u_{l+1}^2 + u_l^2) \end{aligned}$$

or

$$\|\delta_{x+}u\|_{\underline{\mathcal{D}}} \leq \frac{2}{h}\|u\|'_{\mathcal{D}} \leq \frac{2}{h}\|u\|_{\mathcal{D}} \quad (5.27)$$

Thus the norm of a grid function under a spatial difference is bounded from above by the norm of the grid function itself, times a factor of $2/h$. Notice that the bound is slightly tighter when the primed norm is used. Similarly, one has, for the grid function $\delta_{x-}u$,

$$\|\delta_{x-}u\|_{\overline{\mathcal{D}}} \leq \frac{2}{h}\|u\|'_{\mathcal{D}} \leq \frac{2}{h}\|u\|_{\mathcal{D}}$$

Bounds on higher differences of a grid function may be obtained through repeated use of the above bounds. For instance, for a grid function $\delta_{xx}u$, one has

$$\|\delta_{xx}u\|_{\underline{\mathcal{D}}} = \|\delta_{x+}\delta_{x-}u\|_{\underline{\mathcal{D}}} \leq \frac{2}{h}\|\delta_{x-}u\|_{\overline{\mathcal{D}}} \leq \frac{4}{h^2}\|u\|_{\mathcal{D}} \quad (5.28)$$

It is possible to relate the bounds presented here to the Fourier representation—see Problems 5.1 and 5.9.

5.2.12 Interpolation and Spreading Revisited

For a given system requiring both interpolation and spreading at the same location, it is useful to note the following: Supposing a given q th order spreading function $J_{l,q}(x_o)$ operating at location x_o , and an interpolation function $I_q(x_o)$, it is true that for any grid function f ,

$$\langle f, J_q(x_o) \rangle_{\mathcal{D}} = I_q(x_o)f \quad (5.29)$$

Note that this is only true when the interpolation and spreading functions are of the same order. This fact is important in proving stability for certain point-excited systems such as, e.g., the bow-string interaction—see §7.4.

Spreading of an input time series onto a grid is also associated with a slight change in gain. This gain for a spreading operator $J(x_i)$, for a given fractional remainder α may be written as

$$\text{Gain}(J) = \sqrt{h}\|J\|_{\mathcal{D}}$$

and will depend on the fractional part α of the truncation to an integer grid point. For polynomial interpolants J_p of the type discussed in §5.2.4, one may show the following bound, which holds for all α :

$$\text{Gain}(J_p) \leq 1 \quad (5.30)$$

See Problem 5.10.

5.3 Coordinate Changes

In some problems, particularly those involving spatial variation of the vibrating material (such as strings of variable density, or bars of variable cross-section—see §7.9), it can be useful to introduce changes of spatial coordinate which match, in some sense, the variation in the medium itself. In 1D, this is quite straightforward. For a given spatial coordinate x , define a new coordinate $\alpha = \alpha(x)$. In

general, the map $x \rightarrow \alpha(x)$ should be smooth, one-to-one, and more generally preserve the ordering of the points in the medium, i.e., if $x_1 \leq x_2$, then $\alpha(x_1) \leq \alpha(x_2)$.

Partial derivatives may thus be transformed under such a mapping, i.e.,

$$\frac{\partial}{\partial x} \rightarrow \alpha' \frac{\partial}{\partial \alpha} \qquad \frac{\partial^2}{\partial x^2} \rightarrow \alpha' \frac{\partial}{\partial \alpha} \left(\alpha' \frac{\partial}{\partial \alpha} \right) \tag{5.31}$$

where now, the function $\alpha' = \frac{\partial \alpha}{\partial x}$ intervenes. A PDE in a spatial variable x may thus be transformed to an equivalent PDE in the new coordinate α .

The interest in performing such changes of variables at the level of the PDE is to avoid the hassle of variable grid spacing in the original coordinates—once the transformation has been carried out, the grid spacing may again be chosen uniform, and all the difference scheme analysis then applies as usual. The most commonly occurring differential operator, at least in musical acoustics problems, is the second derivative, $\partial^2/\partial x^2$. In transformed coordinates, a useful centered approximation to the second derivative is

$$\delta_{xx} \rightarrow \alpha' \delta_{\alpha+} ((\mu_{\alpha-} \alpha') \delta_{\alpha-}) \tag{5.32}$$

which is second order accurate. Note in particular that the continuous form of α' is assumed known. See Problem 5.11, and Programming Exercise 5.5.

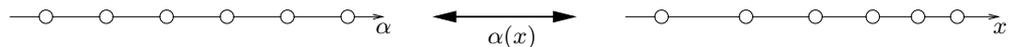


Figure 5.4: Coordinate change from coordinates x to coordinates α . A uniform grid in α coordinates maps to a variable grid in x coordinates.

Integration by parts and summation by parts still hold, as long as a weighting function $1/\alpha'$ is inserted into the formulae. For example, in the continuous case,

$$\left\langle \frac{1}{\alpha'} f, \alpha' (\alpha' g) \right\rangle_{\mathbb{R}} = -\langle f_{\alpha}, \alpha' g_{\alpha} \rangle_{\mathbb{R}} = \langle \alpha' (\alpha' f)_{\alpha}, \frac{1}{\alpha'} g \rangle_{\mathbb{R}}$$

and in the discrete case,

$$\left\langle \frac{1}{\alpha'} f, \alpha' \delta_{\alpha+} ((\mu_{\alpha-} \alpha') (\delta_{\alpha-} g)) \right\rangle_{\mathbb{Z}} = -\langle \delta_{\alpha-} f, (\mu_{\alpha-} \alpha') (\delta_{\alpha-} g) \rangle_{\mathbb{Z}} = \langle \alpha' \delta_{\alpha+} ((\mu_{\alpha-} \alpha') (\delta_{\alpha-} f)), \frac{1}{\alpha'} g \rangle_{\mathbb{Z}}$$

5.4 Problems

Problem 5.1 From (5.13a), it is known that the application of the operator δ_{xx} to a grid function u_l corresponds, in the spatial frequency domain, to multiplication of the Fourier transform $\tilde{u}(\beta)$ by a factor $-4 \sin^2(\beta h/2)/h^2$. For what value of β is the magnitude of this factor a maximum, and what is the value of the factor? Sketch the grid function $u_l = e^{j l \beta h}$ for this value of β .

Problem 5.2 Consider the operator δ_{xxx} , as defined in (5.10b), applied to a grid function u_l , now defined over the semi-infinite domain, $\mathcal{D} = \mathbb{Z}^+ = [0, 1, \dots]$. At which grid points will the evaluation of this operator require access to grid points at imaginary locations (i.e., outside of \mathbb{U}_N)? Consider also the following three sets of boundary conditions,

$$u = \delta_x \cdot u = 0 \tag{5.33a}$$

$$u = \delta_{xx} u = 0 \tag{5.33b}$$

$$\delta_{xx} u = \delta_x \cdot \delta_{xx} u = 0 \tag{5.33c}$$

In each case, use the conditions at $l = 0$ to arrive at expressions for $\delta_{xxx}u_l$ over the entire domain \mathbb{Z}^+ , purely in terms of values of u over the domain interior. (In the setting of bar vibration, useful in modelling percussion instruments such as xylophones, these three types of conditions correspond to clamped, simply supported, and free conditions, respectively. See §7.1.2.)

Problem 5.3 Write the operator δ_{xxx} , as defined in §5.2.2 in matrix form over the infinite domain $\mathcal{D} = \mathbb{Z}$, similarly to those presented in §5.2.7.

Problem 5.4 Consider the difference operator δ_{xx} applied to the grid function u_l over \mathbb{U}_N , under the zero-derivative boundary approximations $\delta_{x-}u_0 = \delta_{x+}u_N = 0$, and express $\delta_{xx}u_0$ and $\delta_{xx}u_N$ purely in terms of values over the domain \mathbb{U}_N (i.e., without using values at imaginary points). Express the difference operator in matrix form, similar to that given at the end of §5.2.8.

Problem 5.5 Continuing from Problem 5.2, with the operator δ_{xxx} applied to the grid function u_l over \mathbb{Z}^+ , construct the semi-infinite matrix form of the operator under the three sets of boundary conditions described. Under the conditions described in (5.33a) and (5.33b), you may omit the evaluation at the endpoint $l = 0$. It is helpful to use the infinite form of the matrix operator you constructed in Problem 5.3 as a starting point.

Problem 5.6 For each of the identities involving products of time series under the action of time difference operators, from (2.22), show that there is a corresponding identity for the case of a grid function $u = u_l^n$, defined over the domain \mathcal{D} . (As a starting point, you may wish to compare identity (2.22a) with its corresponding distributed form (5.21).)

Problem 5.7 Consider the discrete inner product defined, for simplicity, over $\mathcal{D} = \mathbb{Z}^+$ as

$$\langle f, g \rangle_{\mathbb{Z}^+}^{\epsilon} = \sum_{l=1}^{\infty} h f_l g_l + \frac{\epsilon}{2} h f_0 g_0$$

with respect to a free parameter $\epsilon > 0$. Show that the following summation by parts identity holds:

$$\langle f, \delta_{x+}g \rangle_{\mathbb{Z}^+}^{\epsilon} = \langle \delta_{x-}f, g \rangle_{\mathbb{Z}^+}^{\epsilon} - f_0 ((2 - \epsilon)\mu_{x+} + (\epsilon - 1))g_0$$

Show also that the bound

$$\|\delta_{x+}u\|_{\mathbb{Z}^+} \leq \frac{2}{h} \|u\|_{\mathbb{Z}^+}^{\epsilon}$$

holds in general only for values of ϵ such that $\epsilon \geq 1/2$.

For an interesting application of this definition of an inner product to the analysis of numerical boundary conditions, see Problem 6.5 in the next chapter.

Problem 5.8 Demonstrate, using the primed l_2 inner product defined in (5.23), over a domain \mathcal{D} , the summation by parts identity

$$\langle f, \delta_{xx}g \rangle'_{\mathcal{D}} = \langle \delta_{xx}f, g \rangle'_{\mathcal{D}} - f_{d-} \delta_x g_{d-} + g_{d-} \delta_x f_{d-} + f_{d+} \delta_x g_{d+} - g_{d+} \delta_x f_{d+}$$

Thus the boundary terms which appear involve difference approximations which are centered about the endpoints of the domain. This is in contrast with the corresponding identity (5.24) in the case of the standard unprimed inner product.

Problem 5.9 Find a grid function u_l defined over $\mathcal{D} = \mathbb{U}_N$ such that the bound (5.28) is satisfied with equality. Your answer from Problem 5.1 may be of help here.

Problem 5.10 Prove bound (5.30) for the spreading functions J_0 , J_1 and J_3 , and also find the minimum value of the gain of each of the above operators, for any possible value of α . (In the case of J_3 , a numerical proof will suffice.)

Problem 5.11 Consider the second derivative under a coordinate transformation $\alpha(x)$, as given in (5.31). If $\alpha(x) = x^2$, rewrite the expression for the second derivative purely in terms of α (i.e., without any occurrence of x , which appears, e.g., in $\alpha' = d\alpha/dx$).

5.5 Programming Exercises

Exercise 5.1 For a input vector of length $N + 1$ values, (corresponding, ultimately, to values of a function $u(x)$ at locations $x = 0, 1/N, \dots, 1$), create a matlab script which interpolates a value $u(x_o)$, for some $0 \leq x_o \leq 1$. The order of interpolation should be adjustable, through an input parameter, such that it performs zeroth, first and third order interpolation, as discussed in §5.2.4. In the case of third order interpolation, you may revert to first order interpolation when the observation point x_o lies within a single grid point of either end of the domain.

Exercise 5.2 Create a Matlab script which generates the matrix operator \mathbf{D}_{xx} , corresponding to the difference operator δ_{xx} , operating over the unit interval $\mathbb{U}_N = [0, 1, \dots, N]$. Your code should operate under a combination of choices of numerical boundary conditions at either end of the domain, namely fixed (Dirichlet) or free (Neumann). You may use the fixed condition $u = 0$, and the centered numerical Neumann boundary condition $\delta_x u = 0$. Make sure that your script generates sparse matrices (learn about the matlab function `sparse` for this purpose), and that it generates matrices of the appropriate size—for example, under Neumann conditions at both ends, \mathbf{D}_{xx} will be an $(N + 1) \times (N + 1)$ matrix, but for Dirichlet conditions, it will be $(N - 1) \times (N - 1)$.

Exercise 5.3 Create a Matlab script which generates the matrix operator \mathbf{D}_{xxx} , corresponding to the difference operator δ_{xxx} , operating over the unit interval $\mathbb{U}_N = [0, 1, \dots, N]$. Your code should be capable of setting any of the conditions (5.33) separately at each end of the boundary (i.e., at $l = 0$ or $l = N$), giving a total of nine possible outputs. Make sure that your script generates sparse matrices (learn about the matlab function `sparse` for this purpose), and that it generates matrices of the appropriate size—for example, under conditions (5.33a) or (5.33b) at each end of the domain, \mathbf{D}_{xxx} will be an $(N + 1) \times (N + 1)$ matrix, but for conditions (5.33c) at both ends, it will be $(N - 1) \times (N - 1)$.

Exercise 5.4 Using the Matlab script you created above which generates the matrix \mathbf{D}_{xx} , solve the equation $\mathbf{D}_{xx}\mathbf{x} = \mathbf{b}$, where \mathbf{b} is a known vector, and where \mathbf{x} is the unknown, in two different ways:

- 1) Explicitly compute the inverse \mathbf{D}_{xx}^{-1} , and solve for \mathbf{x} as $\mathbf{x} = \mathbf{D}_{xx}^{-1}\mathbf{b}$.
- 2) Use Matlab's standard iterative method to solve the equation. You may perform this calculation easily, as

```
x=Dxx\b
```

Using the `tic` and `toc` commands, compare the performance of these two methods for a variety of sizes of the vectors \mathbf{x} and \mathbf{b} . You may use the form of \mathbf{D}_{xx} under fixed boundary conditions, and you may also wish to choose \mathbf{b} of the form of a sinusoid or polynomial function, so that you can be sure that your calculated results are correct (\mathbf{b} should be approximately the second derivative of \mathbf{x} !).

Exercise 5.5 For a given function $u(x)$ defined over $\mathcal{D} = \mathbb{U}$, calculate the second derivative numerically in two different ways:

- 1) Directly, using the operator δ_{xx} , and $N + 1$ samples of the function $u(x)$ at locations $x = l/N$, for some integer N (the grid size).
- 2) Under the coordinate transformation $\alpha(x) = x^2$. In this case, you will make use of samples of u which are equally spaced in the coordinate α —take $N + 1$ of them as well. Use the transformed difference operator in α given in (5.32). Be sure that you have written α' explicitly in terms of α , as per Problem 5.11.

In both cases, you need only calculate approximations to the second derivative at interior points in the domain. Plot both results together on the same set of axes. You may wish to choose your function u of the form of a sinusoid or polynomial function, so that you can be sure that your calculated results are correct.

Chapter 6

The 1D Wave Equation

The one-dimensional wave equation is, arguably, the single most important partial differential equation in musical acoustics, if not in physics as a whole. Though, strictly speaking, it is useful only as a test problem, variants of it serve to describe the behaviour of strings, both linear and nonlinear, as well as the motion of air in an enclosed acoustic tube, among many other systems. It is well worth spending a good deal of time examining this equation as well as various means of arriving at a numerical solution, because the distinctions between well-known synthesis techniques such as lumped models, modal synthesis, digital waveguides and classical time-domain methods such as finite differences appear in sharp contrast.

It is important to point out, however, that in some respects, the 1D wave equation is a little too special to serve as a useful model problem. Though, as mentioned above, it is frequently used as a test problem for numerical methods, it is the rare example of a PDE which admits an exact numerical solution—furthermore, the solution may be computed in a very efficient manner. Though digital waveguide methods are built directly around such efficient numerical solutions, one must beware of the temptation to generalize such techniques to other problems. As a result, digital waveguides will be discussed here, but the emphasis will be on those aspects of numerical techniques which do not rely on special behaviour of a particular equation.

In §6.1, the one-dimensional wave equation is defined, in the first instance, for simplicity, over the entire real line. Frequency domain analysis is introduced, followed by a discussion of phase and group velocity. The Hamiltonian formulation for the wave equation appears next, and is used in order to motivate a discussion of bounds on solution growth, as well as appropriate boundary conditions. Boundary conditions, coupled with frequency domain methods lead to a representation of the solution in terms of modes. The section is concluded by a brief look at traveling wave solutions. A simple finite difference scheme is presented in §6.2, followed by frequency domain (or von Neumann) analysis, yielding a simple (Courant-Friedrichs-Lewy) stability condition, and information regarding numerical dispersion, as well as its perceptual significance in sound synthesis. A look at the matrix form of the finite difference scheme follows, as well as the very important special case of the discrete traveling wave solution. Other varieties of finite difference schemes appear in §6.3, and a treatment of loss appears in §6.5. Finally, in §6.6, various synthesis methods, specifically modal synthesis, lumped networks, and digital waveguides are compared in this special case of the wave equation.

References for this chapter include: [244, 265, 242, 258, 153]

6.1 Definition and Properties

The one-dimensional wave equation is defined as

$$u_{tt} = c^2 u_{xx} \quad (6.1)$$

It is a second-order partial differential equation in the dependent variable $u = u(x, t)$, where x is a variable representing distance, and t , as before, is time. The equation is defined over $t \geq 0$, and for $x \in \mathcal{D}$, where \mathcal{D} is some simply connected subset of \mathbb{R} . c is a constant often referred to as the wave speed. See §5.1 for a description of the notation used here in representing partial differential operators.

As mentioned above, the 1D wave equation appears in a range of applications across all of physics, and occupies a privileged position in musical acoustics (see Figure 6.1). It is a simple first approximation, under low amplitude conditions, to the transverse motion of strings, in which case $c = \sqrt{T_0/\rho A}$, where T_0 is the applied string tension, ρ is the material density, and A the string cross-sectional area. It is also an approximation to the longitudinal motion of a uniform bar [174], with $c = \sqrt{E/\rho}$ and where E and ρ are Young's modulus, and again material density for the bar under consideration; longitudinal motion of a bar is of somewhat less interest in sound synthesis applications compared with transverse motion, which will be taken up in detail in §7.1. In the context of wind instrument modeling, it also describes the longitudinal vibration of an air column in a tube of uniform cross-section, with $c = \sqrt{K/\rho}$, and where K and ρ are the bulk modulus and material density of the air in the tube. Finally, it is also worth pointing out that the wave equation also applies to the case of the lossless electrical transmission line [55], where $c = \sqrt{1/LC}$, with L and C the inductance and capacitance per unit length of the line. The electrical transmission line, though not of intrinsic musical interest, has served as a conceptual basis for the construction of scattering based numerical methods (digital waveguides and wave digital filters among them [24, 241]). Under musical conditions, the wave equation is rather a better approximation in the case of the acoustic tube than that of the string; string nonlinearities which lead to perceptually important effects will be dealt with in detail in Chapter 8.

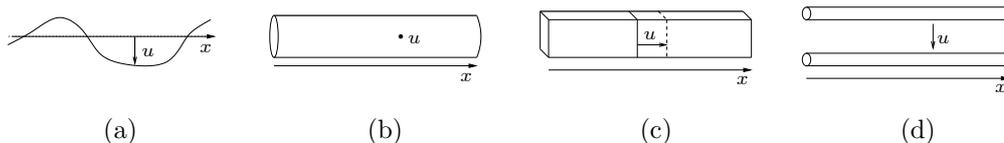


Figure 6.1: *Different physical systems whose dynamics are described by the wave equation (6.1). $u(x, t)$ may represent: (a) transverse displacement of a lossless string vibrating at low amplitude, (b) deviation in pressure about a mean in an acoustic tube, under lossless conditions, (c) deviation of a point from its position at rest for a bar vibrating longitudinally, or (d) voltage across a lossless transmission line.*

The wave equation has been derived from first principles in countless works; perhaps the best treatment in the present context of acoustics appears in the classic texts of Morse [173] and Morse and Ingard [174]. Though such a derivation could well be skipped here, one particular form, based on the limiting behaviour of an array of masses and springs, is of particular relevance in digital sound synthesis.

6.1.1 Linear Array of Masses and Springs

Consider an equally-spaced array of masses M , of spacing h , connected by linear springs of spring constant K , and of equilibrium length h . Suppose that the masses are constrained to move only horizontally, and that the deviation of the l th mass from its rest position is written as $u_l(t)$. See Figure 6.2. The equation of motion of the l th mass will then be

$$M \frac{d^2 u_l}{dt^2} = f_{l+\frac{1}{2}} - f_{l-\frac{1}{2}} \quad (6.2)$$

where here, $f_{l+\frac{1}{2}}(t)$ represents the force exerted by the spring connecting the l th mass to the $(l+1)$ th, where a positive force is assumed to act to the right. Under linear spring behaviour, an expression for the force will be, in terms of the displacements u_i themselves,

$$f_{i+\frac{1}{2}} = K (u_{i+1} - u_i) \quad (6.3)$$

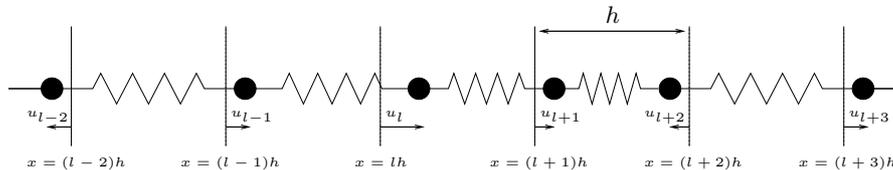


Figure 6.2: *Linear array of masses and springs, constrained to move longitudinally. The displacement of the l th mass, located at equilibrium at spatial position $x = lh$, is written as u_l .*

The defining equations (6.2) and (6.3) may be combined into a single equation as

$$M \frac{d^2 u_l}{dt^2} = K (u_{l+1} - 2u_l + u_{l-1}) \quad (6.4)$$

Now, defining $M = \rho Ah$, where ρ is a linear mass density and A is a constant with dimensions of area, and $K = EA/h$, where E is a stiffness parameter (actually, Young's modulus), the above may be written as

$$\frac{d^2 u_l}{dt^2} = \frac{E}{\rho} \left(\frac{u_{l+1} - 2u_l + u_{l-1}}{h^2} \right) \quad (6.5)$$

In the limit as h becomes small, it is not difficult to see that, if the displacements $u_l(t)$ are viewed as displacements of a continuous medium $u(x, t)$ at locations $x = lh$, then the right side of the above equation is no more than a simple centered finite difference approximation to a second spatial derivative (see §5.2.2). The above equation then approaches the 1D wave equation (6.1), with $c = \sqrt{E/\rho}$, modelling, in this case, longitudinal vibrations of a stiff system, such as a bar.

Though the above is a method of deriving the wave equation from a linear array of masses and springs, it may be viewed, before the limit of small h is taken, in two separate ways, corresponding to separate philosophies of sound synthesis. The lumped network approach to sound synthesis, mentioned briefly in §1.2.1, would take a system of ODEs, defined by an equation such as (6.5) (or, more properly, the combined system of (6.2) and (6.3)) as the starting point for developing a numerical method. One can easily apply time integration methods such as those outlined in Chapter 3. For a distributed system, such as that described by the wave equation, another way of proceeding is to simply deal with the distributed system directly, thus bypassing any notion of a network of elements¹. This is the point of view taken in this book. In the case presented here, as it turns out,

¹Indeed, for a distributed system, the entire apparatus of network theory is elegantly wrapped up in the PDE

the two approaches lead to very similar results. Indeed, if the second time derivative in (6.5) is approximated by a second time difference, a standard finite difference scheme for the wave equation results—see §6.2. For more complex systems, however, the correspondence is less direct. See §16.2.3 for some general comments on this subject.

6.1.2 Non-dimensionalized Form

Non-dimensionalization is an extremely useful means of simplifying physical systems, especially in preparation for the application of numerical simulation techniques. As a first step, one may introduce the dimensionless coordinate $x' = x/L$, for some constant L with dimensions of length. In the general case of the wave equation defined over an arbitrary spatial domain \mathcal{D} , this constant is arbitrary, but over a finite spatial domain, such as $x \in [0, x_0]$, it may be taken to be the length $L = x_0$, thus yielding a system defined over the domain $x' \in [0, 1]$. The wave equation (6.1) then becomes, in primed coordinates,

$$u_{tt} = \gamma^2 u_{x'x'} \quad (6.6)$$

where $\gamma = c/L$ is a constant with dimensions of frequency. Such a description is useful in relating the wave equation to the simple harmonic oscillator, as described in Chapter 3. The term “wave speed” will still be used when referring to the constant γ .

This is a good example of a reduction in the number of parameters necessary to specify a system, which is of particular importance in physical modeling sound synthesis—it is redundant to specify both the wave speed c and the length L , and the single parameter γ suffices. The fewer in number the parameters, the easier it will be for the musician working with a synthesis algorithm to navigate the space of possible timbres². In general, forms in this book will be presented, whenever possible, in such a spatially-nondimensionalized form, and the primed notation will be suppressed.

In the standard numerical analysis literature, it is commonplace to see a further temporal nondimensionalization, through the introduction of a temporal variable $t' = t/T$, for a characteristic time constant T . A judicious choice of $T = L/c$ leads to the form

$$u_{t't'} = u_{x'x'} \quad (6.7)$$

Though this form is apparently simpler than (6.6) above, in practice, especially when programming synthesis routines, there is no real advantage to such a further step (i.e., an extra parameter, namely the time step, will be re-introduced during the discretization procedure). In addition, the crucial frequency domain behaviour of such a fully nondimensionalized system is slightly obscured through the introduction of such temporal scaling. For this perceptual reason, namely that we are only interested in hearing the outputs of the resulting simulations, which themselves are time-domain waveforms, only “half-way” spatial nondimensionalization will be employed in this book.

6.1.3 Initial Conditions

The 1D wave equation is a second order (in time) partial differential equation, and just as for the SHO, requires the specification of two initial conditions. Normally these are the values of the variable u and its time derivative, at time $t = 0$, i.e.,

$$u(x, 0) = u_0(x) \quad u_t(x, 0) = v_0(x) \quad (6.8)$$

itself, and need never be explicitly employed!

²This replacement of physical parameters by a smaller set of perceptual parameters in some ways goes against the spirit of physical modeling synthesis, where one might want access to a full set of physical parameters. Notice, however, that in this case, the physics itself remains unchanged under such data reduction.

which are now functions of x . Other equivalent choices (such as wave variables which are of use in a digital waveguide implementation) are possible.

In some physical modeling synthesis applications, such as the modeling of struck or plucked strings, it is convenient in the first instance to initialize the model itself using the above conditions. For a strike, for instance, one could choose a function v_0 , perhaps peaked at a desired location corresponding to the strike center, and set u_0 to zero. For a pluck, one could make a similar choice for u_0 , and set v_0 to zero. One commonly-employed choice of initial displacement distribution, in the case of the string, is the triangular function, defined over the unit interval $\mathcal{D} = \mathbb{U}$ by

$$c_{tri}(x) = \begin{cases} \frac{c_0}{x_0}x, & 0 \leq x \leq x_0 \\ \frac{c_0}{x_0-1}(x-1), & x_0 \leq x \leq 1 \end{cases} \quad (6.9)$$

where c_0 and x_0 are the peak displacement and its location. For a strike, often localized to a very small region in the domain, one often uses a simple delta function distribution, i.e., one sets $u_t(x, 0) = c_0\delta(x - x_0)$, where c_0 and x_0 represent the amplitude and position of the strike.

One particular artificial (but physically reasonable) choice of initial condition profile, characterized by a small number of parameters, is the raised cosine distribution, defined by

$$c_{rc}(x) = \begin{cases} \frac{c_0}{2}(1 + \cos(\pi(x - x_0)/x_{hw})), & |x - x_0| \leq x_{hw} \\ 0, & |x - x_0| > x_{hw} \end{cases} \quad (6.10)$$

Here, c_0 is the peak amplitude, x_0 the spatial center, and x_{hw} the half-width length of the distribution. See Figure 6.3. Such a function may be used in order to set either u_0 or v_0 , and, depending on the choices of parameters, it may (very roughly) approximate the triangular and delta distributions mentioned above. It may also be extended in a natural way to two dimensions—see §11.1.

In a more complete physical modeling framework, however, generally such interactions are modeled not through initialization, but through applied forces, perhaps from a hammer or hand model; many examples of such interactions will be discussed in this book.

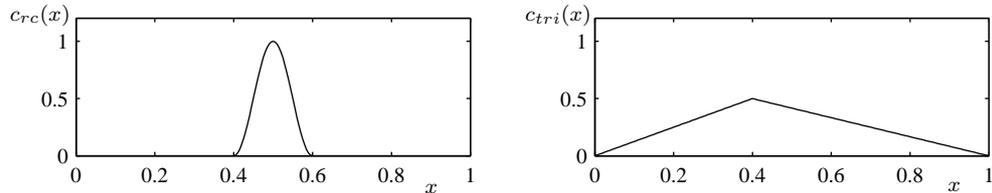


Figure 6.3: *Different examples of a initialization functions. Left, $c_{rc}(x)$ as defined in (6.9), with $c_0 = 1$, $x_0 = 0.5$, and $x_{hw} = 0.1$. Right, $c_{tri}(x)$ as defined in (6.10) with $c_0 = 0.5$, and $x_0 = 0.4$.*

6.1.4 Strikes and Plucks: Time Evolution of Solution

Leaving aside the issue of boundary conditions, the wave equation (6.6), when supplemented by initial conditions, is complete. This is a good chance to look at some of behaviour exhibited under different types of initial excitation, in particular pluck-like excitations, in which case the initial position is set to a non-zero distribution, and the initial velocity to zero, and struck excitations, which are just the opposite. In Figure 6.4, the time evolution of the solution is shown at successive instants under both types of conditions, using the initializing function $c_{rc}(x)$ given in (6.10); the

triangular initialization function will be employed slightly later, once boundary conditions have been discussed.

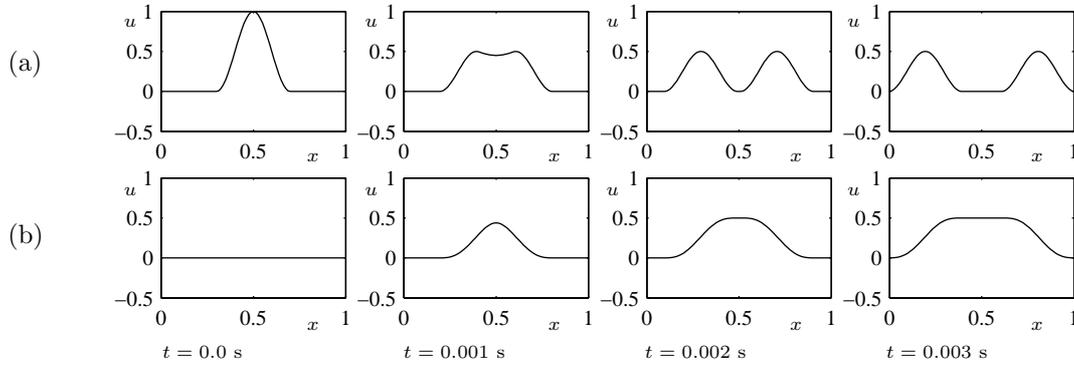


Figure 6.4: Time evolution of the solution to the wave equation (6.6), with $\gamma = 100$: (a), under a plucked initial condition, with $u_0 = c_{rc}(x)$, and $x_0 = 0.5$, $c_0 = 1$ and $x_{hw} = 0.2$. (b), under a struck initial condition of $v_0 = c_{rc}(x)$, with $x_0 = 0.5$, $c_0 = 500$, and $x_{hw} = 0.2$. Successive snapshots of the solution profile are shown at the times indicated, over the interval $x \in [0, 1]$.

6.1.5 Dispersion Relation

Frequency domain analysis is extremely revealing in the case of the wave equation in particular. Considering the case of a string defined over the spatial domain $\mathcal{D} = \mathbb{R}$, a Fourier transform (in space) and a two-sided Laplace transform (in time) may be used. See Chapter 5 for more details. As a short-cut, one may analyze the behaviour of the test solution

$$u(x, t) = e^{st+j\beta x} \quad (6.11)$$

where s is interpreted as a complex frequency variable, $s = \sigma + j\omega$, and β is a real wavenumber. The resulting characteristic equation (see §5.1.2) is

$$s^2 + \gamma^2 \beta^2 = 0 \quad (6.12)$$

which has roots

$$s_{\pm}(\beta) = \pm j\gamma\beta \quad \implies \quad \sigma = 0, \quad \omega(\beta) = \pm\gamma\beta \quad (6.13)$$

This is the dispersion relation for the system (6.6); the frequency and wavenumber of a plane-wave solution are not independent. The fact that there are two solutions results from the fact that the wave equation is second order in time—the solutions correspond to left-going and right-going wave solutions.

6.1.6 Phase and Group Velocity

In the case of the wave equation above, the phase velocity and group velocity are both constant, i.e.,

$$v_{\phi} = v_g = \gamma \quad (\text{1D wave equation}) \quad (6.14)$$

See §5.1.2. In particular, the phase and group velocities are independent of frequency ω . Thus any wave-like solution to the 1D wave equation travels at a constant speed. One might surmise, using arguments from Fourier theory, that if all components of a solution travel at the same speed, then

any possible solution must travel with this speed—this is, in fact, true, and is another means of arriving at the so-called traveling wave solution to the wave equation, to be discussed next.

6.1.7 Travelling Wave Solutions

As is well-known, the 1D wave equation possesses travelling-wave solutions [174] of the form

$$u(x, t) = u^{(+)}(x - \gamma t) + u^{(-)}(x + \gamma t) \quad (6.15)$$

for arbitrary³ functions $u^{(+)}$ and $u^{(-)}$, which represent rightward and leftward travelling waves of spatially non-dimensionalized speed γ , respectively. This decomposition, due to d'Alembert, may be arrived at directly in the time-space domain, through a change of variables, or through frequency domain analysis. Such a simple decomposition is peculiar to the 1D wave equation alone, and does not extend in general to more complex variants of the 1D wave equation (with some exceptions) or to the case of general wave equation in multiple dimensions. It is, however, the starting point for digital waveguide modelling, which is perhaps the best known of all physical modelling sound synthesis techniques, and will be discussed in more detail in Section 6.2.11. See Figure 6.5 for a graphical representation of the traveling wave decomposition.

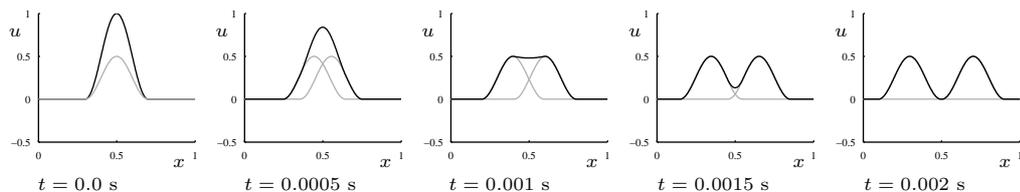


Figure 6.5: *Time evolution of the solution to the wave equation (6.6), with $\gamma = 100$, under a plucked initial condition, with $u_0 = c_{rc}(x)$, and $x_0 = 0.5$, $c_0 = 1$ and $x_{hw} = 0.2$. The solution is shown in black, and traveling wave components in grey. Successive snapshots of the solution profile are shown at the times indicated, over the interval $x \in [0, 1]$.*

Initialization is slightly more complex in the case of traveling wave components. The functions $u^{(+)}(x)$ and $u^{(-)}(x)$ may be related to the $u_0(x)$ and $v_0(x)$ by

$$u^{(+)}(x) = \frac{1}{2}u_0(x) - \frac{1}{2\gamma} \int^x v_0(x') dx' \quad u^{(-)}(x) = \frac{1}{2}u_0(x) + \frac{1}{2\gamma} \int^x v_0(x') dx' \quad (6.16)$$

The lower limit in the integration above has been left unspecified, but corresponds to the left end point of the domain \mathcal{D} over which the wave equation is defined. In digital waveguide methods, it is these variables which must be initialized, rather than physically observable quantities such as, e.g., displacement and velocity.

6.1.8 Energy Analysis

The frequency domain analysis above has been applied to the case of the wave equation defined over an infinite domain $\mathcal{D} = \mathbb{R}$, and thus boundary conditions have not been taken into account. Another way of examining the behaviour of the wave equation is similar to that which was discussed in Section 3.1.2, with regard to the oscillator, through the use of energetic techniques. Besides the

³Arbitrary, barring technical considerations having to do with continuity of the functions themselves, and their derivatives.

fact that it may be applied to systems which are not linear and shift invariant, energy analysis also provides extremely useful insights regarding correct boundary termination, as well as important bounds on solution growth.

In the first instance, consider again the wave equation defined over the entire real line, i.e., for $\mathcal{D} = \mathbb{R}$. Taking the inner product of (6.6) with u_t over \mathbb{R} (see §5.1.3 for the definition of the inner product and notational details) gives

$$\langle u_t, u_{tt} \rangle_{\mathbb{R}} = \gamma^2 \langle u_t, u_{xx} \rangle_{\mathbb{R}} \quad (6.17)$$

and, employing integration by parts, as per (5.9), one may arrive at

$$\langle u_t, u_{tt} \rangle_{\mathbb{R}} + \gamma^2 \langle u_{tx}, u_x \rangle_{\mathbb{R}} = 0 \quad (6.18)$$

Both of the terms in the above equation may be written as total derivatives with respect to time, i.e.,

$$\frac{d}{dt} \left(\frac{1}{2} \|u_t\|_{\mathbb{R}}^2 + \frac{\gamma^2}{2} \|u_x\|_{\mathbb{R}}^2 \right) = 0 \quad (6.19)$$

or, more simply,

$$\frac{d\mathfrak{H}}{dt} = 0 \quad \text{with} \quad \mathfrak{H} = \mathfrak{T} + \mathfrak{V} \quad \text{with} \quad \mathfrak{T} = \frac{1}{2} \|u_t\|_{\mathbb{R}}^2 \quad \mathfrak{V} = \frac{\gamma^2}{2} \|u_x\|_{\mathbb{R}}^2 \quad (6.20)$$

Here, \mathfrak{T} and \mathfrak{V} are, respectively, the kinetic and potential energy for the wave equation, and \mathfrak{H} is the total energy, or Hamiltonian. (As in the case of the SHO, these quantities must be scaled by a constant with dimensions of mass.) Equation (6.20), and the non-negativity of the terms \mathfrak{T} and \mathfrak{V} above imply that

$$\mathfrak{H}(t) = \mathfrak{H}(0) \geq 0 \quad (6.21)$$

Clearly, the non-negativity of \mathfrak{T} and \mathfrak{V} also imply that

$$\|u_t\|_{\mathbb{R}} \leq \sqrt{2\mathfrak{H}(0)} \quad \|u_x\|_{\mathbb{R}} \leq \frac{\sqrt{2\mathfrak{H}(0)}}{\gamma} \quad (6.22)$$

Once boundary conditions are taken into account, it may be possible to improve upon these bounds—see §6.1.10.

6.1.9 Boundary Conditions

The 1D wave equation involves second-order differentiation in space, and, as such, requires the specification of a single boundary condition at any endpoint of the spatial domain. A typical such condition often employed at an endpoint, such as $x = 0$, of the domain is the following:

$$u(0, t) = 0 \quad (6.23)$$

Such a condition is often referred to as being of Dirichlet type. If the 1D wave equation is intended to describe the displacement of a string, then this is clearly a fixed termination. If $u(x, t)$ represents the pressure variation in an acoustic tube, such a condition corresponds (roughly) to an open end of the tube.

Another commonly-encountered condition is the Neumann condition:

$$u_x(0, t) = 0 \quad (6.24)$$

Such a condition is easily interpreted in the context of the acoustic tube as corresponding to a closed tube end, and less easily in the case of string, where it may be viewed as describing a string endpoint which is free to move in a transverse direction, but not longitudinally (it is best to visualize the end of such a string as constrained to move along a vertical “guide rail”). The Dirichlet and Neumann conditions are often generalized to include a non-zero constant on the right-hand side [113].

Both conditions are lossless—in order to get a better idea of what this means (and to gain some insights into how such conditions may be arrived at), energy techniques are invaluable. Consider now the wave equation defined not over the infinite domain $\mathcal{D} = \mathbb{R}$, but over the semi-infinite domain⁴ $\mathcal{D} = \mathbb{R}^+ = [0, \infty]$. Taking an inner product of the wave equation with u_t , now over the semi-infinite domain gives (6.17) as before, but integration by parts yields, instead of (6.18),

$$\langle u_t, u_{tt} \rangle_{\mathbb{R}^+} + \gamma^2 \langle u_{tx}, u_x \rangle_{\mathbb{R}^+} = -\gamma^2 u_t(0, t) u_x(0, t) \triangleq \mathfrak{B} \quad (6.25)$$

An extra boundary term \mathfrak{B} has thus appeared, and one can thus not proceed in general to a statement of conservation of energy such as (6.20), but rather

$$\frac{d\mathfrak{H}}{dt} = \mathfrak{B} \quad (6.26)$$

where the definition of \mathfrak{H} and its constituent terms are taken over \mathbb{R}^+ rather than \mathbb{R} . Now, the lossless interpretation of the Dirichlet and Neumann conditions above should be clear; the boundary term vanishes in either case, and one again has exact energy conservation⁵, as per (6.21). (Note, however, that in the case of the Dirichlet condition, the true losslessness condition, from the above, is rather that u_t vanish at the boundary, which is in fact implied by the more realistic condition that u itself vanish.)

Also, it follows directly that when the spatial domain of the problem at hand is finite, e.g., if $\mathcal{D} = \mathbb{U} = [0, 1]$, as is the case in all musical systems of interest, the energy balance becomes

$$\frac{d\mathfrak{H}}{dt} = \mathfrak{B} \triangleq \gamma^2 (u_t(1, t) u_x(1, t) - u_t(0, t) u_x(0, t)) \quad (6.27)$$

The boundary conditions at both ends of the domain come into play.

Lossy Boundary Conditions

The Dirichlet and Neumann conditions are by no means the only possible boundary conditions for the wave equation; there is in fact an infinite number of possible terminations, some of which possess a physical interpretation. For example, a simple lossy condition is given by

$$u_t(0, t) = \alpha u_x(0, t) \quad (6.28)$$

for some constant $\alpha > 0$. In this case, one clearly has

$$\mathfrak{B} = -\alpha \gamma^2 (u_x(0, t))^2 \leq 0 \quad (6.29)$$

implying that the energy is in fact non-increasing. It becomes very simple, using energetic techniques, to categorize sources of loss according to whether they occur at the boundary, or over the problem interior. When translated to a numerical setting, it becomes similarly possible to isolate potential sources of numerical instability—some are of global type, occurring over the problem interior, but often instability originates at an improperly set numerical boundary condition. The condition (6.28) above, under a special choice of α , can lead to a perfectly absorbing (i.e., impedance matched) boundary condition—see Problem 6.2.

One can go much further along these lines, and analyze nonlinear boundary conditions as well. An example appears in Problem 6.3.

⁴One of the nice features of energy analysis is that it allows boundary conditions at different locations to be analyzed independently; this is a manifestly physical approach, in that one would not reasonably expect boundary conditions at separate locations to interact, energetically. The same simplicity of analysis follows for numerical methods, as will be seen in the first instance later in this chapter.

⁵For those with an engineering background, including those familiar with digital waveguides, the boundary term may be interpreted as the power supplied to the problem due to interaction at the boundary.

Energy-storing Boundary Conditions

The Dirichlet, Neumann, and lossy boundary conditions lead to conservation of energy, as a whole, or to instantaneous dissipation—they do not have “memory,” and are incapable of storing energy. In some cases, one might expect a boundary termination to exhibit this property—some examples include termination of a string by a mass-spring system, or also the open (radiative) termination of an acoustic tube.

One condition, often used in speech applications in order to model such radiation from the open end of a tube [197] at $x = 0$, is the following:

$$u_x(0, t) = \alpha_1 u_t(0, t) + \alpha_2 u(0, t) \quad (6.30)$$

for constants $\alpha_1 \geq 0$ and $\alpha_2 \geq 0$. Now, the energy balance (??) becomes:

$$\frac{d\mathfrak{H}}{dt} = -\alpha_1 \gamma^2 (u_t(0, t))^2 - \alpha_2 \gamma^2 u_t(0, t) u(0, t) \quad (6.31)$$

or

$$\frac{d(\mathfrak{H} + \mathfrak{H}_b)}{dt} \leq 0 \quad \text{where} \quad \mathfrak{H}_b = \frac{\alpha_2 \gamma^2}{2} (u(0, t))^2 \quad (6.32)$$

The total energy of the system is thus that of the problem over the interior, \mathfrak{H} plus that stored at the boundary, \mathfrak{H}_b , and is again non-negative and non-increasing. The term in α_2 is often viewed, in the acoustic transmission-line framework, as corresponding to the reactive (i.e., energy-storing) part of the terminating impedance of the tube.

This condition will be returned to, in the context of tube modelling in §9.5.1.

Reflection of Traveling Waves at a Boundary

The result of the choice of boundary conditions on the behaviour of the solution to the wave equation is perhaps most easily approached using the traveling wave decomposition. Considering the wave decomposition (6.15), and a boundary condition at $x = 0$ of Dirichlet type, one immediately has that

$$u^{(+)}(-t) = -u^{(-)}(t) \quad (6.33)$$

Thus the traveling waves reflect, with sign inversion, at such a boundary. For the Neumann condition (6.24), one has that

$$u^{(+)}(-t) = u^{(-)}(t) \quad (6.34)$$

Under this condition, traveling waves reflect without sign inversion. It is useful to examine this behaviour, as illustrated in Figure 6.6. The particularly simple manner in which boundary conditions may be viewed using a traveling wave decomposition has been exploited with great success in digital waveguide synthesis. It is worth noting, however, that more complex boundary conditions generally will introduce a degree of dispersion into the solution, and traveling waves become distorted upon reflection.

6.1.10 Bounds on Solution Size

Under conservative boundary conditions, the bounds (6.22) hold, regardless of whether \mathcal{D} , the spatial domain, is defined as the entire x axis, a semi-infinite domain, or a finite interval. It is important to note that both such bounds apply to derivatives of the dependent variable, and not the dependent variable itself. This is in direct contrast to the case of the harmonic oscillator, and might seem counterintuitive, but follows directly from the definition of the 1D wave equation itself: note that

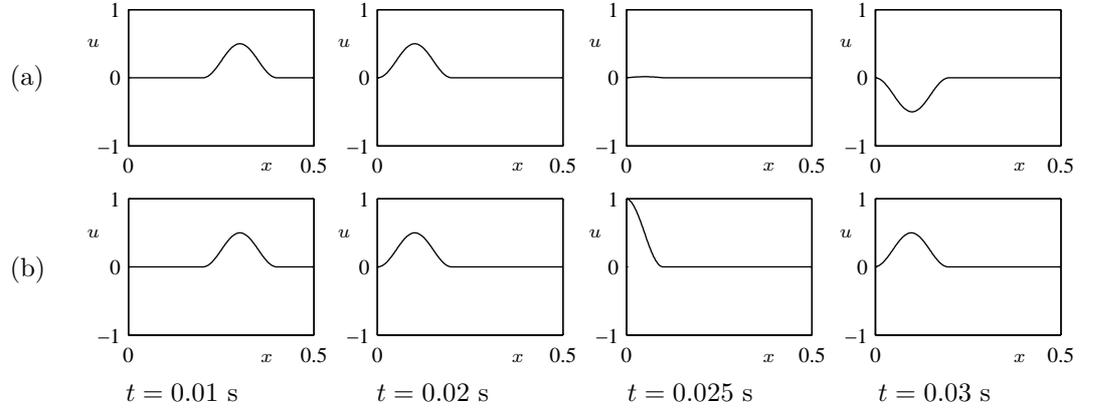


Figure 6.6: Reflection of a leftward-traveling pulse at a boundary at $x = 0$, (a) with inversion under Dirichlet, or fixed conditions, and (b) without inversion under Neumann, or free conditions. In this case, the wave equation is defined with $\gamma = 20$, and the solution is plotted over the left half of the domain, at the times as indicated.

only second derivatives appear, so that any solution of the form

$$u(x, t) = a_{00} + a_{01}t + a_{10}x + a_{11}xt \tag{6.35}$$

automatically satisfies the wave equation for any constants a_{00} , a_{10} , a_{01} and a_{11} , and such a solution can clearly not be bounded for all t and all x . The wave equation, unless properly terminated, allows a solution which is capable of drifting.

Nevertheless, conditions may indeed be employed to derive bounds of a less strict type on the size of the solution itself. Take, for instance, instance, the first of conditions (6.22). One may write, for any domain \mathcal{D} ,

$$\|u\|_{\mathcal{D}} \frac{d}{dt} \|u\|_{\mathcal{D}} = \frac{1}{2} \frac{d}{dt} \|u\|_{\mathcal{D}}^2 = \langle u, u_t \rangle_{\mathcal{D}} \leq \|u\|_{\mathcal{D}} \|u_t\|_{\mathcal{D}} \leq \|u\|_{\mathcal{D}} \sqrt{2\mathcal{H}(0)} \tag{6.36}$$

where the first inequality above follows from the Cauchy-Schwartz inequality, and the second results from the first of bounds (6.22). One thus has

$$\frac{d}{dt} \|u\|_{\mathcal{D}} \leq \sqrt{2\mathcal{H}(0)} \quad \rightarrow \quad \|u\|_{\mathcal{D}}(t) \leq \|u\|_{\mathcal{D}}(0) + \sqrt{2\mathcal{H}(0)}t \tag{6.37}$$

and the norm of the solution at time t is bounded by an affine function of time t which depends only on initial conditions; growth is no faster than linear.

As one might expect, better bounds are possible if fixed boundary conditions are employed. Considering again the wave equation defined over the semi-infinite domain $\mathcal{D} = \mathbb{R}^+$, with the Dirichlet condition (6.23) applied at $x = 0$, one has, at any point $x = x_0$,

$$|u(x_0, t)| = \left| \int_{x=0}^{x=x_0} u_x(x, t) dx' \right| = |\langle u_x, 1 \rangle_{[0, x_0]}| \leq \|u_x\|_{[0, x_0]} \|1\|_{[0, x_0]} \tag{6.38}$$

$$\leq \|u_x\|_{\mathbb{R}^+} \|1\|_{[0, x_0]} \tag{6.39}$$

$$\leq \frac{1}{\gamma} \sqrt{2\mathcal{H}(0)x_0} \tag{6.40}$$

where in this case, “1” represents a function of value 1. Here, again, the Cauchy-Schwartz inequality has been used in the first inequality, and for the third, the second of bounds (6.22) has been employed. The magnitude of the solution at any point x_0 in the semi-infinite domain may thus be bounded in terms of its distance from the end point. Notice that this is in fact a much stronger condition than a bound on the norm of the solution (in fact, it is such a bound, but in a Chebyshev, or L^∞ type norm). If the spatial domain \mathcal{D} above is changed to the finite interval $\mathbb{U} = [0, 1]$, the above analysis is unchanged, and, as long as the boundary condition at $x = 1$ is conservative, one may go further and write

$$|u(x_0, t)| \leq \frac{1}{\gamma} \sqrt{2\mathcal{H}(0)} \quad \implies \quad \|u\|_{\mathbb{U}} \leq \frac{1}{\gamma} \sqrt{2\mathcal{H}(0)} \quad (6.41)$$

which is indeed a bound on the L_2 norm of the solution. If the boundary condition at $x = 1$ is also of Dirichlet type, an improved bound is possible. See Problem 6.1.

6.1.11 Modes

As mentioned in Chapter 1, modal techniques appeared early on as a physical modeling sound synthesis method. The 1D wave equation is the usual point of departure in acoustics texts [174] for any discussion of the concept of modes of vibration. The procedure for determining the modal behaviour of an LTI system such as the wave equation (6.1) is slightly more general than that of dispersion analysis, in that boundary conditions are taken into account.

As a starting point, consider the behaviour of a test solution of the form

$$u(x, t) = U(x)e^{j\omega t} \quad (6.42)$$

When inserted into the wave equation, the following ordinary differential equation results:

$$-\omega^2 U = \gamma^2 \frac{d^2 U}{dx^2} \quad (6.43)$$

which has solutions

$$U(x) = A \cos(\omega x / \gamma) + B \sin(\omega x / \gamma) \quad (6.44)$$

So far, boundary conditions have not been enforced—for a string of infinite length, any frequency ω yields a possible solution of the above form. Consider now the wave equation over the finite domain $\mathcal{D} = \mathbb{U}$, with fixed boundary conditions at $x = 0$ and $x = 1$. Clearly, in order for $U(x)$ to vanish at $x = 0$, one must have $A = 0$; the condition at $x = 1$ leads, immediately, to

$$\omega_p = p\pi\gamma \quad U_p(x) = B_p \sin(p\pi x) \quad \text{for integer } p \neq 0 \quad (6.45)$$

where the subscript p on ω_p , $U_p(x)$ and B_p indicates the restriction of frequencies and solutions to a countably infinite set. Thus the test functions (6.42) solve the wave equation with fixed boundary conditions when they are of the form

$$u_p(x, t) = B_p \sin(p\pi x) e^{jp\pi\gamma t} \quad (6.46)$$

for some constants B_p . A general solution is then of the form

$$u(x, t) = \sum_{p=1}^{\infty} B_p \sin(p\pi x) e^{jp\pi\gamma t} \quad (6.47)$$

Insisting on a real-valued solution leads a semi-infinite expansion,

$$u(x, t) = \sum_{p=1}^{\infty} (\alpha_p \cos(p\pi\gamma t) + \beta_p \sin(p\pi\gamma t)) \sin(p\pi x) \quad (6.48)$$

for some real constants α_p and β_p . These constants may be related to initial conditions by noting

that

$$u(x, 0) = \sum_{p=1}^{\infty} \alpha_p \sin(p\pi x) \quad u_t(x, 0) = \sum_{p=1}^{\infty} p\pi\gamma\beta_p \sin(p\pi x) \quad (6.49)$$

From Fourier series arguments, and given two initial conditions $u(x, 0) = u_0(x)$ and $u_t(x, 0) = v_0(x)$, one may then deduce that

$$\alpha_p = 2 \int_{\mathbb{U}} u_0(x) \sin(p\pi x) dx \quad \beta_p = \frac{2}{\omega_p} \int_{\mathbb{U}} v_0(x) \sin(p\pi x) dx, \quad p = 1, \dots, \infty \quad (6.50)$$

The above modal frequencies $\omega_p = p\pi\gamma$ and functions $U_p(x)$ hold for the wave equation under fixed, or Dirichlet conditions, and must be recomputed as the boundary conditions are varied. Under free conditions at both ends, it is straightforward to show that the modal frequencies and functions are of the form

$$\omega_p = p\pi\gamma \quad U_p(x) = A_p \cos(p\pi x) \quad p = 0, \dots, \infty \quad (6.51)$$

The allowed frequencies are the same as in the case of fixed-fixed termination, with the minor exception that a zero-frequency solution is possible (with $p = 0$)—this follows immediately from rigid body motion, which is possible under free conditions. Under mixed conditions (i.e., fixed at $x = 0$ and free at $x = 1$), the frequencies and functions are given by

$$\omega_p = (p - 1/2)\pi\gamma \quad U_p(x) = A_p \sin((p - 1/2)\pi x) \quad p = 1, \dots, \infty \quad (6.52)$$

In this case, the lowest modal frequency is half of that in the case of fixed/fixed or free/free terminations. When such conditions are used to model the behaviour of a musical instrument, the pitch will be an octave lower than in the other cases, and with only odd harmonics present. Such conditions are by no means unrealistic in a musical setting, and serve to describe, to a crude approximation, wind instruments which make use of a uniform cylindrical section of tubing, closed at one end and open at the other.

Representations of the modal functions for all three sets of boundary conditions are given in Figure 6.7.

Though for the simple boundary conditions discussed here, the modal frequencies may be expressed in closed form, and the modal functions themselves may be written as trigonometric functions, it is worth keeping in mind that for even slightly more complex systems (such as, e.g., the 1D wave equation under some other more involved boundary termination, or virtually any other system), modal frequencies and functions must be determined numerically. See Problem 6.4, and the musically relevant case of the vibrating bar, in §7.1.3.

Modal Density and Degrees of Freedom

As a prelude to discrete time simulation of the 1D wave equation, it is useful to say a word about the number of degrees of freedom of the system, which will ultimately determine computational complexity for any algorithm. See the general comments on this topic in §1.3.3. The modal decomposition allows such information to be extracted in a particularly easy way.

The modal description of the 1D wave equation is complete, in the sense that an expansion such as (6.130) is a general solution to the PDE formulation (6.1). The initial conditions may be expanded onto the various modes, yielding, at each frequency ω_p , two coefficients, α_p and β_p , corresponding to an initial displacement and velocity for a given mode. These two coefficients are sufficient to determine the time evolution of the mode, and thus each mode possesses two degrees of freedom. In any practical digital application (such as sound synthesis), the number of modes must be restricted to a finite set—a natural choice is those whose frequencies which lie below half

of the audio sampling rate, f_s . In other words, one needs only those (positive) frequencies for which $\omega_p \leq \pi f_s$. Considering, for example, the case of the 1D wave equation with Dirichlet conditions imposed at either end, the modal frequencies fall in a series given in (6.45) above. For the digital system operating at sample rate f_s , the total number of degrees of freedom N_m will be twice the number of frequencies which appear in the range $\omega_p \in [0, \pi f_s]$, or,

$$N_m \approx 2f_s/\gamma \quad (6.53)$$

The number N_m is a measure of the complexity of the 1D wave equation in a discrete-time simulation at sample rate f_s —it is, roughly, the number of units of computer memory that will be required in order to compute such a solution, and the number of arithmetic operations also scales with N_m . It is important to note that this number remains roughly the same, regardless of the type of boundary conditions which are applied. (If for instance, one makes use of Dirichlet conditions at one end of the domain, and Neumann conditions at the other, the set of modal frequencies given in (6.52) results, leading to the same estimate of complexity as in (6.53) above.) Even more important, and as will be seen at various instances throughout this chapter, is that the complexity remains roughly the same for *any* numerical method which simulates the wave equation at a given sample rate f_s , with the interesting exception of digital waveguides—see §6.2.11.

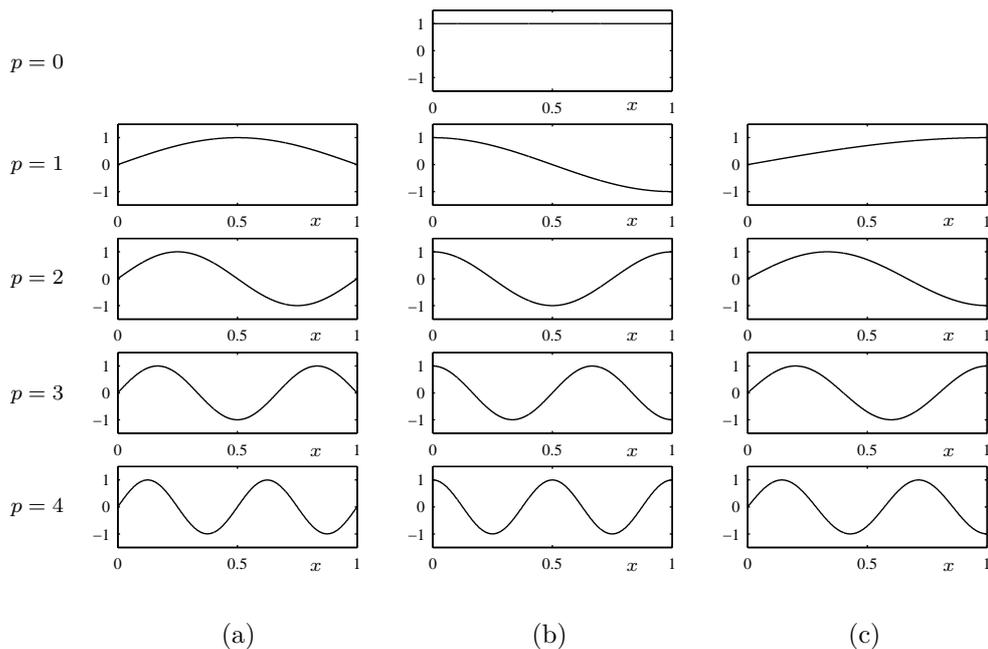


Figure 6.7: Modal functions for the 1D wave equation (6.1), as a function of $x \in \mathbb{U} = [0, 1]$, under various different boundary terminations: (a) fixed at both ends, (b) free at both ends, and (c) fixed at the left end, and free at the right end. Modal functions are indexed by integer p —in cases (a) and (c) above, the first four modes are shown, and in case of free-free termination, an extra constant mode with $p = 0$ (which can be thought of as a DC offset) is also shown. All modal functions are normalized to to the range $[-1, 1]$.

6.2 A Simple Finite Difference Scheme

The most rudimentary finite difference scheme for the 1D wave equation (and in almost all respects, the best) is given, in operator form, as

$$\delta_{tt}u = \gamma^2 \delta_{xx}u \quad (6.54)$$

Here, as mentioned previously, u is shorthand notation for the grid function u_i^n , representing an approximation to the solution of the wave equation at $x = lh$, $t = nk$, where again, h is the spacing between adjacent grid points, and k is the time step. (See Chapter 5 for an overview of the difference operator notation used here and elsewhere in this chapter.) As the difference operators employed are second-order accurate, the scheme itself is in general second order accurate in both time and space. (In fact, under a special choice of h and k , its order of accuracy is infinite—in other words, it can yield an exact solution. This special case is confined to the simple wave equation alone, but forms the basis for digital waveguides, to be discussed in Section 6.2.11.)

When the action of the operators is expanded out, a recursion results:

$$u_i^{n+1} = 2(1 - \lambda^2)u_i^n - u_i^{n-1} + \lambda^2(u_{i-1}^n + u_{i+1}^n) \quad (6.55)$$

The important dimensionless parameter λ , often referred to as the Courant number has been defined here by

$$\lambda = \gamma k/h \quad (6.56)$$

This scheme may be updated, explicitly, at each time step n , from previously computed values at the previous two time steps. It is perhaps easiest to see the behaviour of this algorithm through a dependence plot showing the “footprint” of the scheme (6.54), shown in Figure 6.8.

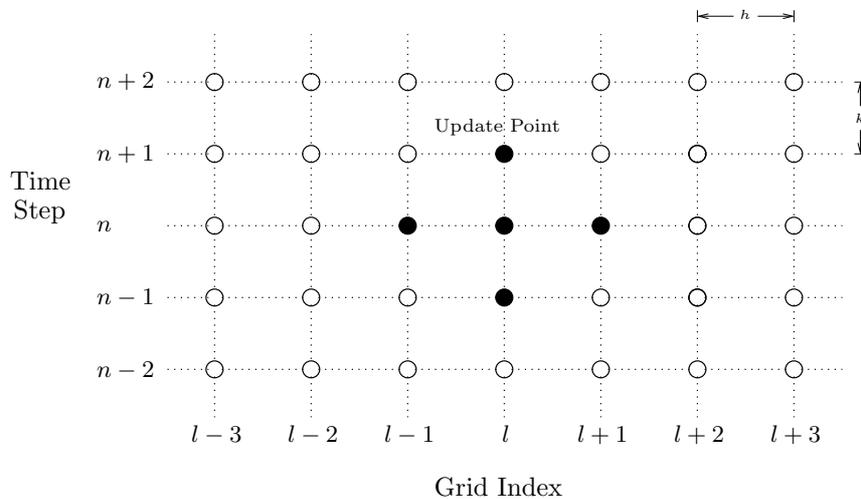


Figure 6.8: *Computational footprint of scheme (6.54)—a value of the grid function u_i^n at the update location $(l, n + 1)$, as indicated, is updated using values at the previous two time steps. The set of points related under the scheme at this update location is indicated in black.*

For the moment, it is assumed that the spatial domain of the problem is infinite; the analysis of boundary conditions may thus be postponed temporarily, simplifying analysis somewhat.

6.2.1 Initialization

Scheme (6.54), like the wave equation itself, must be initialized with two sets of values. From the explicit form of the recursion (6.55), it should be clear that one must specify values of the grid function u_l^n at time steps $n = 0$ and $n = 1$, namely u_l^0 and u_l^1 . Supposing that the initial displacement u_l^0 is given, and that one also has a grid function v_l^0 representing the initial velocity, one may then write for u_l^1 :

$$u_l^1 = u_l^0 + kv_l^0 \quad (6.57)$$

which is exactly the same procedure as in the case of the simple harmonic oscillator. See §3.2.2, and Problem 3.4. This initialization strategy, however, can lead to unexpected results, however, especially when the initializing distribution is very sharply peaked—see Programming Exercise 6.1.

6.2.2 von Neumann Analysis

Frequency domain analysis of the scheme (6.54) is similar to that applied to the continuous time/space wave equation. As a shortcut to full z -transform, and spatial discrete Fourier transform analysis, consider again the behaviour of a test solution of the form

$$u_l^n = z^n e^{jl\beta h} \quad (6.58)$$

where $z = e^{nsk}$. (See §5.2.6.) This again may be thought of as a wave-like solution, and in fact, it is identical to the wave-like solution employed in the continuous case, sampled at $t = nk$ and $x = lh$. When substituted into the difference scheme (6.55), the following characteristic equation results:

$$z + 2(2\lambda^2 \sin^2(\beta h/2) - 1) + z^{-1} = 0 \quad (6.59)$$

which is analogous to (6.12) for the 1D wave equation. The roots are given by

$$z_{\pm} = 1 - 2\lambda^2 \sin^2(\beta h/2) \pm \sqrt{(1 - 2\lambda^2 \sin^2(\beta h/2))^2 - 1} \quad (6.60)$$

Again, just as for the wave equation, there are two solutions (resulting from the use of a second difference in time), representing the propagation of the test solution in opposite directions.

One would expect that, in order for a solution such as (6.58) to behave as a solution to the wave equation, one should have $|z| = 1$ for any value of β , the wavenumber—otherwise such a solution will experience exponential growth or damping. This is not necessarily true for scheme (6.54). From simple inspection of the characteristic equation (6.59), and using the same techniques applied to the harmonic oscillator in §3.2.4, one may deduce that the roots are complex conjugates of unit magnitude when

$$|2\lambda^2 \sin^2(\beta h/2) - 1| \leq 1 \quad (6.61)$$

which can be rewritten as

$$\lambda^2 \sin^2(\beta h/2) \leq 1 \quad (6.62)$$

This inequality must be satisfied for any possible value of β . Because $\sin^2(\beta h/2)$ is bounded by one, the condition

$$\lambda \leq 1 \quad (6.63)$$

is sufficient for stability. This is the famous Courant-Friedrichs-Lewy condition [68], which has the interesting physical interpretation as illustrated in Figure 6.9.

It is useful to note that for λ slightly greater than unity, condition (6.62) will be violated near the maximum of the function $\sin^2(\beta h/2)$, which occurs at the wavenumber $\beta = \pi/h$, corresponding to a wavelength of $2h$ —from basic sampling theory, this is the shortest wavelength which may be represented on a grid of spacing h . A typical result of numerical instability is then the explosive

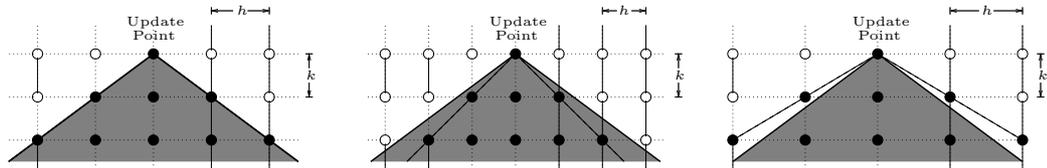


Figure 6.9: A geometrical interpretation of the Courant-Friedrichs-Lewy stability condition (6.63) for scheme (6.54). At a given update point (as indicated in the figures), the value of the solution to the continuous time wave equation depends on values traveling on solution characteristics (solid dark lines), defined by $x - \gamma t = \text{constant}$ and $x + \gamma t = \text{constant}$. The cone of dependence of the solution may be illustrated as the interior of this region (in grey). The scheme (6.54) at the update point possesses a numerical cone of dependence, illustrated by black points, and bounded by dashed black lines. At left, $h = \gamma k$, and the characteristics align exactly with values on the grid—in this case, the numerical solution is exact. At center, a value $h < \gamma k$ is chosen, violating stability condition (6.63)—the numerical cone of dependence lies strictly within the region of the dependence of wave equation, and as such, the scheme cannot compute an accurate solution, and is unstable. At right, with a choice of $h > \gamma k$, the numerical cone of dependence of the scheme includes that of the wave equation, and the scheme is stable.

growth of such a component at the spatial Nyquist wavelength. See Figure 6.10.

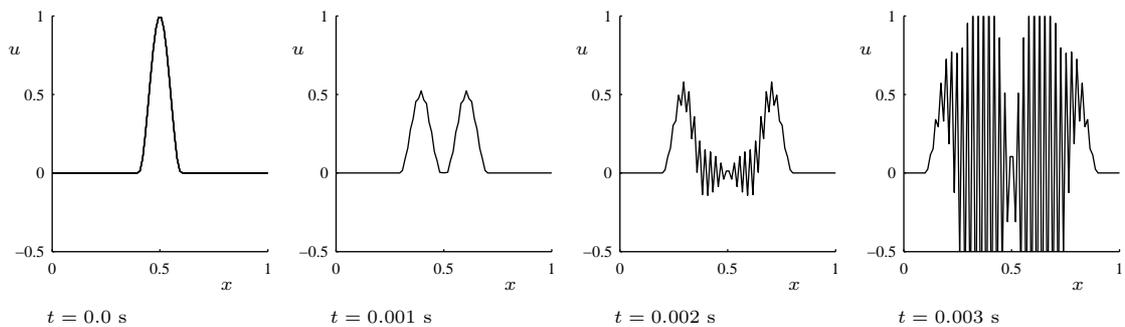


Figure 6.10: Numerical Instability. Left to right: successive outputs u of scheme (6.54), with a value of $\lambda = 1.0125$, violating condition (6.63). The sample rate is chosen as 8000 Hz, γ as 100, and initial conditions are of a plucked raised cosine, amplitude 1, half-width 0.1, and centered at $x = 0.5$. As time progresses, components of the solution near the spatial Nyquist frequency quickly grow in amplitude; within a few time steps after the last plot in this series, the computed values of the solution become out of range.

The CFL Condition as a Bound on Computational Complexity

As seen above, the condition (6.63) above serves as a bound which must be respected in order that scheme (6.54) remain stable. It does not, however, say anything about how close to this bound λ should be chosen. In fact, it is usually a good idea to choose λ as close to the stability limit as possible, as will be shown here.

First, consider the scheme (6.54), defined over the unit interval $\mathcal{D} = \mathbb{U}$. For a grid spacing h , the number of grid points covering the unit interval will be approximately $1/h$, and for a two step scheme such as (6.54), the number of memory locations required, degrees of freedom N_{fd} will be twice this number. The stability condition (6.63) can then be rewritten as a bound on N_{fd} :

$$N_{fd} \leq \frac{2f_s}{\gamma} \quad (6.64)$$

Notice the similarity with the measure of the number of degrees of freedom N_m for a modal simulation, from (6.53). When (6.64) holds with equality, the two measures are the same. What is interesting, however, is that N_{fd} may be chosen smaller than this: the CFL bound (6.63) specifies a lower bound on h in terms of k , which seems to imply that one could reduce the number of grid points at which a numerical solution is to be calculated, resulting in a cheaper numerical method. One could, in fact, do just this, but in fact, one is merely cheating the dynamics of the wave equation itself, and wasting valuable audio bandwidth. It is easiest to see this after having a look at effects of numerical dispersion, to be discussed next.

6.2.3 Numerical Dispersion

Consider now scheme (6.54) under the stability condition (6.63). Using $z = e^{j\omega k}$, the characteristic equation (6.59) may be written as

$$-4 \sin^2(\omega k/2) = -4\lambda^2 \sin^2(\beta h/2) \quad \rightarrow \quad \sin(\omega k/2) = \pm \lambda \sin(\beta h/2) \quad (6.65)$$

or

$$\omega = \pm \frac{2}{k} \sin^{-1}(\lambda \sin(\beta h/2)) \quad (6.66)$$

This is a dispersion relation for the scheme (6.54), relating frequency ω and wavenumber β in complete analogy with the relation (6.13) for the wave equation itself.

Just as in the case of a continuous system, one may define phase and group velocities for the scheme, now in a numerical sense, exactly as per (6.14). (See the end of §5.2.6 for some general discussion of numerical phase and group velocities.) In the case of the wave equation, recall that these velocities are constant, and equal to γ for all wavenumbers. Now, however, both the phase and the group velocity are, in general, functions of wavenumber—in other words, different wavelengths travel at different speeds, and the scheme (6.54) is thus dispersive. Dispersion leads to a progressive distortion of a pulse as it travels, as illustrated in Figure 6.11. See Problem 6.12. This type of anomalous behaviour is purely a result of discretization, and is known as numerical dispersion—it should be carefully distinguished from physical dispersion of a model problem itself, which will appear when systems are subject to stiffness—such systems will be examined shortly in Chapter 7. It is also clear that the numerical velocities will also depend on the choice of the parameter λ , which may be freely chosen for scheme (6.54), subject to the constraint (6.63). It is useful to plot the velocity curves, as functions of frequency for different values of λ , as shown in Figure 6.12.

Now, consider the very special case of $\lambda = 1$. Under this condition, the dispersion relation (6.66)

reduces to

$$\omega = \frac{2}{k} \sin^{-1}(\sin(\beta h/2)) = \frac{\beta h}{k} = \gamma \beta \quad (6.67)$$

Now, the numerical dispersion relation is exactly that of the continuous wave equation, and the phase and group velocities are both equal to γ , independently of β . There is thus no numerical dispersion for this choice of λ —this is a reflection of the fact that the scheme (6.54) is exact when $\lambda = 1$, as will be discussed in the next section.

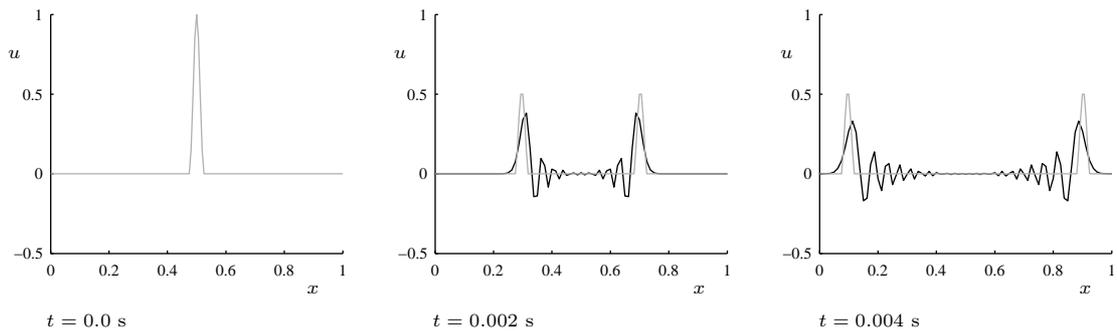


Figure 6.11: *Numerical dispersion.* Outputs for scheme (6.54) for the wave equation, at times as indicated, for $\lambda = 1$ (in grey) and $\lambda = 0.5$ (in black). γ is chosen as 100, the sample rate is 16000 Hz, and the initial conditions are set according to a narrow cosine distribution, of width $1/40$. Notice that for λ away from 1, the higher-frequency components in general lag the wavefront, illustrating the phase velocity characteristic of the scheme. Notice also that the gross speed of the wave packet is slower as well, illustrating the group velocity characteristic.

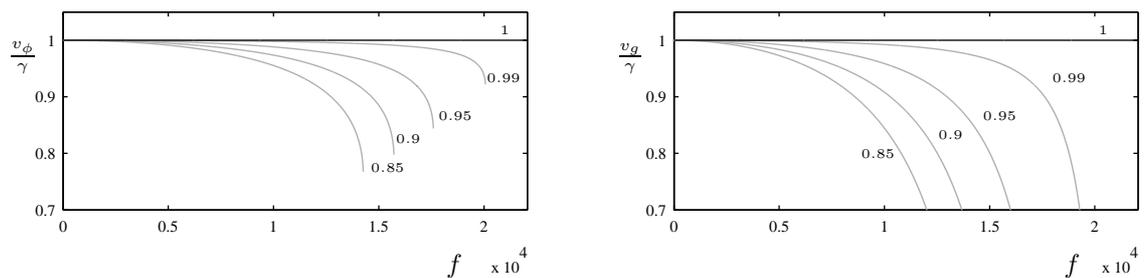


Figure 6.12: *Numerical phase velocity (left) and group velocity as a function of frequency f , normalized by the model velocity γ , for scheme (6.54), for a variety of values of λ , as indicated. The sample rate is chosen as 44100 Hz.*

Operation Away from the CFL Bound

This is a good opportunity to return to the question of raised at the end of §6.2.2, namely that of the possibility of reduced computational complexity when $\lambda < 1$. One way of approaching this is to ask: what frequencies is scheme (6.54) capable of producing? The answer is most easily seen from (6.65) above. Considering only the positive solution, and using $\omega = 2\pi f$ and $f_s = 1/k$, that the maximum such frequency f_{max} will be given by

$$\sin(\pi f_{max}/f_s) = \lambda \quad \rightarrow \quad f_{max} = \frac{f_s}{\pi} \sin^{-1}(\lambda) < \frac{f_s}{2} \quad (6.68)$$

Thus, when $\lambda < 1$, scheme (6.54) is, in essence, not capable of filling the available audio bandwidth; the smaller λ is chosen, the lower the bandwidth of the scheme. See Figure 6.13. As a result, one can conclude that in order to reduce computational complexity, it is a far better idea to simply reduce the audio sample rate of the simulation, rather than to choose an unnaturally large value for the grid spacing (i.e., one which does not allow sufficient resolution of wavelengths at the desired sample rate). As mentioned above, such a choice also leads to undesirable numerical dispersion effects. This may be summarized in the following way:

Rule of Thumb # 1

For an explicit numerical method, the best numerical behaviour (i.e., the least numerical dispersion) when the stability condition is satisfied with equality.

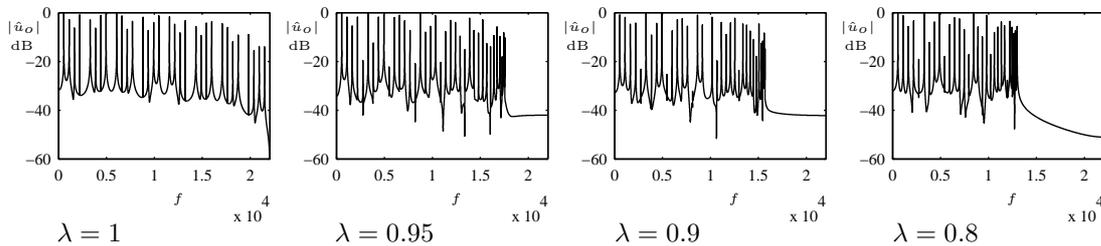


Figure 6.13: Typical magnitude spectra $|\hat{u}_o|$ of outputs from scheme (6.54) in dB as a function of frequency f , under different choices of the parameter λ , as indicated in the figure. In this case, the initial conditions are a raised cosine distribution $c_{rc}(x)$, with $x_0 = 0.3$, $x_{hw} = 0.05$, and $c_0 = 1$, and the readout location is $x = 0.6$. The sample rate is 44100 Hz, and $\gamma = 1102.5$. Notice in particular the reduction in the effective bandwidth of the computed solution, as λ becomes farther from the Courant Friedrichs Lewy bound at $\lambda = 1$. Also visible is the inharmonicity resulting from numerical dispersion, also increasing as λ is moved away from the CFL bound.

6.2.4 Accuracy

The difference operators δ_{xx} and δ_{tt} employed in scheme (6.54) are second-order accurate (see §5.2.5), and one might suspect that the solution generated will also exhibit this level of accuracy. Recall, however, from the discussion of a special scheme for the harmonic oscillator in §3.3.4, that accuracy of a scheme can, in some cases, be better than that of the constituent operators. Consider, now, the

scheme (6.54) written in the following form:

$$P_d u = 0 \quad \text{with} \quad P_d = \delta_{tt} - \gamma^2 \delta_{xx} \quad (6.69)$$

where u is the grid function u_i^n . Supposing, now, that P_d operates on the continuous function $u(x, t)$, one may again expand the behaviour of P_d in Taylor series, to get

$$P_d = \frac{\partial^2}{\partial t^2} - \gamma^2 \frac{\partial^2}{\partial x^2} + \sum_{p=2}^{\infty} \frac{2}{(2p)!} \left(k^{2(p-1)} \frac{\partial^{2p}}{\partial t^{2p}} - \gamma^2 h^{2(p-1)} \frac{\partial^{2p}}{\partial x^{2p}} \right) \quad (6.70)$$

$$= \frac{\partial^2}{\partial t^2} - \gamma^2 \frac{\partial^2}{\partial x^2} + O(k^2) + O(h^2) \quad (6.71)$$

Thus the difference operator P_d approximates the differential operator corresponding to the 1D wave equation to second order in both k and h . Note, however, that when $\lambda = 1$, or when $k\gamma = h$, the operator may be written as

$$P_d = \frac{\partial^2}{\partial t^2} - \gamma^2 \frac{\partial^2}{\partial x^2} + \sum_{p=2}^{\infty} \frac{2k^{2p-1}}{(2p)!} \left(\frac{\partial^{2p}}{\partial t^{2p}} - \gamma^{2p} \frac{\partial^{2p}}{\partial x^{2p}} \right) \quad (6.72)$$

$$(6.73)$$

Just as in the case of scheme (3.38), each term in the summation above possesses a factor of the form $\frac{\partial^2}{\partial t^2} - \gamma^2 \frac{\partial^2}{\partial x^2}$, showing that, in fact, the difference scheme (6.54) does indeed generate an exact solution. Such delicate cancellation of numerical error has been exploited in the construction of highly accurate techniques often referred to as ‘‘modified equation’’ methods [231, 69].

6.2.5 Energy Analysis

The energetic analysis of scheme (6.54) is very similar to that of the wave equation itself. It is easiest to begin with scheme (6.54) in its condensed operator form, and consider the unbounded spatial domain $\mathcal{D} = \mathbb{Z}$. Taking the inner product of scheme (6.54) with the grid function defined by

$$\delta_t \cdot u \quad (6.74)$$

which is an approximation to the velocity, gives

$$\langle \delta_t \cdot u, \delta_{tt} u \rangle_{\mathbb{Z}} = \gamma^2 \langle \delta_t \cdot u, \delta_{xx} u \rangle_{\mathbb{Z}} \quad (6.75)$$

After employing summation by parts (see §5.2.10), one has

$$\langle \delta_t \cdot u, \delta_{tt} u \rangle_{\mathbb{Z}} + \gamma^2 \langle \delta_{x+} \delta_t \cdot u, \delta_{x+} u \rangle_{\mathbb{Z}} = 0 \quad (6.76)$$

This may be written as the total difference

$$\delta_{t+} \left(\frac{1}{2} \|\delta_{t-} u\|_{\mathbb{Z}}^2 + \frac{\gamma^2}{2} \langle \delta_{x+} u, e_{t-} \delta_{x+} u \rangle_{\mathbb{Z}} \right) \quad (6.77)$$

or, after identifying the terms inside the brackets above with kinetic and potential energy, as

$$\delta_{t+} \mathfrak{h} = 0 \quad (6.78)$$

with

$$\mathfrak{t} = \frac{1}{2} \|\delta_{t-} u\|_{\mathbb{Z}}^2 \quad \mathfrak{v} = \frac{\gamma^2}{2} \langle \delta_{x+} u, e_{t-} \delta_{x+} u \rangle_{\mathbb{Z}} \quad \mathfrak{h} = \mathfrak{t} + \mathfrak{v} \quad (6.79)$$

The total difference (6.78) above is a statement of conservation of numerical energy for the scheme (6.54), and it thus follows that

$$\mathfrak{h}^n = \mathfrak{h}^0 \quad (6.80)$$

Scheme (6.54) is exactly conservative, in a numerical sense, for this special case of a spatial domain of infinite extent. This is true regardless of the values chosen for the time step k and

the grid spacing h . Just as in the case of the scheme for the SHO, discussed in Chapter 3, in an implementation, there will be variations in the energy on the order of “machine epsilon.” See Figure 6.14 for an illustration of this.

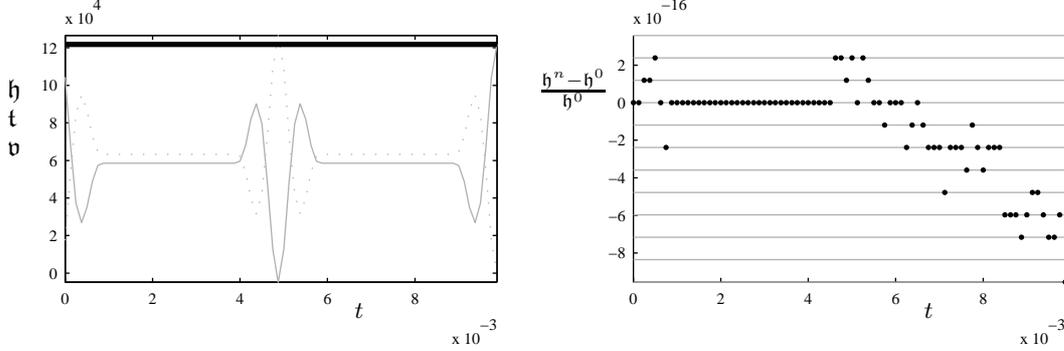


Figure 6.14: Numerical energy conservation for scheme (6.54), with $\gamma = 100$, $f_s = 8000$, $\lambda = 1$, and where the initial condition is a plucked cosine distribution, centered at $x = 0.5$, of amplitude 1, and of half-width 0.1. At left, the total energy \mathfrak{h} (solid black line), \mathfrak{v} (solid grey line) and \mathfrak{t} (dotted grey line) are plotted against time; notice that it is possible, at certain instants, for the numerical potential energy \mathfrak{v} to take on negative values. At right, the variation of the energy $(\mathfrak{h}^n - \mathfrak{h}^0)/\mathfrak{h}^0$ is also plotted against time; multiples of the bit quantization error are plotted as solid grey lines.

6.2.6 Numerical Boundary Conditions

In order to examine boundary conditions in a simple way, it is useful to examine scheme (6.54) defined over the semi-infinite domain $\mathcal{D} = \mathbb{Z}^+$. Now, an inner product of scheme (6.54) with δ_t over \mathbb{Z}^+ gives

$$\langle \delta_t u, \delta_{tt} u \rangle_{\mathbb{Z}^+} = \gamma^2 \langle \delta_t u, \delta_{xx} u \rangle_{\mathbb{Z}^+} \quad (6.81)$$

After employing summation by parts, according to (5.22), boundary terms appear, i.e.,

$$\langle \delta_t u, \delta_{tt} u \rangle_{\mathbb{Z}^+} + \gamma^2 \langle \delta_{x+} \delta_t u, \delta_{x+} u \rangle_{\mathbb{Z}^+} = \mathfrak{b} \triangleq -\gamma^2 (\delta_t u_0) (\delta_{x-} u_0) \quad (6.82)$$

or

$$\delta_{t+} \mathfrak{h} = \mathfrak{b} \quad (6.83)$$

with

$$\mathfrak{t} = \frac{1}{2} \|\delta_{t-} u\|_{\mathbb{Z}^+}^2 \quad \mathfrak{v} = \frac{\gamma^2}{2} \langle \delta_{x+} u, e_{t-} \delta_{x+} u \rangle_{\mathbb{Z}^+} \quad \mathfrak{h} = \mathfrak{t} + \mathfrak{v} \quad (6.84)$$

One again has exact conservation under the conditions

$$u_0 = 0 \quad \text{or} \quad \delta_{x-} u_0 = 0 \implies u_0 = u_{-1} \quad (6.85)$$

The first condition above is exactly equivalent to the Dirichlet (or fixed) boundary condition at a left endpoint; the second is a first difference approximation to the Neumann condition; although it refers to an imaginary grid point at index $l = -1$, in fact, the condition is telling us how to eliminate this variable by setting it in terms of the values in the domain interior (in this case, the value of the grid function at u_0). Thus, at $l = 0$, scheme (6.55) may be modified to

$$u_0^{n+1} = (2 - \lambda^2) u_0^n - u_0^{n-1} + \lambda^2 u_1^n \quad (6.86)$$

Alternative Inner Products and Boundary Conditions

As mentioned in §6.1.9, there are many possibilities for boundary termination of the continuous time-space wave equation beyond the simple Dirichlet and Neumann conditions. In the case of a numerical method, such as (6.54), the number of possible terminations becomes even greater, as there is a multiplicity of ways of approximating a given continuous boundary condition in a discrete setting. Indeed, with the right choice of inner product, one may prove numerical stability for other numerical boundary conditions, some of which may have benefits in terms of accuracy.

The Dirichlet condition $u = 0$ at a boundary of the domain is relatively simple to deal with, numerically: a condition of $u_0 = 0$ falls immediately out of energy analysis, and is exact. The approximation to the Neumann condition arrived at in this way involves a non-centered first difference. One might wonder about the energetic behaviour of a centered condition, such as $\delta_x \cdot u_0$. To this end, perform the energy analysis of scheme (6.54) as previously, but make use of the alternative inner product $\langle \cdot, \cdot \rangle'_{\mathbb{Z}^+}$ instead. In this case, one has

$$\delta_{t+} \mathfrak{h} = \mathfrak{b} \triangleq -\gamma^2 \delta_t \cdot u_0 \delta_x \cdot u_0 \quad (6.87)$$

with

$$\mathfrak{t} = \frac{1}{2} (\|\delta_{t-} u\|'_{\mathbb{Z}^+})^2 \quad \mathfrak{v} = \frac{\gamma^2}{2} \langle \delta_{x+} u, e_{t-} \delta_{x+} u \rangle_{\mathbb{Z}^+} \quad \mathfrak{h} = \mathfrak{t} + \mathfrak{v} \quad (6.88)$$

One again has exact conservation under the conditions

$$u_0 = 0 \quad \text{or} \quad \delta_x \cdot u_0 = 0 \implies u_1 = u_{-1} \quad (6.89)$$

The primed inner product also makes for a convenient analysis of the lossy boundary condition (6.28). Consider the numerical approximation

$$\delta_t \cdot u_0 = \alpha \delta_x \cdot u_0 \quad (6.90)$$

In this case, $\mathfrak{b} = -\alpha \gamma^2 (\delta_t \cdot u_0)^2 \leq 0$, so that numerical energy is non-increasing. (Under a particular choice of α , this boundary condition has the interesting property of being numerically perfectly absorbing—see Problem 6.6 and Programming Exercise 6.3.)

6.2.7 Bounds on Solution Size and Numerical Stability

One might thus wonder how the concept of numerical stability, which follows from frequency domain analysis, intervenes in the energetic framework for scheme (6.54). Take, as a starting point, the expressions for numerical energy which appear in (6.88), in the case of scheme (6.54) defined over $\mathcal{D} = \mathbb{Z}^+$, under conservative numerical boundary conditions (6.89). The key point, just as for the case of the simple harmonic oscillator, is that in contrast with the energy \mathfrak{H} defined for the continuous system, \mathfrak{h} is not necessarily positive definite, due to the indefinite nature of the numerical potential energy term \mathfrak{v} .

To determine conditions under which the numerical energy \mathfrak{h} is positive definite, one may proceed as in the case of the simple harmonic oscillator, and write, for the potential energy,

$$\mathfrak{v} = \frac{\gamma^2}{2} \langle \delta_{x+} u, e_{t-} \delta_{x+} u \rangle_{\mathbb{Z}^+} = \frac{\gamma^2}{2} \left(\|\delta_{x+} \mu_{t-} u\|_{\mathbb{Z}^+}^2 - \frac{k^2}{4} \|\delta_{x+} \delta_{t-} u\|_{\mathbb{Z}^+}^2 \right) \quad (6.91)$$

$$\geq \frac{\gamma^2}{2} \left(\|\delta_{x+} \mu_{t-} u\|_{\mathbb{Z}^+}^2 - \frac{k^2}{h^2} \|\delta_{t-} u\|_{\mathbb{Z}^+}^2 \right) \quad (6.92)$$

Here, in the equality above, the identity (2.22f) for time difference operators has been used, and in

the inequality, the bound (5.27) on spatial differences. Thus, one has for the total energy,

$$\mathfrak{h} = \mathfrak{t} + \mathfrak{v} \geq \frac{1}{2} (1 - \lambda^2) \|\delta_{t-} u\|_{\mathbb{Z}^+}^2 + \frac{\gamma^2}{2} \|\delta_{x+\mu t-} u\|_{\mathbb{Z}^+}^2 \quad (6.93)$$

This discrete conserved energy is thus non-negative under the condition $\lambda \leq 1$, which is the same as the condition (6.63) arrived at using von Neumann analysis.

In order to obtain a general bound on the solution size, independent of boundary conditions, one may note that under the condition (6.63), one then has

$$\|\delta_{t-} u\|_{\mathbb{Z}^+} \leq \sqrt{\frac{2\mathfrak{h}^0}{1 - \lambda^2}} \quad (6.94)$$

which implies, further, that for the grid function u^n at any time step n , one must have

$$\|u\|_{\mathbb{Z}^+} \leq nk \sqrt{\frac{2\mathfrak{h}^0}{1 - \lambda^2}} \quad (6.95)$$

Thus, just as for the bound (6.37) in the case of the continuous wave equation, growth of the l_2 norm of the discrete solution is no faster than linear. If boundary conditions are of Dirichlet or fixed type, then, just as in the continuous case, better bounds are available. See Problem 6.8.

Instability at Boundaries

The proper discretization of boundary conditions is a very delicate business, as far as stability is concerned—though, for a simple system such as the wave equation, the “obvious” numerical choices turn out to work properly (as will be shown in the next section), for more complex systems, there are many more choices, and, thus, many more ways to go astray.

As a rudimentary (though somewhat artificial) example of a “bad” choice of boundary condition in the present case of scheme (6.54) for the wave equation, consider the following Neumann-type condition at grid point $l = 0$:

$$\alpha \delta_{x-} u_0 + (1 - \alpha) \delta_{x \cdot} u = 0 \quad (6.96)$$

where α is a free parameter. This can be viewed as a weighted linear combination of the two conditions $\delta_{x-} u_0 = 0$ and $\delta_{x \cdot} u_0 = 0$, both of which lead to stable simulation results. One might argue then that such a condition must also lead to stable results, but, in fact, depending on the choice of α , it can become unstable, as illustrated in Figure 6.15, with a choice of $\alpha = -0.05$. Notice that the nature of this instability is of a different nature from that resulting from the violation of the CFL condition, illustrated in Figure 6.10. Here, the unstable behaviour originates from the boundary location itself, rather than over the entire domain. From a programming and debugging standpoint, it is obviously helpful to be able to classify such types of instability. It is interesting to note that the boundary condition above is, in fact, conservative, provided one makes the right choice of inner product to build an energy function. Thus, again, energy conservation is not sufficient for numerical stability, but rather non-negativity of conserved energy. See Problem 6.5.

The analysis of numerical stability, including the effect of boundary conditions has been carried out by many authors. One classic, and very powerful technique, introduced by Gustafsson, Kreiss, Sundstrom and Osher [114, 147, 185], is based on frequency domain analysis of the combined scheme/boundary condition system, and reduces essentially to conditions on root locations in the complex plane. Another technique involves performing eigenvalue analysis of a scheme, when written in state space form—see the next section. This is often referred to as the “matrix method.” As might be apparent, such a technique is generally applicable to systems which are LTI (and not merely LSI), though the extension to nonlinear systems becomes problematic. The greater difficulty,

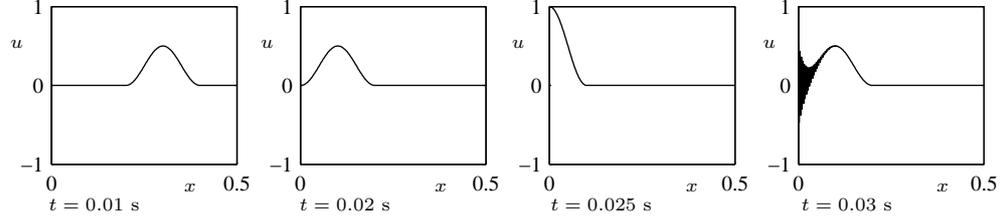


Figure 6.15: *Numerical instability at an improperly chosen boundary condition. In this case, as an approximation to the Neumann condition at $x = 0$, the condition $\alpha\delta_{x-}u_0 + (1 - \alpha)\delta_{x+}u_0 = 0$ is used, with a value of $\alpha = -0.05$. The parameters and initial conditions are otherwise as for the example shown in Figure ???. Within a few time steps of the last figure shown, the computed values are out of range.*

pointed out by Strikwerda [244] is that such analysis does not give much guidance as to how to proceed if, indeed, the eigenvalue analysis reveals a stray unstable mode.

Energy analysis can give sufficient conditions for stability under a particular choice of boundary condition, in particularly insightful way. As an example, consider the wave equation defined over \mathbb{R}^+ , with the radiative boundary condition (6.30) applied to the left end (this condition is crucial for brass and woodwind instrument modeling). Here is one possible numerical boundary condition:

$$\delta_{x-}u_0 = \alpha_1\delta_{t-}u_0 + \alpha_2u_0 \quad (6.97)$$

From the numerical energy balance (6.87) in the primed inner product, one has, for the boundary term \mathfrak{b} ,

$$\mathfrak{b} = -\gamma^2\alpha_1(\delta_{t-}u_0)^2 - \gamma^2\alpha_2(\delta_{t-}u_0)u_0 = -\gamma^2\alpha_1(\delta_{t-}u_0)^2 - \frac{\gamma^2\alpha_2}{2}\delta_{t+}(u_0e_{t-}u_0) \quad (6.98)$$

where, in the second step above, identity (2.22b) has been employed. Now, however, one may write, for energy balance (6.87),

$$\delta_{t+}(\mathfrak{h} + \mathfrak{h}_b) = -\gamma^2\alpha_1(\delta_{t-}u_0)^2 \quad \text{where} \quad \mathfrak{h}_b = \frac{\gamma^2\alpha_2}{2}u_0e_{t-}u_0 \quad (6.99)$$

This balance, involving a stored energy over the problem interior (\mathfrak{h}) and at the boundary (\mathfrak{h}_b) is completely analogous to (6.32) for the continuous system. The numerical energy of the system as a whole must decrease. But, as before, it is the positivity condition on the new combined numerical energy which leads to a stability condition. One may write, for \mathfrak{h} and \mathfrak{h}_b , using steps similar to those employed previously,

$$\mathfrak{h} \geq \frac{1}{2}(1 - \lambda^2) \left(\|\delta_{t-}u\|_{Z^+}' \right)^2 \quad \mathfrak{h}_b \geq -\frac{\gamma^2\alpha_2k^2}{8}(\delta_{t-}u_0)^2 \quad (6.100)$$

or

$$\mathfrak{h} + \mathfrak{h}_b \geq \sum_{l=1}^{\infty} \frac{h}{2}(1 - \lambda^2)(\delta_{t-l}u_l)^2 + \left(\frac{h}{4}(1 - \lambda^2) - \frac{\lambda^2\alpha_2h^2}{8} \right) (\delta_{t-}u_0)^2 \quad (6.101)$$

The first term above is non-negative under the CFL condition (6.63). For the second term, however, the condition is more strict:

$$\lambda \leq \frac{1}{1 + \alpha_2h/2} \quad (6.102)$$

The above condition is sufficient for stability under the boundary condition (6.97). It is not, however, necessary, and there do exist other numerical boundary conditions which do not lead to an extra interfering condition—see Problem 6.7 and Programming Exercise 6.4. Similar issues arise when dealing with connections with lumped objects—see, e.g., §7.7.1 for an example, and discussion of a general principle behind ensuring numerical stability for connections of objects.

6.2.8 State Space Form and Modes

For those with a background in electrical engineering (and, in particular, control theory), it is useful to rewrite the difference scheme (6.54) in state space, or vector-matrix form. Though in this book, the difference operators are sparse, and thus do not require full storage in matrix form, for implicit schemes, it will be necessary to solve linear systems involving such matrices.

Suppose that the values of the grid function u_l^n to be computed at time step n are arranged in a finite-length column vector \mathbf{u} . (If it is necessary compute values of the grid function at the endpoints, the vector will be of the form $\mathbf{u}^n = [u_0^n, \dots, u_N^n]^T$, if not, then it will be of the form $\mathbf{u}^n = [u_1^n, \dots, u_{N-1}^n]^T$.) The operator δ_{xx} in matrix form must be specialized from the infinite form given in (5.15), taking boundary conditions into account. Some examples of such forms include those given in (5.16).

Regardless of the exact form of \mathbf{D}_{xx} , difference scheme (6.54) may be written, in vector-matrix form, as

$$\mathbf{u}^{n+1} = (2\mathbf{I} + \gamma^2 k^2 \mathbf{D}_{xx}) \mathbf{u}^n - \mathbf{u}^{n-1} \quad \text{or} \quad \underbrace{\begin{bmatrix} \mathbf{u}^{n+1} \\ \mathbf{u}^n \end{bmatrix}}_{\mathbf{w}^{n+1}} = \underbrace{\begin{bmatrix} 2\mathbf{I} + \gamma^2 k^2 \mathbf{D}_{xx} & -\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \mathbf{u}^n \\ \mathbf{u}^{n-1} \end{bmatrix}}_{\mathbf{w}^n} \quad (6.103)$$

where here, \mathbf{I} and $\mathbf{0}$ indicate identity and zero matrices of the appropriate size. The first form above is a two-step recursion in the vector \mathbf{u}^n , and the second a one-step recursion in the expanded vector $\mathbf{w}^n = [(\mathbf{u}^n)^T, (\mathbf{u}^{n-1})^T]^T$, which can be thought of as the state of the finite difference scheme (6.54). The one-step form is known as a state-space representation.

Examining the first form above, and assuming solutions of the form $\mathbf{u} = \phi z^n$, the characteristic equation

$$\frac{1}{k^2}(z - 2 + z^{-1})\phi = \gamma^2 \mathbf{D}_{xx} \phi \quad (6.104)$$

results. The solutions, ϕ_p , which are the eigenvectors of \mathbf{D}_{xx} , whatever form it takes, can be interpreted as the modal functions of difference scheme (6.54); in general, due to discretization error, they will be distinct from the modes of the wave equation itself. The associated modal frequencies ω_p will then be given as solutions to the quadratic equations

$$z_p + (-2 - \gamma^2 k^2 \text{eig}_p(\mathbf{D}_{xx})) + z_p^{-1} = 0 \quad (6.105)$$

where $\text{eig}_p(\mathbf{D}_{xx})$ signifies “ p th eigenvalue of,” and where $z_p = e^{j\omega_p k}$. If the eigenvalues of \mathbf{D}_{xx} satisfy the condition $-4/k^2 \leq \text{eig}(\mathbf{D}_{xx}) \leq 0$ (this is similar to the CFL condition (6.63) for the scheme), then the roots of this characteristic equation will be on the unit circle, with the numerical frequencies given by

$$\omega_p = \frac{2}{k} \sin^{-1} \left(k\gamma \sqrt{\text{eig}_p(-\mathbf{D}_{xx})} \right) \quad (6.106)$$

Table 6.1: Modal frequencies, in Hertz, for scheme (6.54) for the wave equation, with $\gamma = 882$, and under fixed boundary conditions, operating at a sample rate of $f_s = 44100$ Hz.

Mode number	$\lambda = 1$	$\lambda = 0.98$	$\lambda = 0.94$	$\lambda = 0.9$	$\lambda = 0.8$
1	441.0	441.0	441.0	441.0	441.0
2	882.0	882.0	881.9	881.8	881.7
3	1323.0	1322.9	1322.7	1322.5	1321.9
4	1764.0	1763.8	1763.4	1762.9	1761.4
5	2205.0	2204.6	2203.8	2202.9	2199.9
6	2646.0	2645.3	2643.9	2642.3	2637.0

Numerical Inharmonicity

In a musical setting, the perceptual counterpart to numerical dispersion, or the artifact of variation in wave speed with frequency, is a detuning of modal frequencies from those of the model equation itself; generally, for finite difference schemes, this will be an effect of increasing importance with higher frequencies.

As an example, consider the 1D wave equation, with parameter $\gamma = 882$, under fixed boundary conditions. The modal frequencies are integer multiples of $\gamma/2 = 441$. It is interesting to look at the behaviour of the modal frequencies of scheme (6.54), operating at sample rate $f_s = 44100$ Hz, under different choices of the Courant number λ —these may be calculated from (6.106) above. See Table 6.1.

There are several things worth noting here. As expected, the frequencies are exact when $\lambda = 1$. For other values of λ , the frequencies become successively detuned from these exact values, both as the modal frequencies become higher, and as λ is moved away from the stability condition at $\lambda = 1$. On the other hand, even under dispersive conditions, the detuning of modal frequencies is rather small, and, in the worst case in the table presented above, a little under 6 cents. This is a rather complex issue, psychoacoustically speaking. One might argue that, given that human frequency sensitivity is very limited beyond about 3000 Hz [291], numerical dispersion is thus not really of perceptual importance. This is partly true—in fact, the sound output generated using such an algorithm will indeed have a pitch of almost exactly 441 Hz, under various choices of λ , as the reader may wish to verify by running the code example provided in §A.4. What the reader will also notice, however, for values of λ away from 1, is that the inharmonic upper partials tend to become very closely spaced in the upper range of the spectrum (see, e.g., Figure 6.13), generating a somewhat unpleasant noise-like component to the output! Though in more realistic models, of, e.g., strings, such an effect will always be subsumed by losses which are strong at high frequencies, the conclusion here is that the question of the perceptual importance of numerical dispersion depends strongly on the problem at hand, and is an important design consideration, if one is employing direct simulation techniques—indeed, modal methods, as well as digital waveguides in the 1D case exhibit no such effect. Another source of detuning is the precise nature of the numerical boundary condition—although the numerical Dirichlet condition applied to the scheme (6.54) is indeed exact, this is not necessarily true of other conditions. The Neumann condition is one example—see Programming Exercise 6.5.

6.2.9 Output and Interpolation

The generation of sound output is, obviously, the end goal of physical modeling synthesis. Clearly, a complete consideration of the signal path from a vibrating object to the human eardrum is enormously complex, and depends not merely on the vibrating object itself, but also the geometry and other properties of the room in which it is vibrating, the position of the listener with respect to the instrument, etc. In principle, one could attempt a full time domain simulation of the accompanying ambient system—but such a calculation is gargantuan, and dwarfs that of the vibrating object. Some comments on the feasibility of full 3D room acoustics simulations appear in §16.1.2. If one is interested in ambient effects, one is far better off making use of simplified systems based on psychoacoustic considerations []. Indeed, as wave propagation in acoustic spaces is linear, the entire perceptual effect of the room on a generated musical sound can be well described using transfer function models, using as an input the vibration amplitude of the musical object at its surface. For these reasons, in this book, any discussion of output will be limited to direct reading of vibration amplitude—if one wants to go further and add in room effects, one may do so in a subsequent step!

For a time-domain method, certainly the easiest way to generate output is to merely read a value from the grid function as it is being updated, at a given observation point. In the case of scheme (6.54), and supposing that the output location is x_o , this may be done using interpolation—see §5.2.4. If $I_p(x_o)$ is the corresponding p th order interpolation operator, an output time series u_o^n may be generated from the grid function u_i^n for scheme (6.54) as

$$u_o^n = I_p(x_o)u_i^n \quad (6.107)$$

This is a “feed-forward” step, and clearly there is no risk of instability, once u^n has been calculated. It is also extremely cheap—no more than a handful of operations are required for this computation, and certainly far fewer than the number required by the recursion (6.54) itself. Indeed, if the output location is fixed, it is probably a good idea to use zeroth-order interpolation (i.e., rounding of the output location to that of the grid point to its left), and avoid arithmetic altogether. One may wish to process the time series u_o^n with, e.g., a simple high-pass filter crudely modeling radiation effects [].

As expected, there will be a spectral effect associated with the choice of output location—see Figure 6.16, where output waveforms and frequency spectra are plotted for distinct output locations. For instance, if output is read from the string center, all even-numbered harmonics will necessarily be absent from the output spectrum. More generally, the choice of output location leads to spectra windowing effects

Moving Output Locations

For a time domain method, there is no reason why the output location cannot itself be moving—see the footnote on page 103 for some justification for this. Indeed, the idea of reading from a physical model at variable locations has been proposed as a synthesis technique known as “scanned synthesis” [?], though in that case, the readout point may be moving at audio rates, effectively modulating the output of the physical model. Here, however, it will be assumed that the motion is slow relative to the speed of wave propagation, leading to a phasing effect, as various harmonics are cancelled. See Figure 6.17. In contrast to the case of a static output location, good interpolation will be necessary, otherwise audible distortions (“clicks,” as well as potentially time-varying low-passing) will occur in the output.

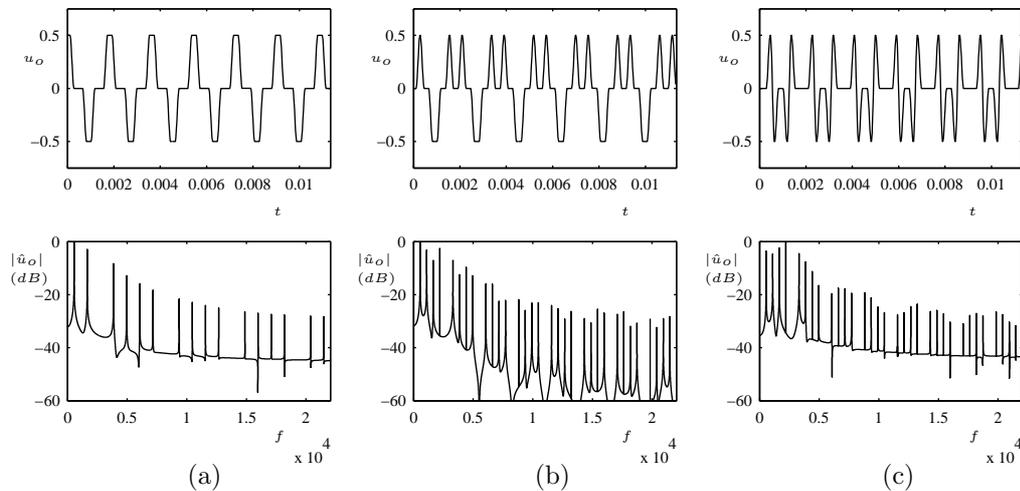


Figure 6.16: Output of scheme (6.54), with $f_s = 44100$ Hz, and $\gamma = 1102.5$, with an initial “plucked” condition of the form of a sharp pulse centered at $x = 0.6$, at different locations: (a) at $x = 0.5$, (b) at $x = 0.3$, and (c) at $x = 0.1$. In each case, the output waveform is shown at top, and the frequency spectrum at bottom.

Multiple Output Locations

It is also straightforward, once one has computed a grid function u_l^n , to read output simultaneously at various points—these will exhibit some slight spectral variations. When mapped to different loudspeakers (possibly many), the spectral variations serve as an excellent means of spatializing (in a very crude sense) physical modeling output. It is important to note that, for time domain methods, computational cost is approximately independent of the number of outputs, because the entire state of the virtual object is directly observable. In this sense, multiple outputs come “for free.” The same is true for digital waveguide models, but not for methods based on modal synthesis—see the comments on this topic in §6.6. See also Programming Exercise 6.6, which deals with multiple moving output locations in the case of the wave equation.

6.2.10 Implementation Details and Computational Requirements

Now that the finite difference scheme (6.54) has been examined from many different perspectives, it is worth discussing some of the practical considerations that arise when one settles down to program such a method for synthesis. Basic example code provided in §A.4 for reference on the points raised here.

One minor consideration is that of the choice of time step k and grid spacing h . For synthesis (in stark opposition to other mainstream applications), it is the time step which is chosen first, as $k = 1/f_s$, where f_s is a typical audio sample rate such as $f_s = 32000, 44100, 48000$ Hz, etc. The grid spacing h must then be chosen according to the CFL condition (6.63). Clearly, at least in the case of the 1D wave equation, a choice of $\lambda = 1$ is ideal, as it leads to an exact solution. But in real-world

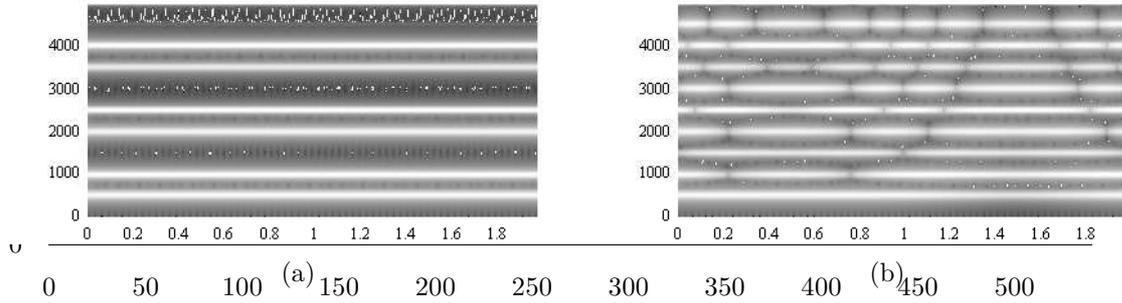


Figure 6.17: Spectrograms of output of scheme (6.54), with $f_s = 44100$ Hz, and $\gamma = 1000$, under a “plucked” initial condition of the form of a sharp spike. In (a), the output location is fixed at $x = 1/3$, and in (b), the output location varies sinusoidally about $x = 1/3$, at a sub-audio rate.

applications, the spatial domain will always be the unit interval $\mathcal{D} = \mathbb{U}$, and the grid spacing h must divide this domain into an integer number of parts, i.e., $h = 1/N$, for some integer N . As discussed in §6.2.3, however, it is always advantageous to choose the grid spacing such that the CFL condition is satisfied as near to equality as possible. Thus, in an initial step, one may set $\lambda = 1$, and then choose N as:

$$N := \text{floor}(\lambda/\gamma k) \quad \text{then :} \quad h := 1/N \quad \text{and reset} \quad \lambda := \gamma k/h \quad (6.108)$$

Generally, for sufficiently high sample rates and low values of γ (or fundamental frequency), the dispersion introduced will be minimal. Another strategy is to simply choose $\lambda = 1$ and h accordingly, and allow the domain to be of a length slightly different from 1—this is less desirable, however, as it can lead to an audible mistuning of the fundamental, a phenomenon which does not occur if the previous method is employed. Digital waveguide methods on the other hand (see next section), rely on the choice of $\lambda = 1$, and thus only this second method is viable.

In general, the computational requirements will depend on the choice of the sample rate, the grid spacing (mediated by the CFL condition), as well as the complexity of the scheme footprint. In this case, as may be read from the explicit form of the recursion (6.55), updating at a given grid point requires, for a general choice of λ , two multiplications and three additions, or five arithmetic operations. When the domain is limited to the unit interval, evaluations at approximately $1/h$ such points will be required—one says approximately because, depending on boundary conditions, one will have some minor variations in the number of grid points. Thus $5/h$ arithmetic operations are required per step in the recursion, and thus, in one second, $5/hk$ operations will be required. If h and k are chosen according to (6.63) close to equality, then the full count will be $5f_s^2/\gamma$ operations per second. Thus computational complexity scales as the square of the sample rate. The number N_{fd} of required units of memory is as given in (6.64), corresponding to two grid functions worth of data. Note that in the implementation in §A.4, for programming simplicity, three separate grid functions are used—this can be reduced to two, in the same way as has been discussed with respect to the simple harmonic oscillator—see §3.2.7 and Programming Exercise 3.1.

6.2.11 Digital Waveguide Interpretation

Consider the scheme (6.54), for the special case of $\lambda = 1$. As was noted in §6.2.3, the numerical phase and group velocities of the scheme are identical to those of the wave equation itself, and thus independent of frequency. The expanded recursion (6.55) reads, now, as

$$u_i^{n+1} = u_{i+1}^n + u_{i-1}^n - u_i^{n-1} \quad (6.109)$$

Notice, in particular, that (1) there are no longer any multiplications necessary to update the scheme, and (2), the center grid point in the scheme footprint (see Figure 6.8) is no longer used.

It is straightforward to identify this simplified scheme as an exact traveling wave solution to the wave equation. First consider the traveling wave solutions themselves, defined by

$$w^{(+)}(x, t) = u^{(+)}(x - \gamma t) \quad w^{(-)}(x, t) = u^{(-)}(x + \gamma t) \quad (6.110)$$

At a location $x = lh$, $t = nk$, one has, immediately, that

$$\begin{aligned} w^{(+)}(lh, nk) &= u^{(+)}(lh - \gamma nk) = u^{(+)}(h(l - n)) = u^{(+)}(h(l - 1) - h(n - 1)) = w^{+}((l - 1)h, (n - 1)h) \\ w^{(-)}(lh, nk) &= u^{(-)}(lh + \gamma nk) = u^{(-)}(h(l + n)) = u^{(-)}(h(l + 1) + h(n - 1)) = w^{-}((l + 1)h, (n - 1)h) \end{aligned}$$

The above relations follow only under the condition that $\lambda = 1$, or, equivalently, that $h = \gamma k$. Notice that no approximations have been employed. Replacing the functions $w^{(+)}$ and $w^{(-)}$ by grid functions, one has

$$w_l^{(+),n} = w_{l-1}^{(+),n-1} \quad w_l^{(-),n} = w_{l+1}^{(-),n-1} \quad u_l^n = w_l^{(+),n} + w_l^{(-),n} \quad (6.111)$$

The first two identities above indicate propagation of discrete wave-like variables to the right and left respectively, and the third that the observable physical variable u may be written as a sum of two such waves.

Given the three definitions above, one may then write:

$$\begin{aligned} u_l^{n+1} = w_l^{(+),n+1} + w_l^{(-),n+1} = w_{l-1}^{(+),n} + w_{l+1}^{(-),n} &= u_{l+1}^n + u_{l-1}^n - w_{l-1}^{(-),n} - w_{l+1}^{(+),n} \\ &= u_{l+1}^n + u_{l-1}^n - w_l^{(-),n-1} - w_l^{(+),n-1} \\ &= u_{l+1}^n + u_{l-1}^n - u_l^{n-1} \end{aligned}$$

which is identical to the simplified scheme (6.109).

The key to an efficient implementation is the recognition of the fact that the solution may be advanced purely through shifting operations applied to the wave variables $w^{(+)}$ and $w^{(-)}$. u itself need only be computed in a “feed-forward” step at points at which an output is desired. See Figure 6.18. Typically, in audio, scalar outputs only are desired, and one does not need to observe the entire state of the object under consideration. There is thus a major distinction between sound synthesis and other applications, which perhaps explains why although such exact discrete traveling wave solutions to the wave equation have been known for decades [3], this potential for increased efficiency was only seized upon by Smith [242]. More will be said about this in §6.6.

In terms of the wave variables $w^{(+)}$ and $w^{(-)}$, as mentioned above, only shifting operations are required to advance the solution, leading to the well-known bidirectional delay line form, as shown in Figure 6.18. For the sake of reference, an implementation of the digital waveguide is provided in §A.6.

Initialization may be carried through some discrete approximation to (6.16). That is, if the continuous initializing functions $u_0(x)$ and $q_0(x) = \int v_0(x') dx'$ are known, they may be sampled

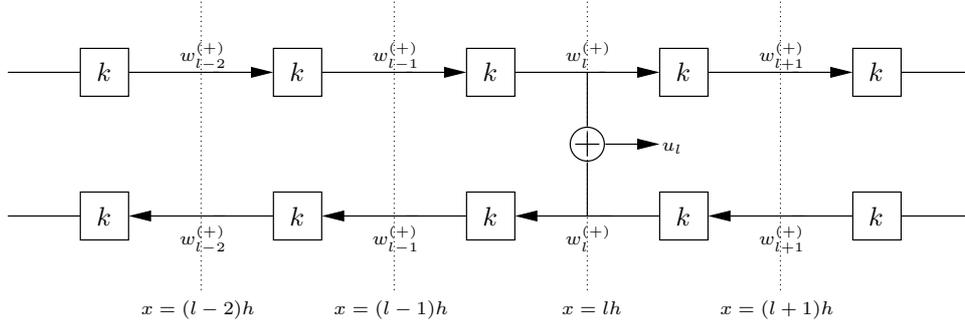


Figure 6.18: Schematic showing the delay line, or digital waveguide interpretation of (6.111). Here, delays of duration k seconds are indicated by boxes. The spacing between grid locations h must be related to the delay k by $h = \gamma k$.

directly to yield the wave variables as

$$w_l^{(+),0} = \frac{1}{2}u_0(lh) - \frac{1}{2\gamma}q_0(lh) \quad w_l^{(-),0} = \frac{1}{2}u_0(lh) + \frac{1}{2\gamma}q_0(lh) \quad (6.112)$$

This is clearly simpler when the excitation is of “plucked” type (i.e., $v_0(x) = 0$). If $q_0(x)$ is not expressible in closed form, then $v_0(x)$ may be integrated numerically—for instance, one could recursively generate the grid function $q_{0,l} = q_{0,l-1} + hv_0(lh)$, and use it in place of the samples $q_0(lh)$ in the above formula.

Boundary Conditions

Boundary termination of digital waveguides can be straightforward, for simple choices of boundary conditions. The most common types are as shown in Figure 6.19, in the case of a termination at the left end of the domain.

Termination with sign inversion, with the leftward-propagating delay line feeding back into the rightward propagating line is particularly easy to analyze. If the termination point is situated directly on the boundary, at $x = 0$, then one has $w_0^{(+),n} = -w_0^{(-),n}$. This immediately implies, from summing of waves as per (6.111), that $u_0^n = w_0^{(+),n} + w_0^{(-),n} = 0$, corresponding to the Dirichlet type numerical boundary condition, as discussed in §6.1.9.

Another type of termination involves feeding back the signal from the leftward propagating delay line into the rightward propagating delay line, without sign inversion, and with an extra element of delay. It can be shown that this corresponds to a Neumann-type numerical boundary condition in scheme (6.54):

$$\begin{aligned} u_0^{n+1} = w_0^{(+),n+1} + w_0^{(-),n+1} = w_0^{(-),n} + w_1^{(-),n} &= u_0^n - w_0^{(+),n} + u_1^n - w_1^{(+),n} \\ &= u_0^n + u_1^n - w_0^{(-),n-1} - w_0^{(+),n-1} \\ &= u_0^n + u_1^n - u_0^{n-1} \end{aligned}$$

This is exactly the scheme (6.54), with $\lambda = 1$, at the grid point $l = 0$, under the condition that $\delta_x^- u_0 = 0$.

Both these conditions were shown, earlier, to conserve numerical energy, leading to stability conditions. In the waveguide picture, however, stability is obvious: the delay line is itself obviously lossless, and terminations such as sign inversions, or simple delays cannot serve to cause solution

growth. More elaborate settings for boundary conditions terminating digital waveguides are a central feature of waveguide modeling. See the text by Smith for much more on this topic.

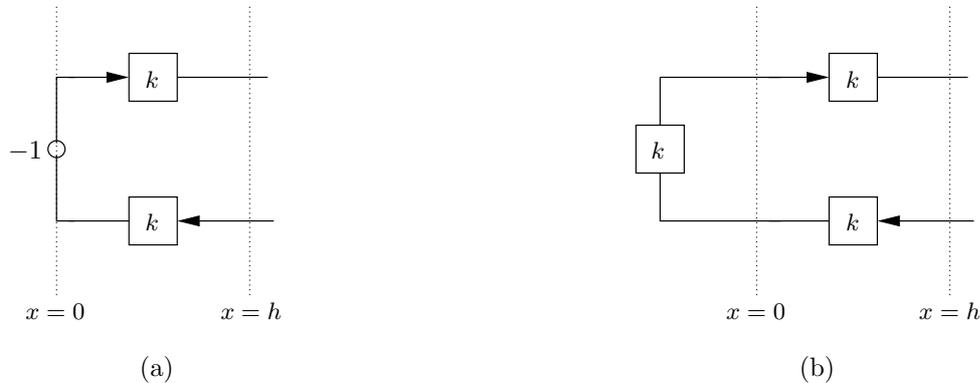


Figure 6.19: Terminations of a digital waveguide at a left end corresponding to (a) a fixed, or Dirichlet boundary condition, and (b) a free or Neumann condition.

There is obviously a lot more to say about waveguides, and though there will be some further reference to scattering structures (see §6.6 in this chapter for some general comments on waveguides, and comparison with other synthesis methods, and §11.1.6 for the 2D extension of the digital waveguide), the present treatment must unfortunately end here. The interested reader is referred to the ample literature on this subject—see the references in §1.2.3.

6.3 Other Schemes

Given that scheme (6.54) is exact for $\lambda = 1$, there is not practical interest in exploring more elaborate schemes for the 1D wave equation. On the other hand, few systems permit such an exact solution, and the 1D wave equation is thus quite singular in this regard, and not at all representative of any other system of interest in musical acoustics. For this reason, some more complex finite difference constructions are briefly described in this section. For much more on elaborate finite difference constructions for the 1D wave equation, see, e.g., [261, 181, 182, 69, 23].

6.3.1 A Stencil-width Five Scheme

A simple explicit generalization of scheme (6.54) involves a wider-stencil centered approximation to the spatial derivative term:

$$\delta_{tt}u = \gamma^2 (\alpha + (1 - \alpha)\mu_{x.}) \delta_{xx}u \quad (6.113)$$

This scheme involves a free parameter α , and reduces to scheme (6.54) when $\alpha = 1$; it is nominally second-order accurate in both space and time, which is immediately evident since all operators are centered. Notice, in particular, the use of the operator $\alpha + (1 - \alpha)\mu_{x.}$, which is an approximation to the identity operation, as discussed at the end of §5.2.2. The form above may be expanded into the following recursion:

$$u_i^{n+1} = (2 + \lambda^2(1 - 3\alpha))u_i^n + \lambda^2(2\alpha - 1)(u_{i-1}^n + u_{i+1}^n) + \frac{\lambda^2}{2}(1 - \alpha)(u_{i+2}^n + u_{i-2}^n) - u_i^{n-1} \quad (6.114)$$

This scheme clearly makes use of points two grid spacings removed from the update point, and thus has a stencil width of five. See Figure 6.20(a) for a representation of the stencil or “footprint”

corresponding to this scheme.

The characteristic equation for scheme (6.113) is easily obtained, again through the insertion of a test solution of the form $u_l^n = z^n e^{jl\beta h}$:

$$z + (-2 + 4\lambda^2 (1 - 2(1 - \alpha) \sin^2(\beta h/2)) \sin^2(\beta h/2)) + z^{-1} = 0 \quad (6.115)$$

The condition that the roots of the characteristic equation be of unit magnitude is thus, from (2.15), that

$$0 \leq \lambda^2 (1 - 2(1 - \alpha) \sin^2(\beta h/2)) \sin^2(\beta h/2) \leq 1 \quad (6.116)$$

for all values of β . In order to approach the analysis of this apparently unwieldy expression, note that what is really needed is to find the maximum and minimum values of a polynomial over an interval. Writing $p = \sin^2(\beta h/2)$, and noting that $0 \leq p \leq 1$, the conditions above may be written as

$$\min_{p \in [0,1]} F(p) \geq 0 \quad \text{and} \quad \lambda^2 \leq \frac{1}{\max_{p \in [0,1]} F(p)} \quad \text{where} \quad F(p) = p(1 - 2(1 - \alpha)p) \quad (6.117)$$

The left-hand inequality is satisfied for

$$\alpha \geq 1/2 \quad (6.118)$$

Given the above restriction on α , the right hand inequality yields a rather complex bound on λ , which is dependent on the chosen value of α :

$$\lambda \leq \begin{cases} \sqrt{8(1 - \alpha)}, & \frac{1}{2} \leq \alpha \leq \frac{3}{4} \\ \sqrt{\frac{1}{2\alpha - 1}}, & \alpha > \frac{3}{4} \end{cases} \quad (6.119)$$

See Problem 6.14.

One problem with the use of such a scheme is that because the stencil or footprint is now wider, it will be necessary to set an additional boundary condition when the scheme is restricted to a finite domain, such as \mathbb{U}_N . It is clear that updating u_l^{n+1} at grid points of index $l = 0$ and $l = 1$ requires access to values outside the domain interior, but the wave equation itself allows only a single boundary condition at the endpoint of the domain. This extra numerical boundary condition must thus be set very carefully—this is a general problem with numerical methods of high accuracy which do not operate locally. Energetic analysis for this scheme is also possible, and in fact leads to the same stability conditions above, as well as to proper settings for the numerical boundary conditions; because this scheme is of academic interest only, such analysis will not be presented here. Energy analysis of wider stencil schemes will, however, be dealt with at a later stage, in the case of the ideal bar, in §7.1.4.

In general, the choice of the parameter α has a rather large influence on the numerical phase and group velocities (and thus on the tuning of mode frequencies). See Figure 6.21(a).

6.3.2 A Compact Implicit Scheme

A generalization of scheme (6.54) of an entirely different character is the following:

$$(\theta + (1 - \theta)\mu_x) \delta_{tt}u = \gamma^2 \delta_{xx}u \quad (6.120)$$

This scheme also involves a free parameter θ , and reduces to scheme (6.54) when $\theta = 1$, and is again nominally second-order accurate in both space and time. Variants of this scheme have indeed been used for sound synthesis [51] from models of vibrating strings, and xylophone bars [116], though the version presented here is slightly different from that used in the references above (see Problem 6.16). The symbol θ has been chosen to indicate the relationship with so-called “ θ -schemes,” used for simulating first order equations, particularly in fluid dynamics applications [].

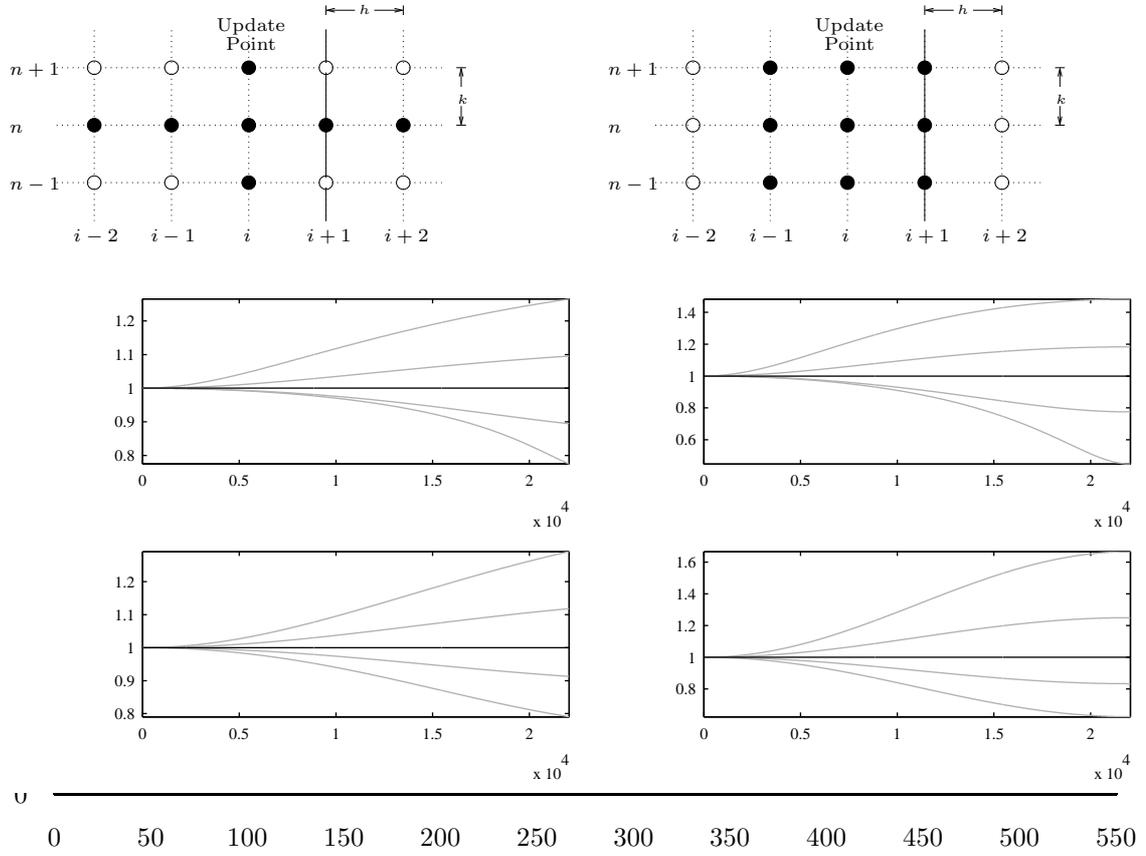


Figure 6.21: Numerical phase and group velocity for (a) scheme (6.113) and (b) scheme (6.120), under various choices of the free parameters α and θ , respectively, as indicated.

The following recursion results:

$$\theta u_i^{n+1} + \frac{1-\theta}{2} (u_{i+1}^{n+1} + u_{i-1}^{n+1}) = (\lambda^2 + 1 - \theta) (u_{i+1}^n + u_{i-1}^n) + 2(\theta - \lambda^2)u_i^n - \theta u_i^{n-1} + \frac{\theta-1}{2} (u_{i+1}^{n-1} + u_{i-1}^{n-1}) \quad (6.121)$$

In this case, the values of the grid function at the update time step $n + 1$, written on the left side of the above recursion, are coupled; this is the first instance in this book of a so-called implicit finite difference scheme. It is not possible to solve the above recursion without the use of linear system solution techniques, which are generally more costly than simple explicit updating. In order to better understand the implementation of such a scheme, matrix representations of difference operators as discussed in §5.2.7 are essential. Assuming the column vector \mathbf{u}^n to contain the values of the grid function u_i^n , then the scheme may be written as

$$\mathbf{u}^{n+1} = \mathbf{A}^{-1}\mathbf{B}\mathbf{u}^n - \mathbf{u}^{n-1} \quad \text{with} \quad \mathbf{A} = \theta\mathbf{I} + (1-\theta)\mathbf{M}_x, \quad \mathbf{B} = \gamma^2 k^2 \mathbf{D}_{xx} + 2\mathbf{A} \quad (6.122)$$

where it is assumed that if the grid function is limited to a finite set of values, boundary conditions (to be discussed shortly) will be incorporated into the above matrices. \mathbf{D}_{xx} and \mathbf{M}_x are matrix representations of the difference operators δ_{xx} and μ_x , which appear, in infinite form, in §5.2.7. The

real work involved here is in the solution of the linear system involving the matrix \mathbf{A} at each time step. A full matrix inversion of \mathbf{A} (or even full storage of the matrix \mathbf{A} itself) is to be avoided in any implementation, as \mathbf{A} is tridiagonal and thus very sparse, and various fast solution techniques, most notably the Thomas algorithm [251, 244] are often employed. See Programming Exercise 5.4 in the previous chapter, which illustrates the use of all-purpose iterative methods in the Matlab programming environment.

In exchange for this added complexity, such schemes are potentially much more accurate than explicit schemes, and often operate under much looser stability conditions (though it should be kept in mind that in this very special case, scheme (6.120) is exact when $\theta = 1$). To this end, it is worth again writing the characteristic polynomial corresponding to scheme (6.120), again employing the variable $p = \sin^2(\beta h/2)$:

$$(1 - 2(1 - \theta)p)z + (4\lambda^2 p - 2(1 - 2(1 - \theta)p)) + (1 - 2(1 - \theta)p)z^{-1} = 0 \quad (6.123)$$

The stability analysis here is similar to that of the five-point scheme discussed in the previous section, except that now, one must determine the maximum and minimum values of a rational function over the unit interval; the stability conditions are then

$$\min_{p \in [0,1]} F(p) \geq 0 \quad \text{and} \quad \lambda^2 \leq \frac{1}{\max_{p \in [0,1]} F(p)} \quad \text{where} \quad F(p) = \frac{p}{1 - 2(1 - \theta)p} \quad (6.124)$$

which are satisfied when

$$\theta \geq \frac{1}{2} \quad \text{and} \quad \lambda \leq \sqrt{2\theta - 1} \quad (6.125)$$

See Problem 6.15. Notice here that as θ becomes large, λ can also be large as well, meaning that, for a given grid spacing h , the time step k can become large as well, leading to fewer recursion steps relative to the standard scheme (6.54), for which λ is bounded strictly by 1. This is the justification often given for using implicit methods []—though each step in the recursion becomes more complex, computationally, than that of an explicit scheme, one may get away with fewer such steps. For sound synthesis, this is probably the wrong way of looking at the relative merits of explicit and implicit methods. The time step k is fixed, from the outset, from the audio sample rate. A better interpretation is that implicit methods, *operating at a given sample rate* can allow for much lower numerical dispersion than an explicit scheme *operating at the same sample rate*. Some examples of this will be provided at later stages in this book, particularly in the case of bar vibration in §7.1.5, but a good question is: in the context of musical sound synthesis, is the reduction of numerical dispersion of enough perceptual importance to justify the use of more complex schemes? The answer, however, can only be dependent on the specifics of the problem at hand. The reduction of numerical dispersion, however, is but one advantage of implicit methods; another is easier control over stability, one which will be seen to be of great utility in dealing with sound synthesis methods for nonlinear problems. See, for example, Chapter 8 for some examples of the use of implicit methods in the case of nonlinear string vibration.

One potential complication of the use of implicit methods is in setting boundary conditions—it can be far less clear how to do this properly when the update values at the boundaries themselves are coupled to other unknowns. Again, energy analysis is invaluable in this respect. For scheme (6.120), it is useful to employ identity (5.12), and rewrite the scheme as

$$\left(1 + \frac{(1 - \theta)h^2}{2} \delta_{xx}\right) \delta_{tt}u = \gamma^2 \delta_{xx}u \quad (6.126)$$

Considering the scheme defined over the semi-infinite domain $\mathcal{D} = \mathbb{Z}^+$, taking an inner product with

$\delta_t.u$ gives, as usual,

$$\delta_{t+}\mathfrak{h} = \mathfrak{b} \quad (6.127)$$

where in this case, one has

$$\mathfrak{h} = \mathfrak{t} + \mathfrak{v} \quad \text{with} \quad \mathfrak{t} = \frac{1}{2} \|\delta_{t-}u\|_{\mathbb{Z}^+}^2 - \frac{(1-\theta)h^2}{4} \|\delta_{t-}\delta_{x+}\|_{\mathbb{Z}^+}^2 \quad \mathfrak{v} = \frac{\gamma^2}{2} \langle \delta_{x+}u, e_{t-}\delta_{x+}u \rangle_{\mathbb{Z}^+} \quad (6.128)$$

and

$$\mathfrak{b} = -\gamma^2 (\delta_t.u_0) (\delta_{x-}u_0) + \frac{(1-\theta)h^2}{2} (\delta_t.u_0) (\delta_{x-}\delta_{tt}u_0) \quad (6.129)$$

The scheme will be conservative when \mathfrak{b} vanishes, and it is easy to see from the above expression that this will be true when, for example, $u_0 = 0$, or $\delta_{x-}u_0 = 0$, which are the same discrete Dirichlet and Neumann conditions as for the simple scheme (6.54).

6.4 Modal Synthesis

The modal solution presented here indicates a straightforward means of performing synthesis. Once the modal frequencies and functions have been determined (and in the present case of the 1D wave equation under the simple types of boundary terminations presented here, these are expressible in closed form), and, for a given pair of initial conditions $u(x, 0)$ and $u_t(x, 0)$, the weighting coefficients α_p and β_p have been determined, through a Fourier series decomposition such as (6.50), the solution at an output position such as $x = x_o$ may be written directly as

$$u(x_o, t) = \sum_p (\alpha_p \cos(p\pi\gamma t) + \beta_p \sin(p\pi\gamma t)) \sin(p\pi x_o) \quad (6.130)$$

Leaving aside the issue of the infinite sum above, which must be truncated in any implementation, the main computational work will be in the generation of the time-dependent sinusoids above. Though this could, of course, be done using simple table look-up operations, as discussed in the context of abstract sound synthesis in §1.1.3, another way of proceeding, provided one has already calculated the modal frequencies and shapes, is express the wave equation as a set of ODEs, as per

$$U(x_o, t) = \sum_p U_p(x_o, t) \quad \text{with} \quad \frac{d^2 U_p(x_o, t)}{dt^2} = -\omega_p^2 U_p(x_o, t) \quad (6.131)$$

The solution may be expressed as a parallel combination of uncoupled simple harmonic oscillators.

In discrete time, for each SHO in the sum above, one could use a simple scheme such as (??), but there is no reason not to use the exact scheme (??)

6.5 Loss

As in the case of the simple harmonic oscillator, it is straightforward to extend the 1D wave equation through the addition of a term which models loss, or dissipation:

$$u_{tt} = \gamma^2 u_{xx} - 2\sigma_0 u_t \quad (6.132)$$

where σ is a non-negative constant. The loss term may be viewed as resulting from a variety of physical phenomena, including radiation and internal losses [].

The dispersion analysis applied to (6.132) yields the characteristic polynomial

$$s^2 + 2\sigma_0 s + \gamma^2 \beta^2 = 0 \quad (6.133)$$

with roots

$$s_{\pm} = -\sigma_0 \pm \sqrt{\sigma_0^2 - \gamma^2 \beta^2} \quad (6.134)$$

For $\beta \geq \sigma_0/\gamma$, and writing $s_{\pm} = \sigma \pm \omega$, one has

$$\sigma = -\sigma_0 \quad \omega = \sqrt{\gamma^2\beta^2 - \sigma_0^2} \quad (6.135)$$

Thus the rate of loss is frequency independent and equal to σ_0 , but the frequency is of a more complex form than in the lossless case. In fact, it should be clear that the phase and group velocity are now wavenumber dependent, and thus wave propagation is dispersive. An exact traveling wave solution to the lossy wave equation is thus ruled out (though it should be noted that lossy traveling wave solutions do exist for a slight variant of (6.132)—see Problem (6.10)). In general, however, the degree of dispersion is quite small for realistic values of σ_0 . See Problem 6.9. It should also be noted that for small values of the wavenumber, i.e., when $\beta < \sigma_0/\gamma$, the solutions s_{\pm} will be purely real, indicating the presence of non-propagating solutions, and a cutoff wavenumber.

The 60 dB decay time T_{60} for the lossy wave equation, which as noted above is frequency independent under low loss conditions, may be written as

$$T_{60} = \frac{6 \ln(10)}{\sigma_0} \quad (6.136)$$

and is a useful global parameter in a sound synthesis simulation.

From an energetic standpoint, the situation is again very similar to that of the simple harmonic oscillator with loss—see §3.5. Considering the lossy 1D wave equation defined over the infinite domain $\mathcal{D} = \mathbb{R}$, one has, taking the inner product with u_t ,

$$\langle u_t, u_{tt} \rangle_{\mathbb{R}} = \gamma^2 \langle u_t, u_{xx} \rangle_{\mathbb{R}} - 2\sigma_0 \|u_t\|_{\mathbb{R}}^2 \quad (6.137)$$

or

$$\frac{d\mathfrak{H}}{dt} = -2\sigma_0 \|u_t\|_{\mathbb{R}}^2 \leq 0 \quad \implies \quad 0 \leq \mathfrak{H}(t_2) \leq \mathfrak{H}(t_1) \quad \text{for } t_1 \leq t_2 \quad (6.138)$$

where \mathfrak{H} is defined as in the lossless case. Thus the energy is non-negative and decreases monotonically. It is not difficult to show that the boundary conditions discussed in §6.1.9 do not interfere with this strict dissipation—the energetic behaviour of the wave equation is seen to be separated between contributions over the domain interior, and at the boundary. See Problem 6.17.

6.5.1 Finite Difference Scheme

A simple extension of scheme (6.54) to the case of linear loss is given by

$$\delta_{tt}u = \gamma^2 \delta_{xx}u - 2\sigma_0 \delta_t u \quad (6.139)$$

or, in the form of a recursion, as

$$u_i^{n+1} = \frac{2}{1 + \sigma_0 k} \left((1 - \lambda^2) u_i^n - \lambda^2 (u_{i-1}^n + u_{i+1}^n) \right) - \frac{1 - \sigma_0 k}{1 + \sigma_0 k} u_i^{n-1} \quad (6.140)$$

See Programming Exercise 6.7.

The characteristic polynomial for this scheme is

$$(1 + \sigma_0 k)z + (4\lambda^2 \sin^2(\beta h/2) - 2) + (1 - \sigma_0 k)z^{-1} = 0 \quad (6.141)$$

It is simple to show (see Problem 6.13) that the roots are bounded by unity for all wavenumbers β , under the condition (6.63), as for the scheme (6.54) for the lossless wave equation. Thus the addition of a centered difference approximation to the loss term does not affect the stability results obtained in the lossless case.

Just as in the continuous case, energy analysis of the scheme (6.139) leads directly to

$$\delta_{t+\mathfrak{h}} = -2\sigma_0 \|\delta_t u\|_{\mathbb{Z}}^2 \leq 0 \quad (6.142)$$

where the discrete energy is defined just as in the lossless case. Thus the energy of the scheme

(6.139) is monotonically decreasing.

6.6 A Comparative Study: Physical Modeling Sound Synthesis Methods

Though the attention in this chapter has focussed mainly on difference schemes for the wave equation, various other well-known synthesis strategies have been briefly described as well—lumped methods, in §??, digital waveguides in §??, and modal synthesis in §??. This is a good moment to step back to evaluate the relative merits of these techniques.

6.6.1 Accuracy

The accuracy with which a numerical solution is computed can have direct perceptual ramifications on sound output—numerical dispersion can lead to an undesirable mistuning of modes. This effect is illustrated in the present case of the 1D wave equation in §??. In general, all synthesis methods based on direct time/space simulation will be prey to such effects, including finite difference methods and lumped networks. It is possible, however, to design methods for which this effect is reduced, often greatly—see §?? for some examples—but such methods are always more computationally intensive, in that sense that they either must be implicit (requiring more computational work), or involve a wider stencil (requiring more care to be taken when setting boundary conditions). Another remedy, and not one which is advised here due to computational expense, is to make use of a simple scheme at a higher sample rate.

Modal methods, on the other hand, possess the great advantage of allowing an exact solution to be calculated, at least for linear and time invariant problems. Once the problem has been decomposed into modes, each of which behaves as a simple harmonic oscillator, exact numerical methods exist, as have been detailed in §?? and §??. There is thus no mistuning of modes in a properly executed modal simulation. In this very special case of the 1D wave equation, digital waveguides also generate an exact solution, but, unlike the case of modal methods, this property does not extend to virtually any other system.

6.6.2 Memory Use

As should be clear from the discussion at various points in this chapter, memory requirements scale directly with the number of degrees of freedom of the system—this is true for finite difference methods (including lumped methods), modal methods, as well as digital waveguides. For the simple finite difference scheme (6.54), this number is bounded above by $2f_s/\gamma$, which corresponds to two grid functions worth of “state.” The number remains the same in the case of a lumped network representation, which yields the same difference scheme, as discussed in §6.1.1. It is indeed possible to develop scheme which requires less memory, but at the expense of reduced solution bandwidth, and increased numerical dispersion. The memory requirement for a digital waveguide, which is none other than a special case of scheme (6.54), is exactly $2f_s/\gamma$. For modal methods, the number of degrees of freedom required to fill the available audio bandwidth will again be $2f_s/\gamma$, corresponding to f_s/γ modes, each necessarily represented in discrete time by two units of memory. Again, one may attempt to reduce the number of modes in synthesis output, but only at the expense of reduced accuracy, and bandwidth. At the level of memory required to represent solution state, all methods are thus equivalent. It is also important to note that this requirement does *not* depend significantly

on the choice of boundary conditions—this is perhaps most easily understood in the case of finite difference schemes, for which boundary conditions are generally set locally, but even in the case of a modal representation, it is true that the modal density of an LTI problem remains the same, regardless of boundary conditions, at least in the high frequency limit. The addition of loss, generally a very small effect in musical sound synthesis, leads only to a negligible change in these memory requirements.

Modal methods, however, can incur rather heavy costs if the modal frequencies and functions are not expressible in closed form (due, perhaps, to more complex boundary conditions). The storage of the frequency values will amount to another f_s/γ worth of memory. If one were to attempt full storage of all f_s/γ , at f_s/γ separate spatial locations (i.e., with enough values to allow resolution of all frequencies up to the Nyquist), the storage requirement balloons to f_s^2/γ^2 . This will be necessary if one is interested in, say, in changing output locations dynamically.

6.6.3 Precomputation

Computation before run-time for simple finite difference schemes, including lumped methods and waveguides is minimal. For modal methods, if the modal frequencies and shapes are not known, these must be determined through the solution of an eigenvalue problem. This can be quite costly, and potentially a very serious bottleneck if the number of modes is large. Even in the simple

6.6.4 Operation Count

The run-time operation counts per time step, for scheme (6.54) (including the lumped network representation) scales directly with f_s/γ . For more complex schemes, such as those given in §??, the operation count will be higher, but will still scale with this same number. For modal methods,

Waveguides are the great exception—the operation count is $O(1)$ per time step

6.6.5 Stability

6.6.6 Generality and Flexibility

6.7 Problems

Problem 6.1 Consider the 1D wave equation, defined over the finite interval $\mathcal{D} = [0, 1]$, with fixed boundary conditions $u = 0$ at both boundary points. Show that under these conditions, bounds (6.41) may be improved to

$$|u(x_0, t)| \leq \frac{1}{2\gamma} \sqrt{2\mathfrak{H}(0)} \quad \implies \quad \|u\|_{\mathcal{D}} \leq \frac{1}{2\gamma} \sqrt{2\mathfrak{H}(0)} \quad (6.143)$$

Problem 6.2 Consider the 1D wave equation, defined over the semi-infinite interval $\mathcal{D} = \mathbb{R}^+$. Show that when the lossy boundary condition (6.28) is employed at $x = 0$, with $\alpha = \gamma$, left-going traveling waves are completely absorbed. (You may substitute the traveling wave solution (6.15) directly into the boundary condition.)

Problem 6.3 Consider the 1D wave equation, defined over the semi-infinite interval $\mathcal{D} = \mathbb{R}^+$. Show that the nonlinear boundary condition

$$u_x(0, t) = \alpha (u_t(0, t))^2 \operatorname{sgn}(u_t(0, t)) \quad \text{for} \quad \alpha > 0 \quad (6.144)$$

is dissipative, i.e.,

$$\frac{d\mathfrak{H}}{dt} \leq 0 \quad (6.145)$$

If this condition is generalized this to

$$u_x(0, t) = f(u_t(0, t)) \quad (6.146)$$

for some function f , what property must f satisfy in order that the system remain dissipative?

Problem 6.4 Consider again the 1D wave equation, defined now over the finite interval $\mathbb{U} = [0, 1]$. Under the boundary conditions

$$u(0, t) = 0 \quad \text{and} \quad u_x(1, t) = \alpha u(1, t) \quad (6.147)$$

for some constant α , show that the modal frequencies ω must satisfy the equation

$$\frac{\omega}{\gamma\alpha} = \tan\left(\frac{\omega}{\gamma}\right) \quad (6.148)$$

and as such, must be determined numerically. As ω becomes large, show that the density of modal frequencies approaches that of the wave equation under fixed-fixed, free-free or fixed-free conditions.

Problem 6.5 Caution: This is a difficult, but instructive problem, in that some rather more advanced energy-based techniques for the analysis of numerical boundary conditions are introduced.

Consider scheme (6.54), defined over $\mathcal{D} = \mathbb{Z}^+$, under the generalized Neumann boundary condition at $l = 0$:

$$\alpha\delta_{x-}u_0 + (1 - \alpha)\delta_x.u_0 = 0 \quad (6.149)$$

Using the results and definition in Problem 5.7 in the previous chapter, show that, by taking an ϵ inner product of the scheme with $\delta_t.u$, the following energy balance may be obtained:

$$\delta_{t+}\mathfrak{h} = -\gamma^2(\delta_t.u_0)((2\epsilon - 1)\delta_{x-}u_0 + 2(1 - \epsilon)\delta_x.u_0) \quad (6.150)$$

where

$$\mathfrak{h} = \frac{1}{2}\left(\|\delta_t.u\|_{\mathbb{Z}^+}^{\prime,\epsilon}\right)^2 + \frac{\gamma^2}{2}\langle\delta_{x+}u, e_{t-}\delta_{x+}u\rangle_{\mathbb{Z}^+} \quad (6.151)$$

Thus the boundary condition above may be considered conservative under an epsilon inner product with $\alpha = 2\epsilon - 1$.

Go further, and show that

$$\mathfrak{h} \geq \frac{1}{2}\left(\|\delta_t.u\|_{\mathbb{Z}^+}^{\prime,\epsilon}\right)^2 - \frac{\lambda^2}{2}\left(\|\delta_t.u\|_{\mathbb{Z}^+}^{\prime,\epsilon}\right)^2 \quad (6.152)$$

Show that, for non-negativity, beyond CFL condition, the condition $\lambda \leq \sqrt{2\epsilon}$ must also be satisfied, and, given the relationship between α and ϵ , determine the range of α for which this additional condition interferes with the CFL bound.

Problem 6.6 For the lossy numerical boundary condition (6.90) for scheme (6.54) for the 1D wave equation, applied at grid point $l = 0$, show that when $\lambda = 1$ and $\alpha = \gamma$, the condition reduces to $u_0^{n+1} = u_1^n$. See also Programming Exercise 6.3.

Problem 6.7 Consider, instead of the numerical radiation condition (6.97), the condition

$$\delta_x.u_0 = \alpha_1\delta_t.u_0 + \alpha_s\mu_t.u_0 \quad (6.153)$$

in conjunction with scheme (6.54) over the domain $\mathcal{D} = \mathbb{Z}^+$. Show that the conserved numerical energy will be of the form $\mathfrak{h} + \mathfrak{h}_b$, but with a different definition of \mathfrak{h}_b from that which appears in (6.99). Show that in this case, that the energy remains non-negative (and the scheme stable) under the CFL condition (6.63), without any interfering condition for the boundary.

Problem 6.8 As was seen in §6.1.10, from the bound on the energy for scheme (6.54) defined over the semi-infinite domain $\mathcal{D} = \mathbb{Z}^+$, one may deduce that, under any conservative boundary condition at $l = 0$, and under stability condition (6.63), one has the conditions

$$\|\delta_t.u\|_{\mathbb{Z}^+} \leq \sqrt{2\mathfrak{h}} \quad \|\delta_{x+\mu_t}u\|_{\mathbb{Z}^+} \leq \frac{\sqrt{2\mathfrak{h}}}{\gamma} \quad (6.154)$$

The first condition on its own serves to bound the rate of growth of the solution. If the condition at $l = 0$ is of Dirichlet type, i.e., $u_0 = 0$, then it is possible to determine a bound on u_l^n itself. This is

a little trickier than in the continuous case, and requires a good number of the bounds and identities supplied in the previous chapter.

(a) First, show that if $u_0 = 0$, one may write $u_l = \langle 1, \delta_{x+f} \rangle_{[0,l-1]}$, and, thus, using the Cauchy-Schwartz inequality, that

$$|u_l| \leq \sqrt{lh} \|\delta_{x+u}\|_{[0,l-1]} \leq \sqrt{lh} \|\delta_{x+u}\|_{\mathbb{Z}^+} \quad (6.155)$$

(b) Using identity (2.7d), and the triangle inequality (5.19b), show furthermore that

$$|u_l| \leq \sqrt{lh} \left(\|\delta_{x+\mu_t-u}\|_{\mathbb{Z}^+} + \frac{k}{2} \|\delta_{x+\delta_t-u}\|_{\mathbb{Z}^+} \right) \leq \sqrt{lh} \left(\|\delta_{x+\mu_t-u}\|_{\mathbb{Z}^+} + \frac{k}{h} \|\delta_{t-u}\|_{\mathbb{Z}^+} \right) \quad (6.156)$$

(c) Use the separate bounds on $\|\delta_{t-u}\|_{\mathbb{Z}^+}$ and $\|\mu_t-\delta_{x+u}\|_{\mathbb{Z}^+}$ given at the beginning of this problem to arrive at a final bound on $|u_l|$ in terms of γ, λ, l and the energy \mathfrak{h} .

Problem 6.9 Consider the lossy wave equation (6.132), defined over the unit interval $\mathcal{D} = \mathbb{U}$, with fixed boundary terminations.

(a) Assume allowed wavenumbers β_i are of the form $\beta_i = i\pi$, and determine, from the dispersion relation (??), and expression for the frequency, in Hertz, of the i th mode.

(b) Assuming that $\gamma = 100$, and $T_{60} = 3$, calculate the frequency of the first mode (these parameters are typical of a musical string). By how many cents does it deviate from the frequency of the first mode in the lossless case?

Problem 6.10 The wave equation with a loss term exhibits a (very slight) degree of dispersion, so clearly a strict traveling wave solution does not exist, ruling out an efficient solution by digital waveguides. But dispersionless wave propagation can extend to the case of loss, if a slightly different model is employed. Consider the so-called distortionless transmission line equation:

$$u_{tt} = \gamma^2 u_{xx} - 2\sigma_0 u_t - \sigma_0^2 u \quad (6.157)$$

defined over the real line $\mathcal{D} = \mathbb{R}$.

(a) Perform a dispersion analysis of this equation, by inserting a test solution of the form $u(x, t) = e^{st+j\beta x}$, and derive the characteristic equation relating s and β .

(b) Solve for the roots $s_{\pm}(\beta)$ of the characteristic equation, and express the real and imaginary parts of the roots separately, i.e., write $s_{\pm}(\beta) = \sigma(\beta) + j\omega(\beta)$. Show that, given this expression for $\omega(\beta)$, the phase and group velocity are again constant.

(c) Show, by direct substitution, that damped traveling wave solutions of the form $e^{-\sigma_0 t} u_+(x - \gamma t)$ and $e^{-\sigma_0 t} u_-(x + \gamma t)$ satisfy the distortionless transmission line equation (6.157), for arbitrary distributions u_+ and u_- .

(d) By taking an inner product of (6.157) over $\mathcal{D} = \mathbb{R}$, find an expression for the energy \mathfrak{H} , and show that it is monotonically decreasing. Furthermore, find a bound on $\|u\|_{\mathbb{R}}$ in terms of the initial energy $\mathfrak{H}(0)$.

Problem 6.11 Consider the following approximation to the distortionless wave equation (6.157) presented in the previous problem:

$$\delta_{tt} u = \gamma^2 \delta_{xx} u - 2\sigma_0 \delta_t u - \sigma_0^2 \mu_{t+} \mu_{t-} u \quad (6.158)$$

defined over the infinite domain $\mathcal{D} = \mathbb{Z}$.

(a) Write the characteristic polynomial corresponding to scheme (6.158). Show that the stability condition on λ , the Courant number, is unchanged from the Courant Friedrichs Lewy condition (6.63).

(b) Write the explicit recursion in the grid function u_l^n corresponding to the compact operator form given above. How does the computational footprint of the scheme simplify when a choice of $\lambda^2 = 1 - \sigma_0^2 k^2 / 4$ is chosen?

(c) Show that, under the assumption that $\lambda^2 = 1 - \sigma_0^2 k^2 / 4$, a lossy discrete traveling wave solution exists. That is, given a traveling wave decomposition $u_l^n = u_l^{(+),n} + u_l^{(-),n}$, and where the wave variables $u_l^{(+),n}$ and $u_l^{(-),n}$ are updated according to $u_l^{(+),n} = r u_{l-1}^{(+),n-1}$, $u_l^{(-),n} = r u_{l+1}^{(-),n-1}$, for some constant r , show that the scheme (6.158) is implied, and find the value of r in terms of σ_0 and k .

(d) The rate of loss in the scheme (6.158) is not quite equivalent to that of the partial differential equation (6.157). Show that, in fact, scheme (6.158) solves equation (6.157) exactly, with σ_0 replaced by a slightly different value σ_0^* , and find the values of σ_0^* in terms of σ_0 and k .

Problem 6.12 Given the dispersion relation (6.66) for scheme (6.54) for the wave equation, show by direct calculation that the phase and group velocities, as functions of frequency ω , are equal to

$$v_\phi(\omega) = \frac{h\omega}{2\sin^{-1}(\sin(\omega k/2)/\lambda)} \quad v_g(\omega) = \frac{\gamma\sqrt{1 - \sin^2(\omega k/2)/\lambda^2}}{\cos(\omega k/2)} \quad (6.159)$$

Show that in the limit as ω or k approaches zero, the limiting value of both these expressions is γ , the wave speed for the continuous time/space wave equation.

Problem 6.13 Show that the roots of the characteristic equation (??) for the scheme (??) for the lossy wave equation is satisfied under the condition $\lambda \leq 1$, by using the result (??).

Problem 6.14 Considering the five-point stencil scheme (6.113) for the wave equation, show the stability conditions (6.118) and (6.119), through the determination of the maximum and minimum values of the function $F(p)$ over the unit interval $p \in [0, 1]$, as in (6.117).

Problem 6.15 Considering the implicit scheme (6.120) for the wave equation, show the stability conditions (6.125), through the determination of the maximum and minimum values of the function $F(p)$ over the unit interval $p \in [0, 1]$, as in (6.124).

Problem 6.16 Show that the two implicit schemes below for the 1D wave equation,

$$(\theta + (1 - \theta)\mu_x) \delta_{tt}u = \gamma^2 \delta_{xx}u \quad \delta_{tt}u = \gamma^2 (\phi + (1 - \phi)\mu_t) \delta_{xx}u \quad (6.160)$$

in the free parameters θ and ϕ , respectively, are equivalent when $\theta = 1 + \lambda^2(1 - \phi)$. The identity (5.11) may be of use here.

Problem 6.17 Extend the energy analysis of the wave equation with loss (6.132) to the domain $\mathcal{D} = \mathbb{R}^+$, and show that under Dirichlet, Neumann or the lossy boundary condition (6.28) at $x = 0$, the stored energy \mathfrak{H} will be non-increasing.

6.8 Programming Exercises

Exercise 6.1 Consider the 1D wave equation, under a sharp “plucked” excitation. In the most extreme case, one may write $u(x, 0) = \delta(x - x_i)$, $u_t(x, 0) = 0$, where $\delta(x - x_i)$ is a Dirac delta function at location $x = x_i$. The resulting solution should thus be, from the traveling wave decomposition, $u(x, t) = 0.5(\delta(x - x_i - \gamma t) + \delta(x - x_i + \gamma t))$ —in other words, it should be composed of two spikes traveling in opposite directions away from the excitation point.

In the scheme (6.54), and assuming x_i to lie directly on a grid point at l_i , where $x_i = l_i h$, the first-order initializing strategy of (6.57) prescribes $u_{l_i}^0 = u_{l_i}^1 = 1/h$. Modify the code example given in §A.4 such that the scheme is initialized in the above way, and examine the results after a few time steps. For simplicity, choose values of the wave speed and the sample rate such that the Courant condition may be satisfied with equality (i.e., $\lambda = 1$). Does the result correspond to the traveling wave solution given above?

Instead of the above setting for the initial conditions, choose $u_{l_i}^0 = 1/h$, $u_{l_i-1}^1 = u_{l_i+1}^1 = 1/2h$. Are the results more in line with the traveling wave solution? Can you explain what is going wrong with initialization strategy (6.57)?

Exercise 6.2 Modify the Matlab code provided in §A.4 so that, in the main loop, the numerical energy \mathfrak{h}^n is calculated. Now, with Dirichlet conditions at both ends of the domain (which is \mathbb{U}_N , the energy function is

$$\mathfrak{h} = \frac{1}{2} \|\delta_{t-}\|_{\mathbb{U}_N}^2 + \frac{\gamma^2}{2} \langle \delta_{x+}u, e_{t-} \delta_{x+}u \rangle_{\mathbb{U}_N} \quad (6.161)$$

Verify that it is constant to machine accuracy (in double precision floating point, about 15 decimal places), and plot the variation in the energy from the initial energy \mathfrak{h}^0 , normalized by the initial energy as a function of time step.

Exercise 6.3 Consider the 1D wave equation, under the lossy condition (6.28) at the left end, and under a fixed Dirichlet condition at the right end. Modify the code example given in §A.4 such that the numerical boundary condition (6.90) is employed at the left end. After appropriate initialization, show that when $\alpha = \gamma$, and $\lambda = 1$, the lossy boundary condition is perfectly absorbing (i.e., after a finite number of time steps, the state u of your recursion should consist entirely of zeros).

Exercise 6.4 Consider the 1D wave equation, under the radiation condition (6.28) at the left end, and a Dirichlet condition at the right end. Modify the boundary condition in the code example in §A.4 two separate ways: according to the numerical condition (6.97), and that given in (6.153). For simplicity, consider the case for which $\alpha_1 = 1$ (i.e., the lossless case). For a given choice of γ and the sample rate f_s , experiment with both terminations for a variety of values of α_2 and λ . Demonstrate that for the boundary condition (6.97), the condition (6.102) is sufficient for stability, but not necessary—i.e., show that for some values of λ which violate this condition, no instability is observed. How close to necessary is this condition, for a given value of α_2 ? Demonstrate, to yourself, that the numerical boundary condition (6.153) is always stable as long as the CFL condition (6.63) is observed.

Exercise 6.5 Consider the scheme (6.54), under two different types of numerical Neumann condition, applied at both ends of the domain:

$$\delta_{x-}u_0 = \delta_{x+}u_N = 0 \quad \delta_x.u_0 = \delta_x.u_N = 0 \quad (6.162)$$

Create a Matlab script which calculates the numerical modal frequencies for this scheme, under both termination types. (To do this, you will need to find the matrix operator \mathbf{D}_{xx} corresponding to the difference operator δ_{xx} , under the above boundary conditions. It will be of size $(N + 1) \times (N + 1)$. You may then calculate the modal frequencies as per (6.106).) Experiment with different values of λ , and compare the resulting frequencies to the exact modal frequencies, from (6.51). Which type of condition leads to more accurate numerical frequencies?

Exercise 6.6 Modify the code example in §A.4, such that there are two output locations $x_o^{(1)}$ and $x_o^{(2)}$, which move sinusoidally according to

$$x_o^{(1)}(t) = 1/3 + A_{sw} \sin(2\pi f_{sw}t) \quad x_o^{(2)}(t) = 2/3 - A_{sw} \sin(2\pi f_{sw}t) \quad (6.163)$$

where A_{sw} , the sweep-depth satisfies $0 \leq A_{sw} < 1/3$, and where f_{sw} is the sweep frequency, generally under about 5 Hz—both should appear as additional global parameters. In addition, implement truncation, linear, and cubic interpolation, where the choice is determined by an input flag. (The interpolation operations are described in §5.2.4.) Your code should write output to a two-column array, which can be played directly in stereo in Matlab. Experiment with your code for different fundamental frequencies $f_0 = \gamma/2$ —is linear interpolation sufficient to eliminate audible clicks and time-varying low-passing effects?

Exercise 6.7 Extend the code example in §A.4 to cover scheme (6.139) for the wave equation with a linear loss term. The decay time T_{60} should appear as an extra global parameter.

Chapter 7

Bars and Linear Strings

In this chapter, the finite difference methods discussed in the last chapter with regard to the 1D wave equation are extended to the more musically interesting case of bars and stiff strings. Linear, or low-amplitude vibration is characteristic of a variety of instruments, including xylophones and marimbas, and, to a lesser extent, string instruments such as acoustic guitars and pianos, which can, under certain conditions, exhibit nonlinear effects (this topic will be discussed in detail in Chapter 8). Linear time-invariant systems such as these, and the associated numerical methods are still amenable to frequency domain analysis, which can be very revealing with regard to effects such as dispersion (leading to perceived inharmonicity). In the numerical setting, frequency domain analysis is useful in obtaining simple necessary stability conditions, and in dispersion which results from discretization error; energy analysis allows the determination of suitable numerical boundary conditions, and in the extension to problems involving spatial variation.

In §7.1, the idealized thin bar model is introduced—while too simple to use for sound synthesis purposes, many important ideas relating to discretization may be dealt with here in a compact manner. The more realistic stiff string model, incorporating stiffness, tension, and loss terms, is dealt with in §7.2. Various musical features are discussed in the following sections, namely the coupling to a hammer model, in §7.5, multiple strings in §7.6, and the preparation of strings in §7.7. Finally, in §7.9.2, the more complex case of strings of varying density, and bars of variable cross-section is introduced; this is the first instance, in this book, of a system with spatial variation, and as such, von Neumann type stability analysis for finite difference schemes no longer applies directly, though energetic methods remain viable.

References for this chapter include: [152, 19, 51, 119, 120, 10, 20, 271, 116, 154, 77, 284, 137]

7.1 The Ideal Uniform Bar

Transverse vibrations of a thin bar¹ of uniform material properties and cross-section are covered by many authors. In the lossless case, the defining PDE is

$$\rho A u_{tt} = -EI u_{xxxx} \quad (7.1)$$

where ρ is the material density of the medium, A is the cross-sectional area, E is Young's modulus, and I is a parameter often referred to as the bar moment of inertia, which depends on the geometry

¹The use of the terms "bar," "beam" and "rod" is somewhat confused in the literature; they will be used interchangeably here, though in some cases, when one discusses a beam, it is assumed that one is referring to its transverse vibration alone, which is indeed the case in most parts of this book.

of the bar cross-section. In this section, it is assumed that these quantities are constants. More will be said about these quantities in §7.9.2, which deals with bars of varying cross-section.

In spatially non-dimensionalized form, where as for the case of the wave equation, x has been replaced by x/L , for some constant L with dimensions of length, (7.1) looks like

$$u_{tt} = -\kappa^2 u_{xxxx} \quad (7.2)$$

where κ , the stiffness parameter, is defined by

$$\kappa = \frac{EI}{\rho L^4} \quad (7.3)$$

Again, one may note that several parameters, E , ρ , I , A , and a characteristic length L , have been replaced by a single parameter κ , allowing for a substantial reduction in the parameter space faced by the programmer and eventual user (composer). This is, like the 1D wave equation, a second-order (in time) PDE in a single variable, and as such, one again requires two initial conditions, generally $u(x, 0)$ and $u_t(x, 0)$. See §6.1.3 for a discussion of these conditions—in short, one may imagine plucking or striking a stiff bar in exactly the same way as for a string described by the wave equation.

This is the simplest possible model of transverse vibration of a uniform stiff medium; many other more complex models are available—see, e.g., [111, 178]. If the medium is not thin, then the linear Timoshenko theory of beam vibration may be applicable—see Problem ?? for a presentation of this system, as well as a discussion of its frequency domain behaviour and range of applicability in musical acoustics.

7.1.1 Dispersion

Information about propagation speeds may be obtained through the insertion of a test solution of the form $u(x, t) = e^{st+j\beta x}$ into (7.2). One arrives at the characteristic polynomial equation

$$s^2 + \kappa^2 \beta^4 = 0 \quad (7.4)$$

which has solutions

$$s_{\pm} = \pm j\kappa\beta^2 \quad \implies \quad \omega_{\pm} = \pm\kappa\beta^2 \quad (7.5)$$

Thus, the frequency ω of a wave component scales as the square of the wavenumber β —this is in sharp contrast to the case of the wave equation, for which ω scales directly with β . Such behaviour, which is referred to as dispersive, may be more readily understood with regard to the phase and group velocity (see §5.1.2), which, in this case, are given by

$$v_{\phi}(\beta) = \frac{\omega}{\beta} = \kappa\beta \quad v_g(\beta) = \frac{d\omega}{d\beta} = 2\kappa\beta \quad (7.6)$$

In other words, components of large wavenumber (or short wavelength) travel faster than those of small wavenumber (long wavelength). These velocities may also be written in terms of frequency ω , using (7.5), as

$$v_{\phi}(\omega) = \sqrt{\kappa\omega} \quad v_g(\omega) = 2\sqrt{\kappa\omega} \quad (7.7)$$

See Figure 7.1 for an illustration of dispersive behaviour in a thin bar. As one might expect, this will have a very pronounced effect (namely inharmonicity) on the natural frequencies of the bar, when the domain is limited to be finite. See §7.1.3. One might note also that the phase and group velocity are unbounded—in the limit as β becomes large (or the wavelength small), these speeds become infinite. This is a consequence of the assumptions underlying this simple (Euler-Bernoulli) bar vibration model, and was later corrected by Timoshenko—see Problem ?. It is important to note that in the case of the bar of infinite extent, only real wavenumbers β may be considered (if

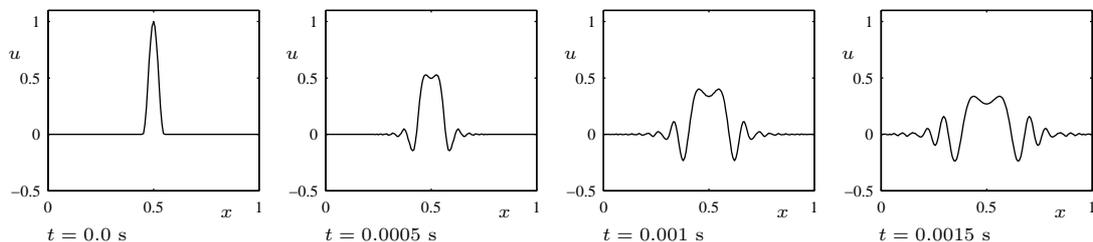


Figure 7.1: Time evolution of the solution to the ideal bar equation (7.1), with $\kappa = 1$, under a plucked initial condition, with $u_0 = c_{rc}(x)$, and $x_0 = 0.5$, $c_0 = 1$ and $x_{hw} = 0.05$. Successive snapshots of the solution profile are shown at the times indicated, over the interval $x \in [0, 1]$.

β has an imaginary part, the test solution will necessarily be unbounded). But when the domain is limited to be finite, such solutions are indeed possible, and have a complicating effect on modal analysis (not to mention synthesis!). See §7.1.3.

7.1.2 Energy Analysis and Boundary Conditions

Energy techniques have already been presented with respect to the 1D wave equation in detail in §6.1.8. In the simple case, for which the bar equation is defined over the real line, i.e., $\mathcal{D} = \mathbb{R}$, one may take the inner product of (7.2) with u_t , to get

$$\langle u_t, u_{tt} \rangle_{\mathbb{R}} + \kappa^2 \langle u_t, u_{xxxx} \rangle_{\mathbb{R}} = 0 \quad (7.8)$$

and, using integration by parts twice, according to (5.9),

$$\langle u_t, u_{tt} \rangle_{\mathbb{R}} + \kappa^2 \langle u_{txx}, u_{xx} \rangle_{\mathbb{R}} = 0 \quad \implies \quad \frac{d\mathfrak{H}}{dt} = 0 \quad (7.9)$$

where

$$\mathfrak{H} = \mathfrak{T} + \mathfrak{V} \quad \text{and} \quad \mathfrak{T} = \frac{1}{2} \|u_t\|_{\mathbb{R}}^2 \quad \mathfrak{V} = \frac{\kappa^2}{2} \|u_{xx}\|_{\mathbb{R}}^2 \quad (7.10)$$

Thus the bar equation (7.2), like the wave equation, possesses a conserved energy, and is thus lossless. As before, the energy is also non-negative, and bounds on solution growth may again be determined.

Lossless Boundary Conditions

In order to derive the usual boundary conditions [174, 111] for the ideal bar at a single end point, one may perform the above energy analysis for the case $\mathcal{D} = \mathbb{R}^+$. Again taking an inner product, one arrives immediately at

$$\langle u_t, u_{tt} \rangle_{\mathbb{R}^+} + \kappa^2 \langle u_t, u_{xxxx} \rangle_{\mathbb{R}^+} = 0 \quad (7.11)$$

Integration by parts now gives

$$\frac{d\mathfrak{H}}{dt} = \kappa^2 (u_t(0, t)u_{xxx}(0, t) - u_{tx}(0, t)u_{xx}(0, t)) \quad (7.12)$$

with \mathfrak{H} defined as in (7.10), but where inner products run over \mathbb{R}^+ instead of \mathbb{R} . The boundary terms vanish under the following pairs of conditions at $x = 0$:

$$u = u_x = 0 \quad \text{Clamped} \quad (7.13a)$$

$$u = u_{xx} = 0 \quad \text{Simply supported} \quad (7.13b)$$

$$u_{xx} = u_{xxx} = 0 \quad \text{Free} \quad (7.13c)$$

For any of the above choices, one again has energy conservation, i.e.,

$$\frac{d\mathfrak{H}}{dt} = 0 \quad (7.14)$$

As in the case of the wave equation, it should be clear that other lossless, or lossy boundary terminations are possible—these are the three which are of most practical use, and which, as a result, appear most frequently in the literature. See Problem 7.2 for more discussion of other types of termination. It should also be clear that such analysis may be extended directly to cover the case of the bar defined over a finite domain, such as $\mathcal{D} = \mathbb{U} = [0, 1]$, and the choice of any pair of conditions given in (7.13) at either end will lead to energy conservation.

7.1.3 Modes

The ideal bar, as defined in (7.2), is LTI, and thus an analysis in terms of modes is straightforward. One might expect that because the system is more generally LSI, that such analysis would be as simple as in the case of the 1D wave equation, yet even for this rudimentary system, complications begin to appear.

Again assume oscillatory solutions of the form $u(x, t) = U(x)e^{j\omega t}$, which must now satisfy

$$-\omega^2 U = -\kappa^2 \frac{d^4 U}{dx^4} \quad (7.15)$$

and thus solutions are of the form

$$U(x) = A \cos\left(\sqrt{\frac{\omega}{\kappa}}x\right) + B \sin\left(\sqrt{\frac{\omega}{\kappa}}x\right) + C \cosh\left(\sqrt{\frac{\omega}{\kappa}}x\right) + D \sinh\left(\sqrt{\frac{\omega}{\kappa}}x\right) \quad (7.16)$$

When the ideal bar is defined over $\mathcal{D} = \mathbb{U}$, the possible values of the frequency ω , as well as three of the four constants A , B , C and D , may be determined after two appropriate boundary conditions are set at either end of the bar (the fourth remains arbitrary, but may be set through normalization of the resulting modal function). For one very special choice, namely simply supported conditions at both ends (this is not particularly realistic in a musical acoustics setting), it is possible to show [174] that one will have $A = C = D = 0$, and a simple closed-form expression for the modal frequencies and shapes results:

$$\omega_{\pm p} = \pm \kappa \pi^2 p^2 \quad \text{and} \quad U_p(x) = \sin(p\pi x) \quad \text{for} \quad p = 1, \dots \quad (7.17)$$

Thus the frequencies increase as the square of the mode number. Note that in this case, one can obtain the expression for the p th modal frequency through substitution of the wavenumber $\beta = p\pi$ into the dispersion relation, given in (7.5).

For nearly all other choices of boundary condition (including most of the useful forms given in (7.13) above), there is not a closed-form expression for the modal frequencies ω_p , and, furthermore, the values of A , B , C and D will depend on the as-yet unknown values of ω_p . For example, under simply-supported conditions at $x = 0$ and clamped conditions at $x = 1$, for example, there results

$A = C = 0$, and the modal frequencies and shapes are given by

$$\tan\left(\sqrt{\frac{\omega_p}{\kappa}}\right) = \tanh\left(\sqrt{\frac{\omega_p}{\kappa}}\right) \quad \text{and} \quad U_p(x) = \sin\left(\sqrt{\frac{\omega_p}{\kappa}}x\right) - \frac{\sin\left(\sqrt{\frac{\omega_p}{\kappa}}\right) \sinh\left(\sqrt{\frac{\omega_p}{\kappa}}x\right)}{\sinh\left(\sqrt{\frac{\omega_p}{\kappa}}\right)} \quad (7.18)$$

While there is no conceptual or practical difficulty in determining the frequencies ω_p , it does become somewhat unwieldy from a synthesis point of view. The modal frequencies must first be determined, necessarily off-line, presumably through the solution of an eigenvalue problem or perhaps an iterative method applied directly to the implicit definition of frequency ω_p above. More difficult, here, is the question of storage. In addition to the memory required to store the frequencies themselves, it will be necessary to store the modal shapes. In this simple case, it is probably advisable to store the functions \sin , \cos , \sinh and \cosh , with sufficient resolution, as well as the proper settings for the coefficients A , B , C and D . For more general problems which are not LSI, however, these possibilities for simplification disappear—there will not be a convenient implicit definition of the modal frequencies, nor will there be a decomposition of the modal shapes into simpler components. All must necessarily be determined through the solution of an eigenvalue/eigenmode problem (again offline), and stored. For reference, plots of modal functions under all combinations of the boundary conditions given in (7.13) are shown in Figure 7.2.

Mode Density and Pitch

As was discussed with reference to the 1D wave equation at the end of §6.1.11, the determination of the distribution of modal frequencies is useful in that it gives a measure of the number of degrees of freedom necessary to describe the system. Considering again the frequencies which result under simply-supported conditions at either end of the bar in (7.17), the frequencies, in Hertz, will be given by $f_{\pm p} = \pm\kappa\pi p^2/2$, and thus the number N_m which fall in the audio band $f_p \in [-f_s/2, f_s/2]$ will be approximately

$$N_m = 2\sqrt{\frac{f_s}{\kappa\pi}} \quad (7.19)$$

This can be viewed as the number of degrees of freedom of the ideal bar. Again, it is important to note that this number will *not* depend in a significant way on the boundary conditions employed—any choice of boundary conditions will lead, in the high frequency limit, to the same density of modes. For example, in the case of the simply supported/clamped conditions discussed above, the implicit definition of the modal frequencies given in (7.18) leads to the following distribution of modal frequencies, in the limit as ω_p becomes large:

$$\omega_{\pm p} \approx \pm\kappa\pi^2\left(p + \frac{1}{4}\right)^2 \quad (7.20)$$

so that again, the number of degrees of freedom is given, approximately, by (7.19). The same is true of other choices of boundary condition—see Problem 7.1.

For the wave equation, which generally possesses a uniformly spaced series of harmonics, or modal frequencies, it is easy to associate a musical pitch with the system parameters: under fixed conditions, for example, one has a fundamental frequency of $f_0 = \gamma/2$. In the case of a bar, this is generally not so, due to inharmonicity. The lowest frequency will be in the range of $\kappa\pi/2$, with variations due to the type of boundary conditions. If this frequency is in the upper musical range (i.e., in say, the right half of the piano keyboard), then a listener will associate a pitch with this frequency. If it is low, however, the series of frequencies will be comparatively dense, and pitch assignment can

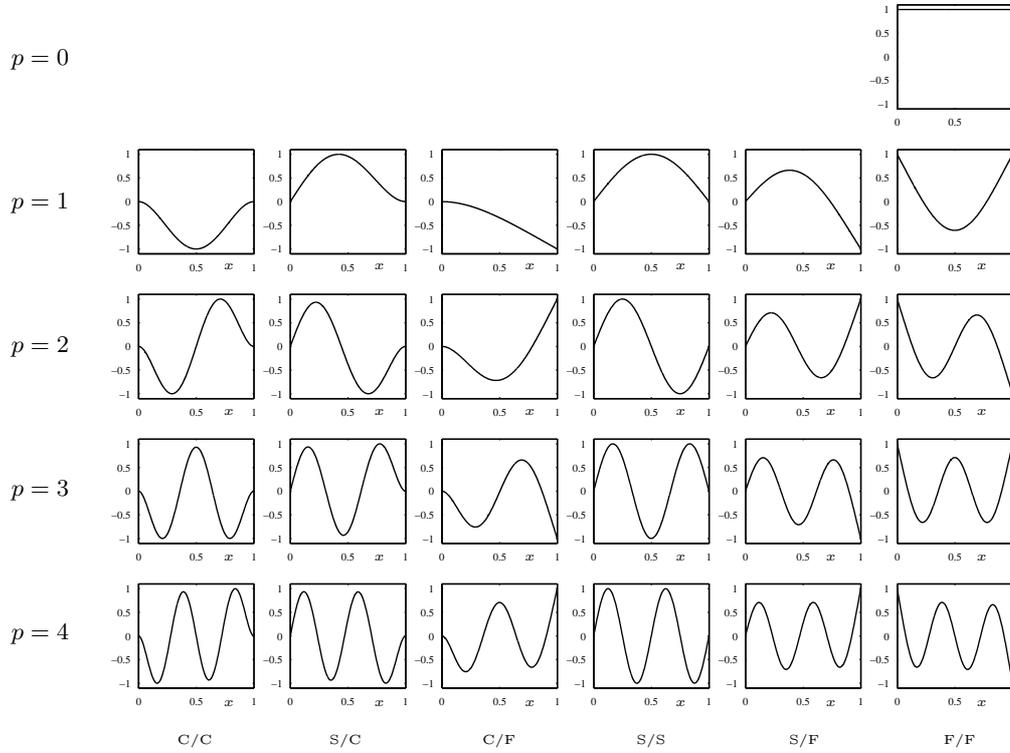


Figure 7.2: Plots of modal functions for the ideal bar, under various combinations of the boundary conditions of clamped (C), simply supported (S) and free (F) types given in (7.13), as indicated at bottom. The first four modal shapes are presented, in order of increasing frequency, down each column. In the case of free-free termination, an additional modal shape, of frequency zero, is also shown. The modal functions are all normalized so that they take on values between -1 and 1.

be difficult. Other factors, such as frequency-dependent loss, also intervene in this respect. For this reason, in implementations, it is perhaps simpler to leave the parameter κ as is, and ask the user to learn its perceptual significance.

7.1.4 A Simple Finite Difference Scheme

The most straightforward finite difference scheme for the ideal bar equation (7.2) may be written, in operator form, as

$$\delta_{tt}u = -\kappa^2 \delta_{xxxx}u \tag{7.21}$$

where the definition of δ_{xxxx} is given in (5.10b). When the operator notation is expanded out, the following two-step recursion results:

$$u_i^{n+1} = (2 - 6\mu^2)u_i^n + 4\mu^2 (u_{i+1}^n + u_{i-1}^n) - \mu^2 (u_{i-2}^n + u_{i+2}^n) - u_i^{n-1} \tag{7.22}$$

In order to be updated, the scheme requires access to values of the grid function two grid spacings away from the update point, and is parameterized by the single number μ ,

$$\mu \triangleq \kappa k / h^2 \quad (7.23)$$

μ plays a role similar to λ , the Courant number, which appears in schemes for the wave equation. Like the two step scheme for the wave equation (6.54), scheme (7.21) must be initialized with values of the grid function u at time steps $n = 0$ and $n = 1$. Scheme (7.21) is, formally, second order accurate in both time and space—unlike the case of scheme (6.54) for the wave equation, there is no special choice of the parameter μ which leads to improved accuracy.

von Neumann Analysis

Just as for schemes for the wave equation, one may employ the test function $u_i^n = z^n e^{j\beta l h}$, which, when inserted in scheme (7.21), gives the following characteristic polynomial in z :

$$z + (16\mu^2 \sin^4(\beta h/2) - 2) + z^{-1} = 0 \quad (7.24)$$

This equation possesses roots of magnitude unity under the condition

$$\mu \leq \frac{1}{2} \quad \implies \quad k \leq \frac{h^2}{2\kappa} \quad (7.25)$$

which is somewhat different from the condition (6.63) which arises in the simple scheme (6.54) for the 1D wave equation. Now, the time step is bounded by the square of the grid spacing, rather than the grid spacing itself. This condition has interesting ramifications with regard to computational complexity, as will be discussed next.

Degrees of Freedom and Computational Complexity

Supposing that the bar equation is defined over the unit interval $\mathcal{D} = \mathbb{U}$, the number of degrees of freedom of the scheme will be, as in the case of schemes for the wave equation, twice the number of grid points over this interval (scheme (7.21) is again a two-step scheme). Given that this number of grid points is approximately $1/h$, and that the sample rate is defined as $f_s = 1/k$, the number of degrees of freedom will be bounded as

$$N_{fd} \leq \sqrt{\frac{2f_s}{\kappa}} \quad (7.26)$$

which is very similar to the number of modes required to represent the bar to frequencies up to f_s , from (7.19), but not the same—it is greater, however, by a factor of $\sqrt{\pi/2}$ which is quite close to one, meaning that a modal representation and the finite difference scheme possess roughly the same number of degrees of freedom. This small discrepancy may be related to effects of numerical dispersion, as discussed in the next section.

It is crucial to note that, in a comparison between the wave equation and the bar equation, for which both possess the same first non-zero modal frequency (or “fundamental,” if one may indeed apply that term to an inharmonic system such as a bar), the bar will possess far fewer degrees of freedom—as a result, computations for dispersive systems such as bars are lighter than those for non-stiff systems such as strings. This may be counterintuitive, given that the bar equation itself *looks more complex* than the wave equation. Looks can be deceiving. Such an efficiency gain for stiff systems may be exploited to great benefit in, e.g., models of plate reverberation, which are far less computationally intensive than even 2D simulations of the wave equation. See §12.3.

Numerical Dispersion

When the stability condition (7.25) is satisfied, the roots of the characteristic equation are of the form $z = e^{j\omega k}$, and thus the characteristic polynomial (7.24) may be written as

$$\sin(\omega k/2) = \pm 2\mu \sin^2(\beta h/2) \quad (7.27)$$

which is analogous to the dispersion relation (7.5) for the continuous ideal bar equation. In contrast with scheme (6.54) for the wave equation, this scheme exhibits numerical dispersion, beyond the natural dispersion inherent in the bar equation itself, for any allowable choice of μ . See Figure 7.3 for plots of numerical phase and group velocity for different choices of μ —it remains true, however, that numerical dispersion is minimized as μ approaches the bound given in (7.25).

It is also true in this case that when the scheme is operating away from the stability condition (7.25) (i.e., when $\mu < 1/2$), there will be some loss in available bandwidth for the computed solution. The dispersion relation above possesses solutions for real ω only when

$$\omega \leq \frac{2}{k} \sin^{-1}(2\mu) \quad (7.28)$$

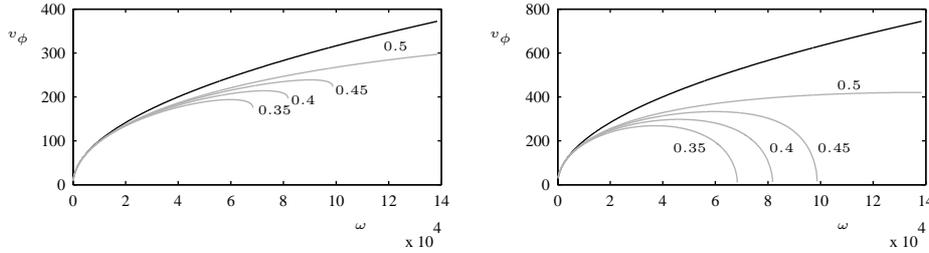


Figure 7.3: Numerical phase velocity v_ϕ (left) and group velocity v_g as functions of frequency ω for scheme (7.21) for the ideal bar equation, under different choices of the parameter μ , are plotted as grey lines. The curves for the continuous model system are plotted as black lines. In this case, the time step was chosen as $k = 1/44100$. Notice the numerical cutoff behaviour visible in the curves for values of $\mu < 1/2$.

Energy Conservation and Non-negativity

Just as for schemes for the wave equation, energy analysis is possible (and indeed quite straightforward) for scheme (7.21). Considering the problem defined over the infinite domain $\mathcal{D} = \mathbb{Z}$, and taking an inner product with the grid function $\delta_t u$, and using the double summation by parts identity (5.26) gives

$$\langle \delta_t u, \delta_{tt} u \rangle_{\mathbb{Z}} = -\kappa^2 \langle \delta_t u, \delta_{xxxx} u \rangle_{\mathbb{Z}} = -\kappa^2 \langle \delta_{xx} u, \delta_t \delta_{xx} u \rangle_{\mathbb{Z}} \quad (7.29)$$

or

$$\delta_{t+} \mathfrak{h} = 0 \quad \text{with} \quad \mathfrak{h} = \mathfrak{t} + \mathfrak{v} \quad \text{and} \quad \mathfrak{t} = \frac{1}{2} \|\delta_t u\|_{\mathbb{Z}}^2 \quad \mathfrak{v} = \frac{\kappa^2}{2} \langle \delta_{xx} u, e_t - \delta_{xx} u \rangle_{\mathbb{Z}} \quad (7.30)$$

The total difference (7.30) above is a statement of conservation of numerical energy for the scheme (7.21), and it thus follows that

$$\mathfrak{h}^n = \mathfrak{h}^0 \quad (7.31)$$

Just as in the case of schemes for the wave equation, in order to find conditions under which this energy is non-negative, one may write, using identity (2.22f) and inequality (5.28),

$$\mathfrak{h} = \frac{1}{2} \|\delta_t u\|_{\mathbb{Z}}^2 + \frac{\kappa^2}{2} \|\mu_{t-} \delta_{xx} u\|_{\mathbb{Z}}^2 - \frac{k^2 \kappa^2}{8} \|\delta_t - \delta_{xx} u\|_{\mathbb{Z}}^2 \quad (7.32)$$

$$\geq \left(\frac{1}{2} - \frac{2k^2 \kappa^2}{h^4} \right) \|\delta_t u\|_{\mathbb{Z}}^2 + \frac{\kappa^2}{2} \|\mu_{t-} \delta_{xx} u\|_{\mathbb{Z}}^2 \quad (7.33)$$

The energy is non-negative under condition (7.25), and under this condition, a bound on solution size for all n follows immediately. Again, energy analysis leads to the same stability condition as von Neumann analysis.

Numerical Boundary Conditions

Numerical boundary conditions may be derived, again as in the case of schemes for the wave equation. Here, however, the analysis is slightly more delicate—slightly different choices of inner product, as well as uses of summation by parts can give rise to different conserved energies, accompanied by distinct numerical boundary conditions. Consider now the scheme (7.21) defined over the semi-infinite domain \mathbb{Z}^+ . Taking the inner product with $\delta_t u$, and using the summation by parts identity from (5.25), gives

$$\delta_{t+} \mathfrak{h} = \mathfrak{b} \quad \text{with} \quad \mathfrak{h} = \mathfrak{t} + \mathfrak{v} \quad \text{and} \quad \mathfrak{t} = \frac{1}{2} \|\delta_t u\|_{\mathbb{Z}^+}^2 \quad \mathfrak{v} = \frac{\kappa^2}{2} \langle \delta_{xx} u, e_{t-} \delta_{xx} u \rangle_{\mathbb{Z}^+} \quad (7.34)$$

and

$$\mathfrak{b} = -\kappa^2 (\delta_t \delta_{x+} u_0) (\delta_{xx} u_0) + \kappa^2 (\delta_t u_0) (\delta_{x-} \delta_{xx} u_0) \quad (7.35)$$

Thus numerical energy conservation follows under the following choices of numerical boundary condition at the left end point of a domain:

$$u = \delta_{x+} u = 0 \quad \text{Clamped} \quad (7.36a)$$

$$u = \delta_{xx} u = 0 \quad \text{Simply supported} \quad (7.36b)$$

$$\delta_{xx} u = \delta_{x-} \delta_{xx} u = 0 \quad \text{Free} \quad (7.36c)$$

These conditions are, clearly, completely analogous to the conditions (7.13) for the ideal bar equation.

The particular form of summation by parts used above, namely that given in (5.25), leads to a simple, tractable expression for the numerical conserved energy, if $\mathfrak{b} = 0$ —tractable in the sense that non-negativity is simple to show, as the expression for the potential energy \mathfrak{v} is limited strictly to the *interior* of the problem domain. Just as in the case of the infinite domain discussed in §7.1.2, one has

$$\mathfrak{h} = \frac{1}{2} \|\delta_t u\|_{\mathbb{Z}^+}^2 + \frac{\kappa^2}{2} \|\mu_{t-} \delta_{xx} u\|_{\mathbb{Z}^+}^2 - \frac{k^2 \kappa^2}{8} \|\delta_t - \delta_{xx} u\|_{\mathbb{Z}^+}^2 \quad (7.37)$$

$$\geq \left(\frac{1}{2} - \frac{2k^2 \kappa^2}{h^4} \right) \|\delta_t u\|_{\mathbb{Z}^+}^2 + \frac{\kappa^2}{2} \|\mu_{t-} \delta_{xx} u\|_{\mathbb{Z}^+}^2 \quad (7.38)$$

and the stability condition (7.25) again leads to non-negativity.

It is important to note that if one had the other form of the summation by parts identity, i.e., that given in (5.24), the derived boundary conditions would be distinct, as would the expression for energy. In some cases, it is not so easy to show non-negativity as \mathfrak{v} may not be limited to the interior of the problem domain. See Problems ??–?? for some variations on energy analysis and numerical boundary conditions for the bar.

Implementation of Numerical Boundary Conditions

It is useful to show how numerical boundary conditions such as (7.36), presented in compact operator form, may be implemented directly in the scheme (7.21).

The clamped condition at $l = 0$ is the simplest to deal with: one may set, simply,

$$u_0 = u_1 = 0 \quad (7.39)$$

at each pass through the recursion. Thus only values u_l of the solution at locations $l = 2, 3, \dots$ need be computed; because the operator δ_{xxxx} is of width five, its value may be computed directly using the settings above for u_0 and u_1 .

The simply supported condition is slightly more complicated. They may be rewritten as

$$u_0 = 0 \quad u_{-1} = -u_1 \quad (7.40)$$

Now, values at the grid locations $l = 1, \dots$ must be updated; under the action of the δ_{xxxx} operator, the value at $l = 1$ will require access to a value at an imaginary grid point $l = -1$ (see §5.2.8). It is sufficient to examine the behaviour of the operator δ_{xxxx} at locations near the boundary, and employ both conditions above:

$$\delta_{xxxx}u_1 = \frac{1}{h^4}(u_3 - 4u_2 + 6u_1 - 4u_0 + u_{-1}) = \frac{1}{h^4}(u_3 - 4u_2 + 5u_1) \quad (7.41)$$

Such conditions may be incorporated into a matrix form of the operator δ_{xxxx} , as discussed in §5.2.8.

The settings for the free boundary condition are left to Problem 7.4.

7.1.5 An Implicit Scheme

The problem of numerical dispersion does not arise in the case of the most straightforward scheme for the 1D wave equation. In the case of the bar, however, the simple scheme presented in the previous section does indeed suffer from such dispersion. It was also noted that the number of degrees of freedom of the scheme, N_{fd} , is slightly larger than the number which follows from modal analysis. As will be shown in this section, these two undesirable phenomena are related, and may be addressed using implicit methods.

A simple implicit scheme for the 1D wave equation, dependent on a single free parameter θ , is presented in §6.3.2. Consider a similar implicit scheme for the ideal bar equation:

$$(\theta + (1 - \theta)\mu_x) \delta_{tt}u = -\kappa^2 \delta_{xxxx}u \quad (7.42)$$

In vector-matrix form, assuming as usual the column vector \mathbf{u}^n to contain the values of the grid function u_l^n , then the scheme may be written as

$$\mathbf{u}^{n+1} = \mathbf{A}^{-1} \mathbf{B} \mathbf{u}^n - \mathbf{u}^{n-1} \quad \text{with} \quad \mathbf{A} = \theta \mathbf{I} + (1 - \theta) \mathbf{M}_x. \quad \mathbf{B} = -\kappa^2 k^2 \mathbf{D}_{xxxx} + 2\mathbf{A} \quad (7.43)$$

where the various matrix operators incorporate boundary conditions. Again, a sparse (tridiagonal) linear system solution is required at each time step.

The von Neumann and energy methods yield the following stability conditions:

$$\theta \geq \frac{1}{2} \quad \mu \leq \frac{\sqrt{2\theta - 1}}{2} \quad (7.44)$$

See Problem 7.3.

It is instructive to plot the phase and group velocities for the scheme, for different values of the free parameter θ —see Figure 7.4. As usual, the scheme parameter μ is chosen as close to the stability bound as possible, so as to maximize the bandwidth of the scheme at a given sample rate $f_s = 1/k$, as discussed on page 168 with regard to the explicit scheme.

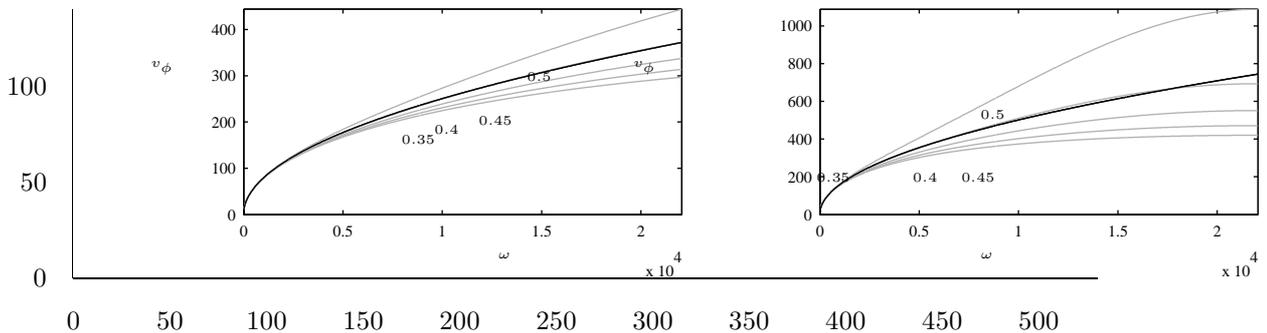


Figure 7.4: Numerical phase velocity v_ϕ (left) and group velocity v_g as functions of frequency f for scheme (7.42) for the ideal bar equation, under different choices of the parameter θ , are plotted as grey lines. The curves for the continuous model system are plotted as black lines. In this case, the time step was chosen as $k = 1/44100$, and μ is chosen so as to satisfy bound (7.44) with equality.

It is interesting to note that for the choice of $\theta \approx 0.7$, the numerical phase velocity is nearly a perfect match to the phase velocity of the model system. This choice of θ can be viewed as resulting from a consideration of the number of degrees of freedom, in the following way.

Supposing that the scheme is defined over the unit interval $\mathcal{D} = \mathbb{U}_N$, and that μ is chosen such as to satisfy the second of conditions (7.44) with equality, then the number of degrees of freedom for the scheme (i.e., twice the number of points on the grid) will be

$$N_{fd} = 2/h = 2\sqrt{\frac{\mu}{k\kappa}} = \sqrt{\frac{2}{k\kappa}} (2\theta - 1)^{1/4} = \sqrt{\frac{\pi}{2}} (2\theta - 1)^{1/4} N_m \quad (7.45)$$

where $N_m = 2\sqrt{\frac{1}{k\kappa\pi}}$ is the number of degrees of freedom resulting from modal analysis, as given in (7.19). In order to have $N_{fd} = N_m$, one must then choose

$$\theta = \frac{1 + 4/\pi^2}{2} = 0.7026\dots \quad (7.46)$$

The lesson here is that, for a parameterized scheme, the best numerical behaviour (i.e., the least numerical dispersion) is obtained when the number of degrees of freedom of the scheme matches that of the physical system. Also, note that, even though implicit schemes are marginally more costly, computationally, than explicit schemes, the number of degrees of freedom of the scheme, and thus the number of grid points is reduced from that of the explicit scheme under this choice of θ .

From a musical point of view, it is useful to look at the effect of the use of an implicit scheme on the calculated numerical modal frequencies. From the vector-matrix form (7.43), the set of such frequencies f_p , indexed by integer p , may be calculated as

$$f_p = \frac{2\pi}{k} \cos^{-1} \left(\frac{1}{2} \text{eig}_p(\mathbf{A}^{-1}\mathbf{B}) \right) \quad (7.47)$$

where eig_p signifies “ p th eigenvalue of.” Note that the modal frequencies for the explicit scheme may be calculated as well, by choosing $\theta = 1$. See Table 7.1.

From these results, one can immediately see that the accuracy of the implicit scheme is far superior to that of the explicit method—indeed, deviations of numerical frequencies from the exact values which are potentially audible for the explicit scheme are close to the threshold of inaudibility in the implicit case. What is more surprising, however, given the fidelity with which the numerical

Table 7.1: Comparison among modal frequencies of the ideal bar, under simply supported conditions, with a fundamental frequency of 100 Hz, and modal frequencies (as well as their cent deviations from the exact frequencies) of the explicit scheme (7.21), and the implicit θ scheme (7.42), with $\theta = 0.7026$, with a sample rate $f_s = 44100$ Hz, and where μ is chosen so as to satisfy (7.44) as close to equality as possible.

Mode number	Exact	Explicit		Implicit	
	Freq.	Freq.	Cent Dev.	freq.	Cent Dev.
1	100.0	99.7	-4.4	100.0	-0.8
2	400.0	396.0	-17.4	399.3	-3.0
3	900.0	880.2	-38.5	896.7	-6.4
4	1600.0	1539.1	-67.1	1590.3	-10.5
5	2500.0	2356.4	-102.4	2478.5	-15.0
6	3600.0	3313.5	-143.6	3560.0	-19.4

phase velocity of the implicit scheme with $\theta = 0.7026$ matches that of the model system, is that the numerical frequencies do not match even better! There are a couple of (admittedly extremely subtle) factors which account for this. The main one is that, due to the truncation of the number of grid points to an integer, it becomes difficult to satisfy the stability condition (7.44) exactly, so that μ undershoots its maximum value by a very small amount. The other is that the numerical boundary condition, here chosen as being of simply supported type, is approximate as well, and not accounted for by dispersion analysis of the scheme over the infinite domain. One could thus go much further toward developing even more refined designs, but, for musical purposes, the above accuracy would appear to be sufficient for most purposes. The curious may be interested in Programming Exercise ??.

The lesson here is:

Rule of Thumb # 2

For a parameterized implicit numerical method, the best numerical behaviour (i.e., the least numerical dispersion) is obtained when the stability condition is satisfied with equality, and when the free parameter(s) are chosen such that the number of degrees of freedom of the scheme matches that of the model system, at a given sample rate.

7.2 Stiff Strings

In some musical instruments (in particular the piano), strings may be subject to a restoring force due not only to applied tension, but also to stiffness. Though the effect of stiffness is usually small relative to that of tension, it does lead to an audible inharmonicity. The PDE model of such a string has been presented by many authors, and, when spatially non-dimensionalized, has the form

$$u_{tt} = \gamma^2 u_{xx} - \kappa^2 u_{xxxx} \quad (7.48)$$

and may be viewed as a combination of the wave equation (6.6) and that of the ideal bar (7.2). It is, again, a second-order (in time) PDE, and as such requires two initial conditions, of the form mentioned in, e.g., §6.1.3.

The dispersion relation for the stiff string, as defined by (7.48), is

$$s^2 + \gamma^2 \beta^2 + \kappa^2 \beta^4 = 0 \quad (7.49)$$

implying that

$$\omega_{\pm} = \pm \beta \sqrt{\gamma^2 + \kappa^2 \beta^2} \quad (7.50)$$

and thus the phase and group velocity will be

$$v_{\phi} = \sqrt{\gamma^2 + \kappa^2 \beta^2} \quad v_g = \frac{\gamma^2 + 2\kappa^2 \beta^2}{\sqrt{\gamma^2 + \kappa^2 \beta^2}} \quad (7.51)$$

Writing these expressions in terms of ω is tedious, but the results are as expected: for low frequencies, the velocities approach γ , and for high frequencies, the velocity of the ideal bar—see Figure ?? . “High” and “low” here now must be taken with regard to the size of κ relative to γ . See Figure 7.5.

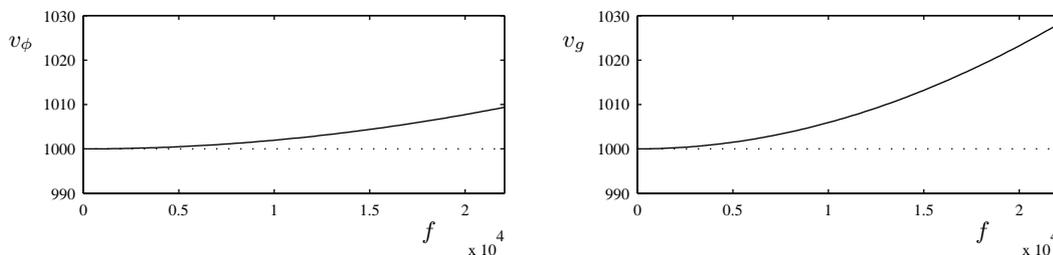


Figure 7.5: Phase velocity v_{ϕ} (left) and group velocity v_g (right) as functions of frequency f for the stiff string equation (7.48), with $\gamma = 1000$, and $\kappa = 1$. The velocities of the system without stiffness are plotted as dotted lines.

7.2.1 Energy and Boundary Conditions

The energy analysis of the stiff string, defined in (7.48) is similar to that performed for the wave equation, and for the ideal bar. Considering the stiff string defined over the semi-infinite domain $\mathcal{D} = \mathbb{R}^+$, and taking the inner product with u_t , one arrives at

$$\frac{d\mathfrak{H}}{dt} = \mathfrak{B} \quad \text{with} \quad \mathfrak{H} = \mathfrak{T} + \mathfrak{V} \quad (7.52)$$

and

$$\mathfrak{T} = \frac{1}{2} \|u_t\|_{\mathbb{R}^+}^2 \quad \mathfrak{V} = \frac{\gamma^2}{2} \|u_x\|_{\mathbb{R}^+}^2 + \frac{\kappa^2}{2} \|u_{xx}\|_{\mathbb{R}^+}^2 \quad (7.53)$$

$$\mathfrak{B} = -\gamma^2 u_t(0, t) u_x(0, t) + \kappa^2 u_t(0, t) u_{xxx}(0, t) - \kappa^2 u_{tx}(0, t) u_{xx}(0, t) \quad (7.54)$$

Here, the potential energy is merely the sum of the potential energies of the wave equation and the bar. The boundary term \mathfrak{B} vanishes, implying losslessness, under the clamped and simply supported conditions (7.13a) and (7.13b), but the free termination (which is not often employed in the musical acoustics of strings) must be modified to

$$u_{xx}(0, t) = \kappa^2 u_{xxx}(0, t) - \gamma^2 u_x(0, t) = 0 \quad (7.55)$$

7.2.2 Modes

Modal analysis is also similar. In the simplest case, namely simply supported conditions, the modal frequencies and functions are of the form

$$U_p(x) = \sin(p\pi x) \quad \omega_p = \gamma p \pi \sqrt{1 + \frac{\kappa^2 \pi^2}{\gamma^2} p^2} \quad (7.56)$$

When the stiffness parameter κ is small, as in the case of most strings, the frequencies, are sometimes written [95], in Hertz, in terms of the fundamental frequency $f_0 = \gamma/2$ for the ideal lossless string, an inharmonicity factor B :

$$f_p = f_0 p \sqrt{1 + B p^2} \quad \text{with} \quad B = \frac{\kappa^2 \pi^2}{\gamma^2} \quad (7.57)$$

Thus the harmonic series becomes progressively detuned, to a degree dependent on B .

Again, for other choices of boundary conditions, the modal functions must be expressed in terms of trigonometric and hyperbolic trigonometric functions, and there is not a closed-form expression for the modal frequencies. Under clamped conditions, for low stiffness, the modal frequencies are changed little from the above series. The number of degrees of freedom will thus be, approximately,

$$N_m = \sqrt{\frac{2}{B} \left(\sqrt{1 + B f_s^2 / f_0^2} - 1 \right)} \quad (7.58)$$

7.2.3 Finite Difference Schemes

Finite difference schemes for (7.48) may be arrived at using the same construction techniques as in the case of the 1D wave equation and the ideal bar equation. The simplest is certainly the following:

$$\delta_{tt} u = \gamma^2 \delta_{xx} u - \kappa^2 \delta_{xxxx} u \quad (7.59)$$

The von Neumann stability condition for this scheme is

$$\lambda^2 + 4\mu^2 \leq 1 \quad (7.60)$$

and is expressed in terms of the Courant number $\lambda = \gamma k/h$, used in schemes for the 1D wave equation, and the parameter $\mu = \kappa k/h^2$, used in the case of schemes for the ideal bar. See Problem ???. It is perhaps easier to rewrite this in a form involving k and h directly:

$$h^4 - \gamma^2 k^2 h^2 - 4\kappa^2 k^2 \geq 0 \quad \rightarrow \quad h \geq h_{min} = \sqrt{\frac{\gamma^2 k^2 + \sqrt{\gamma^4 k^4 + 16\kappa^2 k^2}}{2}} \quad (7.61)$$

Thus, again, for a given k (or sample rate f_s , there is a minimum grid spacing h .

One implicit generalization of the now-familiar θ type will be, then,

$$(\theta + (1 - \theta)\mu_x) \delta_{tt} u = \gamma^2 \delta_{xx} u - \kappa^2 \delta_{xxxx} u \quad (7.62)$$

and stability analysis leads to the condition

$$\theta \geq \frac{1}{2} \quad \text{and} \quad \lambda^2 + 4\mu^2 \leq 2\theta - 1 \quad \rightarrow \quad h \geq h_{min} = \sqrt{\frac{\gamma^2 k^2 + \sqrt{\gamma^4 k^4 + 16\kappa^2 k^2 (2\theta - 1)}}{2(2\theta - 1)}} \quad (7.63)$$

The number of degrees of freedom N_{fd} will thus be, in the ideal case where the bound above is satisfied with equality,

$$N_{fd} = \frac{2}{h_{min}} \quad (7.64)$$

In vector matrix form, the scheme may be written as

$$\mathbf{u}^{n+1} = \mathbf{A}^{-1} \mathbf{B} \mathbf{u}^n - \mathbf{u}^{n-1} \quad (7.65)$$

with

$$A = \theta \mathbf{I} + (\theta - 1) \mathbf{M}_x, \quad \mathbf{B} = 2\mathbf{I} + k^2 (\gamma^2 \mathbf{D}_{xx} - \kappa^2 \mathbf{D}_{xxxx}) \quad (7.66)$$

As expected, the numerical phase and group velocities for the implicit scheme depend heavily on the choice of θ ; it is more difficult to get a good match, in this case, across the entire frequency spectrum, especially under low stiffness conditions. See Figure 7.6. Still, one can do fairly well, as long as θ is chosen properly. As before, the best fit occurs when the number of degrees of freedom of the scheme matches that of the model problem, and one can indeed determine in this way a value for θ , dependent now on γ , κ and f_s —see Problem 7.5 and Programming Exercise 7.1.

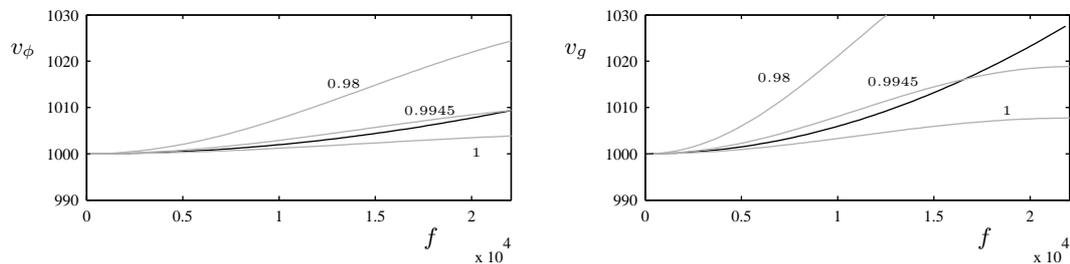


Figure 7.6: Phase velocity v_ϕ (left) and group velocity v_g (right) as functions of frequency f for the stiff string equation (7.48), with $\gamma = 1000$, and $\kappa = 1$, in black. Plotted in grey are the numerical phase velocities of the parameterized scheme (7.62), under different choices of the free parameter θ , as indicated. The sample rate is $f_s = 44100$ Hz, and the grid spacing h is chosen at the stability limit (7.63).

7.3 Frequency Dependent Loss

Real strings exhibit a rather complex loss characteristic—the various components do not all decay at the same rate. In particular, loss generally increases with frequency, leading to a sound with a wide-band attack portion, and decaying to a few harmonics. One model of frequency-dependent loss in linear strings is given by the following PDE:

$$u_{tt} = \gamma^2 u_{xx} - \kappa^2 u_{xxxx} - 2\sigma_0 u_t + 2\sigma_1 u_{txx} \quad (7.67)$$

The term with coefficient σ_0 is the familiar loss term which was encountered earlier in the case of the wave equation in §6.5. On its own, it gives rise to a bulk frequency-independent loss (i.e., a constant T_{60} at all frequencies). The extra term, with coefficient σ_1 is one means of modelling frequency-dependence of the loss characteristic. It is worth noting that this is not the original model of frequency-dependent loss (which employed a third time derivative term) which was proposed by Ruiz [220], and then later used in piano synthesis models by various authors [52].

As usual, the simplest way of analyzing the behaviour of the PDE is through frequency-domain techniques. Again using a test solution of the form $e^{st+j\beta x}$, the characteristic equation for (7.67) is

$$s^2 + 2(\sigma_0 + \sigma_1 \beta^2) s + \gamma^2 \beta^2 + \kappa^2 \beta^4 = 0 \quad (7.68)$$

which has roots

$$s_{\pm} = -\sigma_0 - \sigma_1\beta^2 \pm \sqrt{(\sigma_0 + \sigma_1\beta^2)^2 - (\gamma^2\beta^2 + \kappa^2\beta^4)} \quad (7.69)$$

For small σ_0 and σ_1 , the term under the root above is negative for all wavenumbers over a very small cutoff, and one may write, using $s = \sigma + j\omega$,

$$\sigma(\beta) = -\sigma_0 - \sigma_1\beta^2 \quad \omega(\beta) = \sqrt{\gamma^2\beta^2 + \kappa^2\beta^4 - (\sigma_0 + \sigma_1\beta^2)^2} \quad (7.70)$$

The loss $\sigma(\beta)$ thus increases monotonically as a function of wavenumber, taking the value of σ_0 in the limit as β approaches zero.

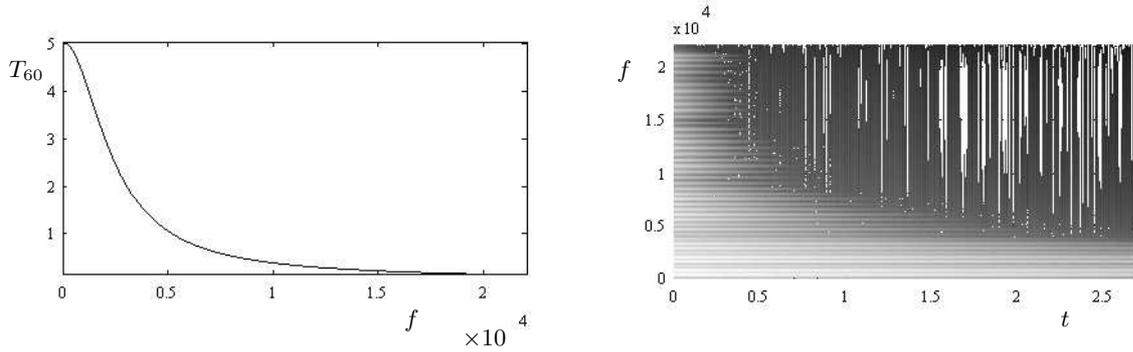


Figure 7.7: (a) A typical plot of T_{60} versus frequency, for system (7.67), with $T_{60} = 5$ at 100 Hz, and $T_{60} = 3$ at 2000 Hz. (b) Spectrogram of sound output, for a stiff string with $\gamma = 800$, and $K = 8$, with the loss parameters as specified above.

Practical Settings for Decay Times

It can be somewhat difficult to get a grasp on the perceptual significance of the parameter σ_1 . To this end, it is worth rewriting the expression for loss $\sigma(\beta)$ as a function of frequency, i.e., $\sigma(\omega)$. From the expression for $\omega(\beta)$ given above, it is reasonable to assume that when the loss parameters σ_0 and σ_1 are small (as they should be in any reasonable model of a bar or string used for musical purposes), that one may write

$$\omega(\beta) \approx \sqrt{\gamma^2\beta^2 + \kappa^2\beta^4} \quad \Rightarrow \quad \beta^2 = \frac{-\gamma^2 + \sqrt{\gamma^4 + 4\kappa^2\omega^2}}{2\kappa^2} \quad (7.71)$$

which further implies that one may write $\sigma(\omega)$ as

$$\sigma(\omega) = -\sigma_0 - \sigma_1\xi(\omega) \quad \text{with} \quad \xi(\omega) = \frac{-\gamma^2 + \sqrt{\gamma^4 + 4\kappa^2\omega^2}}{2\kappa^2} \quad (7.72)$$

With two parameters available to set the decay profile for the string, one could then specify T_{60} decay times at frequencies ω_1 and ω_2 , and set σ_0 and σ_1 , from the above formula, as

$$\sigma_0 = \frac{6 \ln(10)}{\xi(\omega_2) - \xi(\omega_1)} \left(\frac{\xi(\omega_2)}{T_{60}(\omega_1)} - \frac{\xi(\omega_1)}{T_{60}(\omega_2)} \right) \quad \sigma_1 = \frac{6 \ln(10)}{\xi(\omega_2) - \xi(\omega_1)} \left(-\frac{1}{T_{60}(\omega_1)} + \frac{1}{T_{60}(\omega_2)} \right) \quad (7.73)$$

See Figure 7.7, showing a typical loss profile for a two-parameter model, and a spectrogram of an output signal. For this model, for which loss increases monotonically with frequency, one must choose $T_{60}(\omega_2) \leq T_{60}(\omega_1)$ when $\omega_2 > \omega_1$. If one were interested in more a more complex decay profile, one could introduce additional terms in the model PDE, perhaps involving higher spatial

derivatives. See Problem 7.6. This, however, introduces a good deal more complexity into the analysis of boundary conditions, stability, and will inevitably lead to a higher computational cost, though it may be useful in some problems, such as bar-based percussion instruments [116, 77].

7.3.1 Energy and Boundary Conditions

The energy analysis of system (7.67) is similar to that of previous systems, except for the treatment of the frequency-dependent loss term. Again examining the system over the domain $\mathcal{D} = \mathbb{R}^+$, and taking an inner product with u_t leads to the energy balance

$$\frac{d\mathfrak{H}}{dt} = \mathfrak{B} - \sigma_0 \|u_t\|_{\mathbb{R}^+}^2 - \sigma_1 \|u_{xt}\|_{\mathbb{R}^+}^2 \quad \text{with} \quad \mathfrak{H} = \mathfrak{T} + \mathfrak{W} \quad (7.74)$$

and where \mathfrak{T} and \mathfrak{W} are defined as for the lossless stiff string, and where the boundary term must be modified to

$$\mathfrak{B} = -\gamma^2 u_t(0, t) u_x(0, t) + \kappa^2 (u_t(0, t) u_{xxx}(0, t) - u_{tx}(0, t) u_{xx}(0, t)) - 2\sigma_1 u_t(0, t) u_{tx}(0, t) \quad (7.75)$$

Clamped and simply supported terminations, of the form given in (??), may again be viewed as lossless, as \mathfrak{B} vanishes. The free condition is more complex: one possibility is to write:

$$u_{xx} = 0 \quad \kappa^2 u_{xxx} - \gamma^2 u_x - 2\sigma_1 u_{tx} = 0 \quad (7.76)$$

Under any of these conditions, one has, just as in the case of the wave equation with loss, a monotonic decrease in energy, i.e.,

$$\frac{d\mathfrak{H}}{dt} \leq 0 \quad \Rightarrow \quad \mathfrak{H}(t_2) \leq \mathfrak{H}(t_1) \quad \text{for} \quad t_2 \geq t_1 \quad (7.77)$$

7.3.2 Finite Difference Schemes

There are various ways of approximating the mixed derivative term in order to arrive at a finite difference scheme; two slightly different approximations were discussed briefly in §5.2.3. Two schemes are given, in operator form, as

$$\delta_{tt}u = \gamma^2 \delta_{xx}u - \kappa^2 \delta_{xxxx}u - 2\sigma_0 \delta_t u + 2\sigma_1 \delta_{t-} \delta_{xx}u \quad (7.78a)$$

$$\delta_{tt}u = \gamma^2 \delta_{xx}u - \kappa^2 \delta_{xxxx}u - 2\sigma_0 \delta_t u + 2\sigma_1 \delta_t \delta_{xx}u \quad (7.78b)$$

Though these differ only in the treatment of the final term, there is a huge difference, in terms of implementation, between the two: scheme (7.78a) is explicit, and scheme (7.78b) is implicit. Note also, the use of a non-centered difference operator δ_{t-} in the explicit scheme, rendering it formally only first-order accurate, though as this term is quite small, it will not have an appreciable effect on accuracy as a whole, nor, even on sound output.

Because the implicit scheme employs centered operators only, stability analysis is far simpler—in fact, the loss terms have no effect on the stability condition, which remains the same as in the lossless case.

The implicit scheme (7.78a) may be written in matrix form as

$$\mathbf{A}u^{n+1} + \mathbf{B}u^n + \mathbf{C}u^{n-1} \quad (7.79)$$

with

$$\mathbf{A} = (1 + \sigma_0 k)\mathbf{I} - \frac{\sigma_1 k}{h^2} \mathbf{D}_{xx} \quad \mathbf{B} = \lambda^2 \mathbf{D}_{xx} - \mu^2 \mathbf{D}_{xxxx} \quad \mathbf{C} = (1 - \sigma_0 k)\mathbf{I} + \frac{\sigma_1 k}{h^2} \mathbf{D}_{xx} \quad (7.80)$$

The reader may wish to look at the implementation code provided in §A.8. The matrix formulation makes for a remarkably compact recursion, at least in Matlab.

7.4 Coupling with Bow Models

As a first example of a coupling with an excitation mechanism, consider the case of a bow model, as has been introduced in §4.3.1. In this earlier section, the coupling of a bow model with a lumped mass-spring system was considered. The distributed case is not much more difficult to deal with.

Suppose, first, that the bow acts on an ideal string, described by the wave equation defined over the unit interval $\mathcal{D} = \mathbb{U}$, at a location x_i , where x_i can be time-varying, i.e., $x_i = x_i(t)$. If the spatial distribution of the bow is point-like, then the system is described, in non-dimensional form, by

$$u_{tt} = \gamma^2 u_{xx} - \delta(x_i) F_B \phi(u_t(x_i) - v_B) \quad (7.81)$$

where $F_B = F_B(t) \geq 0$ is the bow force/total string mass, $v_B = v_B(t)$ is the bow velocity. ϕ is some characteristic assumed to satisfy properties (4.20) and (4.25). $\delta(x_i)$, it is to be recalled, is a spatial Dirac delta function centered at x_i .

An inner product with u_t over \mathbb{U} gives the energy balance

$$\frac{d\mathfrak{H}}{dt} = \mathfrak{B} - u_t(x_i) F_B \phi(u_t(x_i) - v_B) = \mathfrak{B} - (u_t(x_i) - v_B) F_B \phi(u_t(x_i) - v_B) - v_B F_B \phi(u_t(x_i) - v_B) \quad (7.82)$$

where \mathfrak{H} is, as before, the Hamiltonian for the ideal string, as given in (6.20) and taken over the interval $\mathcal{D} = \mathbb{U}$, \mathfrak{B} is a boundary term, and the other two terms represent dissipation in the bow mechanism, and power supplied by the bow, respectively. Under conservative or lossy boundary conditions, one has $\mathfrak{B} \leq 0$, and thus

$$\frac{d\mathfrak{H}}{dt} \leq -v_B F_B \phi(u_t(x_i) - v_B) \leq |F_B v_B| \quad \Rightarrow \quad \mathfrak{H}(t) \leq \mathfrak{H}(0) + \int_0^t |F_B v_B| dt' \quad (7.83)$$

Thus the dynamics of the bowed string are bounded in terms of the initial conditions of the string, and energy provided by the bow.

Finite Difference Scheme

The case of the bow-string system, for which the bowing point may be moving during playing, requires a somewhat subtle treatment—as one might expect, interpolation and spreading functions play a rather important role in the case of a difference scheme acting over a grid.

Consider the following difference scheme analogous to (7.81):

$$\delta_{tt} u = \gamma^2 \delta_{xx} u - J(x_i^n) F_B \phi(I(x_i^n) \delta_t u - v_B) \quad (7.84)$$

where here, $F_B = F_B^n \geq 0$ and $v_B = v_B^n$ are now time series (perhaps sampled from data captured from a controller of some type). x_i^n is a time series representing the bow position at time step n . Notice now the use of the interpolator $I(x_i^n)$, used in order to interpolate a value of string velocity at the bowing point, and the spreading function $J(x_i^n)$, which distributes the bow force over several neighboring grid points on the string—see §5.2.4. The order of both is unspecified for the moment.

An inner product with $\delta_t u$ over the domain \mathbb{U}_N yields the energy balance

$$\delta_{t+} \mathfrak{h} = \mathfrak{b} - \langle \delta_t u, J(x_i^n) \rangle_{\mathbb{U}_N} F_B \phi(I(x_i^n) \delta_t u - v_B) \quad (7.85)$$

where \mathfrak{h} and \mathfrak{b} are the Hamiltonian and boundary term for this particular scheme for the 1D wave equation, as discussed in §6.2.5 and §6.2.6.

For arbitrary interpolation and spreading operators, one cannot go further and find a stability condition. If, however, one chooses $I = I_q(x_i^n)$ and $J = J_{l,q}(x_i^n)$ of the same order, q , then using

(5.29), one may write

$$\delta_{t+\mathfrak{h}} = \mathfrak{b} - F_B(I(x_i^n)\delta_{t,u})\phi(I(x_i^n)\delta_{t,u} - v_B) \quad (7.86)$$

$$= \mathfrak{b} - F_B(I(x_i^n)\delta_{t,u} - v_B)\phi(I(x_i^n)\delta_{t,u} - v_B) - F_B v_B \phi(I(x_i^n)\delta_{t,u} - v_B) \quad (7.87)$$

In direct analogy with the continuous system, under lossless or dissipative numerical boundary conditions $\mathfrak{b} \leq 0$, one arrives, finally, at

$$\delta_{t+\mathfrak{h}} \leq |F_B v_B| \quad \Rightarrow \quad \mathfrak{h}^n \leq \mathfrak{h}^0 + \sum_{p=0}^{n-1} k |F_B^p v_B^p| \quad (7.88)$$

Again, one may bound the behaviour of the scheme in terms of the known time series F_B and v_B . Notice in particular that the motion of the bowing position has no influence on stability.

Scheme (7.84) is implicit, and just as in the case of the bowed mass, there is the potential for non-uniqueness of numerical solutions. Taking an inner product with $J_{l,q}(x_i^n)$, and again using (5.29) gives

$$I_q(x_i^n)\delta_{tt}u = \gamma^2 I_q(x_i^n)\delta_{xx}u - \|J(x_i^n)\|_{\mathbb{V}}^2 F_B \phi(I(x_i^n)\delta_{t,u} - v_B) \quad (7.89)$$

This equation may be solved uniquely for the relative velocity $I(x_i^n)\delta_{t,u} - v_B$ under the condition

$$k \leq -\frac{2}{\max(F_B) \max(\|J_q\|_{\mathbb{V}}^2) \min(\phi')} \quad \Rightarrow \quad \lambda \leq \frac{2\gamma}{-\max(F_B) \min(\phi')} \quad (7.90)$$

which holds by virtue of the bound (5.30) on the interpolant J_q , and when $\min(\phi') \leq 0$, which is normally the case. The condition above must be satisfied in addition to the usual CFL condition (6.63) for this simple scheme for the wave equation.

Helmholtz Motion

The earliest studies of bowed string dynamics, by Helmholtz and Raman in particular, focussed on a phenomenological description of the motion, often referred to as Helmholtz motion—see [169] or [96] and the references therein for more on the history of the development of such models. For the sake of completeness, it is worth discussing this motion here, very briefly.

The principle is not extremely different from the case of the bowed mass, as presented briefly in §4.3.1.

Moving Bowing Point

When the bowing point is moving, extremely complex aural effects may be obtained, as the bowing point moves past the maximum of a particular mode—see Figure 7.9. In this case, good interpolation is crucial—zeroth order simple truncation of the bowing position to a nearby grid point leads to very unpleasant audible effects. See Figure 7.10. Second order (linear) interpolation vastly reduces the audibility of such motion, and fourth order interpolation is an even safer choice, though it will probably only be necessary for the benefit of the golden ear brigade, or under sublime listening conditions.

Extensions

The model of bowed-string dynamics presented above is very crude, and may be extended in many ways. Given the material in the earlier part of this chapter, a direct extension is to the case of a stiff lossy string, which may be carried out fairly easily—see Problem ?? and Programming Exercise ?. Recently, there have been investigations into more delicate features of the bow-string interaction. Among the more interesting are the modeling of torsional wave propagation in bowed strings, as well

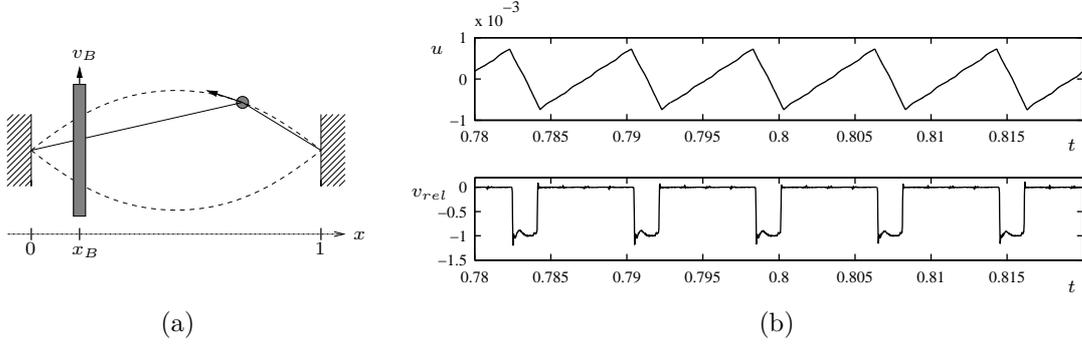


Figure 7.8: (a) Typical phenomenological description of simple bowed string motion: A point on a string, assumed bent into a triangular shape, races around an envelope of parabolic shape. (b) output waveform at the bowing point, and (c) relative velocity between string and bow, for an ideal string with a fundamental frequency of 125 Hz, under fixed end conditions, subject to bowing at constant speed $v_B = 0.2$, and with constant bow force $F_B = 50$, with the bowing point at $x = 0.25$. The bow characteristic shown in Figure 4.8(c) has been used, with $\sigma = 100$. The above outputs were produced using scheme (7.84) has been used, at a sample rate of 44 100 Hz.

as the generalization of the bow to the case of finite width [190]. Both features may be modelled using extensions of the finite difference methods presented here, but are left as an interesting exercise to the reader.

7.5 Coupling with Hammer and Mallet Models

The hammer or mallet interaction forms the basis for many percussion instruments—in the 1D case, xylophones, marimbas, and other instruments are excited in this way (not to mention, of course, the piano itself). The interaction of a simple hammer model with a mass-spring system has been presented in §4.2.3, and, indeed, most of the important issues have already been discussed. The extension to the case of an interaction with a 1D distributed system such as a string or bar is not much more difficult to deal with.

Consider, for simplicity, the case of a hammer, of mass M_H , and with stiffness defined by a one-sided nonlinearity of some type, acting on an ideal string whose dynamics are described by the 1D wave equation. Beginning from a dimensional form, defined over $\mathcal{D} = [0, L]$, and using a power law as the nonlinear characteristic (a common choice [])

$$\rho A u_{tt} = T_0 u_{xx} + \epsilon(x) f \quad f = -M_H \frac{d^2 u_H}{dt^2} = K_H ([u_H - \langle \epsilon, u \rangle_{\mathcal{D}}]^+)^{\alpha} \quad (7.91)$$

Here, ρ , A and T_0 are the mass density, cross-sectional area, and tension in the string, and the distribution $\epsilon(x)$, assumed fixed, represents the striking width of the hammer, with position u_H , mass M_H , and with stiffness parameter K_H , and nonlinear stiffness exponent α —often set in the range between 1 and 3. When spatially nondimensionalized over the interval $\mathcal{D} = [0, 1]$, the system has the form

$$u_{tt} = \gamma^2 u_{xx} + \epsilon(x) \mathcal{M} F \quad F = -\frac{d^2 u_H}{dt^2} = \omega_H^2 ([u_H - \langle \epsilon, u \rangle_{\mathcal{D}}]^+)^{\alpha} \quad (7.92)$$

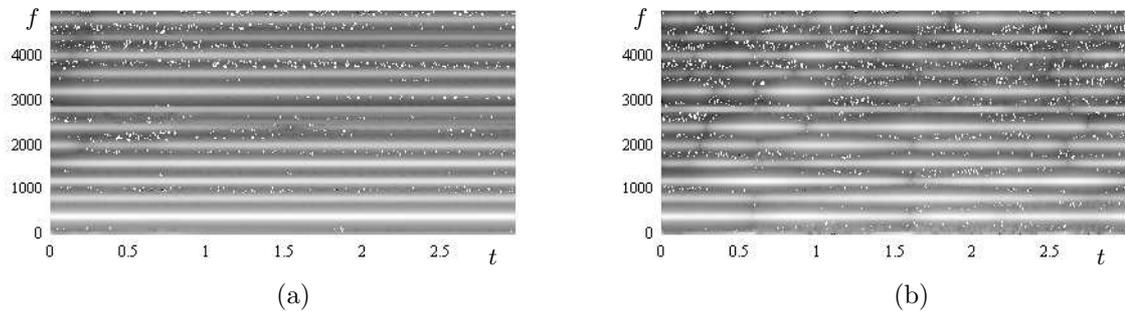


Figure 7.9: (Spectrograms of synthetic sound output for a bowed string, with fundamental frequency 400 Hz, and for a bow acting with constant $F_B = 150$ and $v_B = 0.2$, with a friction characteristic of the form shown in Figure 4.8(c), with $\sigma = 20$. Simulation performed using scheme (7.84), at a sample rate of 44.1 kHz, and readout is taken at a point 0.45 of the way along the string from the left end. (a) String bowed at a location $x = 0.25$, and (b) with a bow location varying linearly between $x = 0.25$ and $x = 0.75$, illustrating the modulation of the amplitudes of the partials in the resulting sound.

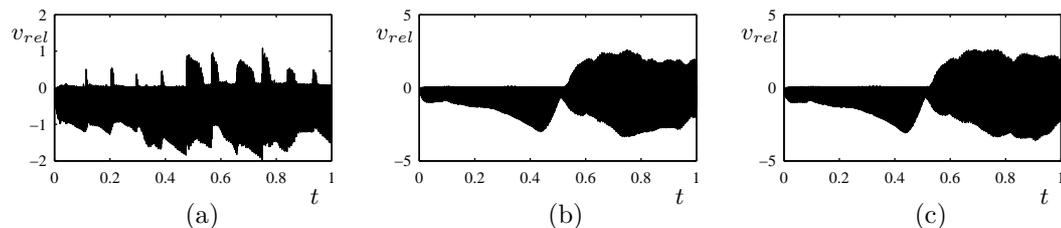


Figure 7.10: Relative velocity of the string to the bow, $v_{rel} = \delta_t u - v_B$, for a difference scheme for an ideal string with fundamental frequency 400 Hz, and with constant bow force $F_B = 150$, $v_B = 0.2$, and where the bowing point moves from $x = 0.25$ to $x = 0.45$ over a one second duration. Difference scheme (7.84) is used at sample rate 44 100 Hz, with interpolation of (a) zeroth order, (b) second order, and (c) fourth order. Notice the periodic jumps in the case shown in (a), corresponding to hard truncations of the moving bowing point to a neighboring grid point.

where the relevant parameters are now $\mathcal{M} = M_H/\rho L$, the mass ratio of the hammer to the string, and a frequency-like parameter $\omega_H = \sqrt{K_H/M_H}$.

The hammer-mallet system, of course, preserves energy:

$$\frac{d\mathfrak{H}}{dt} = 0 \quad \text{with} \quad \mathfrak{H} = \mathfrak{H}_S + \mathfrak{H}_H \quad (7.93)$$

where \mathfrak{H}_S is the energy of the string, and \mathfrak{H}_H , the energy of the hammer, is given by

$$\mathfrak{H}_H = \frac{\mathcal{M}}{2} \left(\frac{du_H}{dt} \right)^2 + \frac{\mathcal{M}\omega_H^2}{\alpha} \quad (7.94)$$

7.6 Multiple Strings

Clearly, regardless of the type of synthesis method employed, once one has implemented one string or bar model, it is straightforward to make use of several at once, to synthesize different

notes of a variety of bar or string based instruments. More interesting is the case of several strings employed in order to generate a single note—such is the case, e.g., for the piano, where normally three strings are struck at once, for each note, over most of the keyboard range. Generally, the multiple strings differ little in their material properties, though depending on the tuning strategy (in synthesis, the algorithm designer has ultimate control over tuning) slight variations in tension give rise to very desirable beating effects which go a long way towards rendering synthesis output more natural.

One may thus consider an (as yet) uncoupled system of M PDEs, of the form

$$u_{tt}^{(q)} = \left(\gamma^{(q)}\right)^2 u_{xx}^{(q)} + \dots \quad \text{for } q = 1, \dots, M \quad (7.95)$$

Here, M is the number of strings, and $u^{(q)}$ is the transverse displacement of the M th string. The ellipsis above refers to the rest of the terms in the string model, which are assumed to be of the form of (7.48) in the variable $u^{(q)}$ for each string. The term involving $\gamma^{(q)}$, however, which incorporates tensioning effects, clearly must vary from one string to the next. (One could, of course, develop a system of PDEs strings for which the other terms, not indicated above, vary from one string to the next.) Notice that in this spatially-nondimensionalized form, all the strings are of unit length.

As far as tuning goes, recall that under low stiffness and loss conditions, and under fixed boundary conditions, the pitch of the q th string will be almost exactly $\gamma^{(q)}/2$. Useful design parameters in this case are f_0 , the center pitch for the set of strings, and D , a detuning parameter, in cents. One can then set the values of $\gamma^{(q)}$ as

$$\gamma^{(q)} = 2^{1 + \frac{(2q-1-M)D}{2400(M-1)}} f_0 \quad (7.96)$$

Thus the M strings are detuned by a total of D cents from the highest pitch to the lowest.

A simple finite difference scheme follows immediately as

$$\delta_{tt} u^{(q)} = \left(\gamma^{(q)}\right)^2 \delta_{xx} u^{(q)} + \dots \quad \text{for } q = 1, \dots, M \quad (7.97)$$

A hitch arises here. The stability condition for the q th scheme will be, at least in the case of lossless non-stiff strings

$$\lambda^{(q)} = \gamma^{(q)} k / h^{(q)} \leq 1 \quad \Rightarrow \quad h^{(q)} \geq \gamma^{(q)} k \quad (7.98)$$

where here, $\lambda^{(q)}$ and $h^{(q)}$ are the Courant number and grid spacing for the q th string. If $\gamma^{(q)}$ exhibits some variation with q , then there will be distinct lower bounds on the grid spacings for the various strings. There are two ways of dealing with this: 1) take a uniform $h \geq \max_q(\gamma^{(q)} k)$ for all the strings, or 2) maintain a distinct $h^{(q)}$ for each string. For a low degree of variation in $\gamma^{(q)}$, option 1) is probably preferable, as it leads to great programming simplicity: the grid functions for all the strings will be of equal length. On the other hand, it is also clear that for under option 1), the grid spacing $h^{(q)}$ is chosen away from the Courant limit for some of the strings—thus there will be some reduction in audio bandwidth, and if this becomes noticeable, option 2) is perhaps preferable. The situation does not change much when loss and stiffness terms are included in the analysis, as there is always a lower bound on the grid spacing for any explicit scheme.

There are many ways of obtaining output. The most direct, considering that the strings in the set are all of equal perceptual importance, is to sum outputs from a given location:

$$u_{out} = \sum_{q=1}^M I(x_{out}) u^{(q)} \quad (7.99)$$

7.7 Prepared Strings

The preparation of pianos, through the insertion of objects such as erasers, washers, etc. adjacent to the strings, was popularized by Cage [46]. It serves as an excellent case study for physical modeling synthesis for several reasons. First, the objects to be modelled (namely, the strings and the preparing objects) are all of a fairly well-understood nature—indeed the whole problem (barring the soundboard, which will be discussed in §??) can be described in terms of objects which are 1D or 0D (i.e., lumped). Second, it will serve as a first example in this book of modularity in synthesis—one can begin to build more complex objects from simple ones. Finally, the motivations of Cage and others who were interested in transforming the timbres of conventional instruments are exactly in line with the aesthetic spirit of physical modeling synthesis. In this section, and that which follows, such fanciful constructions will be examined from a numerical point of view. This is the first appearance in this book of *modular* connections among physical modeling constructs, and thus the first step towards a flexible environment for the design of new virtual instruments. Some such work has been done using scattering based numerical methods (see §1.2.4 and the references therein), and in a more limited fashion, using modal methods (see §1.2.2).

As a starting point, consider again an ideal lossy string, modelled by the 1D wave equation plus a loss term, with a lumped force f acting at a given location x_P . In non-dimensional form, replacing f by $F = f/M_S$, where M_S is the total string mass, this may be written as

$$u_{tt} = \gamma^2 u_{xx} - 2\sigma_0 u_t + \delta(x_P)F \quad (7.100)$$

So far, the form of the force term is unspecified—it could result from a constraint of some kind, dependent purely on the string state itself, or it could represent a coupling to another object. Even at this stage, it is worth noting that, if the force F depends linearly on properties of the string and/or other objects, the modal frequencies of the combined system can not, in general, be related to those of the system in the absence of the force term—thus a modal approach to synthesis will require a complete recalculation of all the modal frequencies and shapes. If the connection is nonlinear, then one can expect a significant departure from the usual behaviour of the ideal string.

An energy balance for this string may be written, under conservative boundary conditions, as

$$\frac{d\mathfrak{H}_S}{dt} = -2\sigma_0 \|u_t\|_{\mathbb{U}}^2 + F u_t(x_P) \quad (7.101)$$

where \mathfrak{H}_S is the usual Hamiltonian for the wave equation, as per (6.20). It is not difficult to add additional terms to the above model representing stiffness and frequency-dependent damping, but the above model suffices for a simple discussion of the main numerical issues.

7.7.1 Springs and Dampers

Perhaps the simplest type of preparation involves massless springs, and damping elements—see Figure ???. A simple combined model of such a model is

$$F = -\omega_0^2 u(x_P) - \omega_1^4 (u(x_P))^3 - 2\sigma_P u_t(x_P) \quad (7.102)$$

Here, there are two terms representing a linear spring, and a cubic nonlinear spring, as well as a damping term. Using the usual techniques, the energy balance becomes

$$\frac{d\mathfrak{H}}{dt} = -2\sigma_0 \|u_t\|_{\mathbb{U}}^2 - 2\sigma_P (u_t(x_P))^2 \quad \text{with} \quad \mathfrak{H} = \mathfrak{H}_S + \mathfrak{H}_P \quad \text{and} \quad \mathfrak{H}_P = \frac{\omega_0^2}{2} (u(x_P))^2 + \frac{\omega_1^4}{4} (u(x_P))^4 \quad (7.103)$$

Here, there is now an additional component to the stored energy, \mathfrak{H}_P , due to the string. Notice that the energy for the combined system is non-negative, and, as before, monotonically decreasing.

There are, of course, many different choices now available for discretization—not all have the same properties! Suppose first that the wave equation (7.100) is discretized using the basic difference scheme,

$$\delta_{tt}u = \gamma^2 \delta_{xx}u - 2\sigma_0 \delta_t u + J_p(x_P)F \quad (7.104)$$

where here, $J_p(x_P)$ is a p th order spreading function (see §5.2.4). One way of discretizing the force equation is as

$$F = -\omega_0^2 \mu_t \eta - \omega_1^4 \eta^2 \mu_t \eta - 2\sigma_P \delta_t \eta \quad \text{where} \quad \eta = I_p(x_P)u \quad (7.105)$$

and $I_p(x_P)$ is a p th order interpolation operator (see §5.2.4). The system, though not linear, has been discretized in such a way as to indicate that the result will be strictly dissipative. Examining each of the terms in F separately: the linear spring term has been treated exactly as in the semi-implicit scheme for the oscillator, as discussed in §3.3.1, the cubic nonlinear oscillator term as per the method given in (4.9b), and the extra damping term by a simple centered difference, as per the lossy SHO—see §3.5.2. As one might expect, the system as a whole possesses a discrete energy balance of the form

$$\delta_{t+} \mathfrak{h} = -2\sigma_0 \|\delta_t u\|_{\mathbb{U}_N}^2 - 2\sigma_P (\delta_t \eta)^2 \quad \text{with} \quad \mathfrak{h} = \mathfrak{h}_S + \mathfrak{h}_P \quad \text{and} \quad \mathfrak{h}_P = \frac{\omega_0^2}{2} \mu_{t-} (\eta^2) + \frac{\omega_1^4}{4} \eta^2 e_{t-} \eta^2 \quad (7.106)$$

where \mathfrak{h}_S is the stored energy for scheme (??) for the wave equation, s given in (6.79). Notice that the expression for the energy of the prepared string is non-negative—thus the total energy \mathfrak{h} must also remain non-negative under the condition $\lambda \leq 1$, and, thus, stable. Under this particular discretization, the addition of a lumped element has no effect on stability of the combined system, which is a very useful feature, from the point of view of modular physical modeling. It is not true of all possible discretizations, however, as will be noted shortly.

Besides stability, the other crucial feature of a numerical method simulating an arbitrary connection among objects is that of computability—note that in the coupled expressions (7.104) and (7.105), there is a simultaneous dependence on u^{n+1} , the unknowns. Complicating matters is the use of interpolation and spreading operators, distributing the coupling over various nearby grid points. This may be resolved in the following way. Taking an inner product of (7.104) with $J_p(x_P)$ (or, equivalently, interpolating the equation with $I_p(x_P)$) gives

$$\delta_{tt} \eta = \gamma^2 \zeta - 2\sigma_0 \delta_t \eta + \|J_p(x_P)\|_{\mathbb{U}_N}^2 F \quad \text{with} \quad \eta = I_p(x_P)u \quad \zeta = I_p(x_P) \delta_{xx} u \quad (7.107)$$

Given that, at update step $n+1$, ζ is known, then one may solve directly for η^{n+1} in terms of known values using (7.105):

$$a = 1 + \sigma_0 k + h k^2 \|J_p(x_P)\|_{\mathbb{U}_N}^2 (\omega_0^2/2 + \omega_1^4 (\eta^n)^2/2 + \sigma_P/k) \quad (7.108)$$

$$b = -1 + \sigma_0 k - h k^2 \|J_p(x_P)\|_{\mathbb{U}_N}^2 (\omega_0^2/2 + \omega_1^4 (\eta^n)^2/2 - \sigma_P/k) \quad (7.109)$$

$$\eta^{n+1} = \frac{1}{a} (\gamma^2 k^2 \zeta + 2\eta^n) + \frac{b}{a} \eta^{n-1} \quad (7.110)$$

Once η^{n+1} is determined, then F^n is known, from (7.105), and then, finally, the scheme (7.104) may be updated explicitly.

The variety of timbres which can be generated using such a simple connection is surprisingly large. Beginning first with the case of a lossless linear spring connection (i.e., $\omega_1 = \sigma_P = 0$), the main effect will be, in the space time domain, to introduce reflection of traveling waves—see Figure 7.11(a); the amount of reflection and transmission depends on the choice of the stiffness parameter ω_0 .

In the frequency domain, the result is a shifting of modal frequencies of the combined spring/string system—for the present case of the wave equation, this means that the resulting spectrum of sound output will be inharmonic, and increasingly so as the lumped stiffness increases in strength—see Figure 7.11(b). (Note that there will be an additional dependence on the positioning of the spring (i.e., on the choice of x_P .) As such, the resulting sounds will be far more like those of percussion instruments than typical stringed instruments.

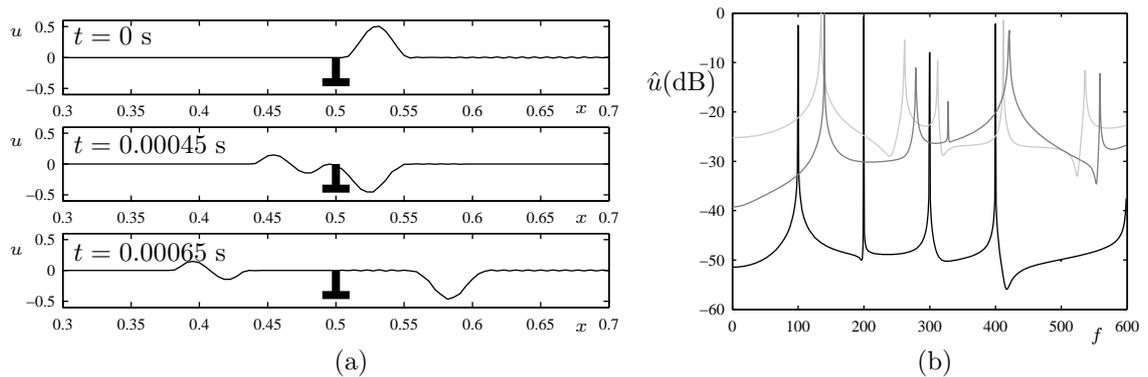


Figure 7.11: *Ideal lossless string, with $\gamma = 200$, with a connected linear spring at $x_P = 0.5$. (a) Reflection and transmission of an incoming pulse from the spring, with stiffness parameter $\omega_0 = 4000$, at times as indicated. (b) Output spectrum for sound created using a plucked raised cosine initial condition of amplitude 1, with output read at $x = 0.8$, under different choices of the spring stiffness: $\omega_0 = 1000$ (in dark grey), $\omega_0 = 2000$ (in light grey), and without a spring connection (in black). Scheme (7.104) coupled to (7.105) is used, and the sample rate is 44100 Hz.*

Considering the case of a pure linear damper (i.e., $\omega_0 = \omega_1 = 0$), the most obvious effect will be that of increased damping, beyond that inherent in the string model itself. What is more interesting, however, is the frequency dependence of the damping introduced, which, again, will depend on the damper position, as well as the loss coefficient σ_P . If σ_P is large, the string is effectively divided into two, yielding completely different pitches from the unprepared string. See Figure 7.12(b), which shows a spectrogram of a typical output for such a point-damped string.

The most interesting case is that of the nonlinear spring (i.e., $\sigma_P = \omega_0 = 0$), especially in the case for which there is some damping in the string model itself. Initially, there will be a rather lively variation in the frequency components, as illustrated in Figure 7.12(a), but as damping intervenes, the behaviour settles down to that of the unprepared string. Note in particular that in the limiting case of low amplitude vibration, the behaviour of the combined system approaches that of the unprepared system, which is distinct from that of the linear spring connection, which always exhibits inharmonicity, regardless of the amplitude of excitation.

Numerical Instability of Connections

The scheme (7.105) which discretizes the connection with the spring/damper system has been specially chosen so as to be numerically stable always—the energy function for the combined string/spring/damper system is positive definite, and monotonically decreasing. But it is clear that due to the implicit nature of the difference scheme, this is not the simplest scheme available.

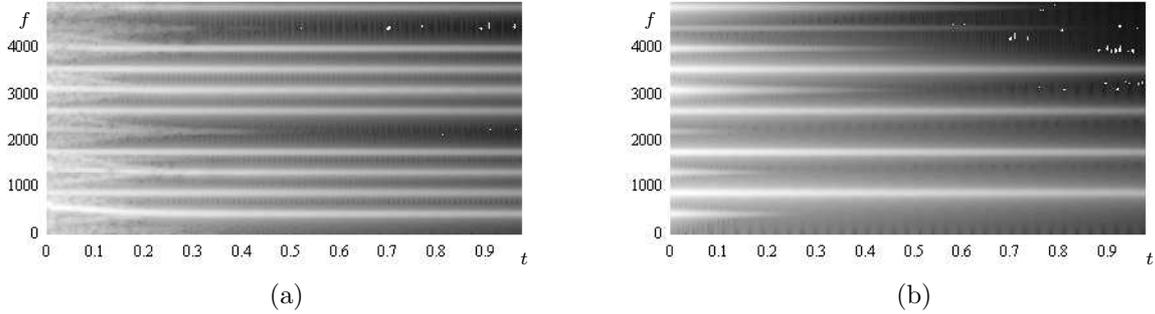


Figure 7.12: Spectrograms of output for a model of an ideal lossless string with loss, with $\gamma = 880$, and $T_{60} = 2$, connected to: (a) a cubic nonlinear spring at $x_P = 0.5$, with $\omega_1 = 10$, and (b) a linear damper, with $T_{60} = 1$. In this second case, the damper partially divides the string into two segments, yielding an effective pitch of twice the fundamental of the string after several hundred milliseconds. Scheme (7.104) coupled to (7.105) is used in both cases, and the sample rate is 44100 Hz, with output taken at $x = 0.8$, and where the initial condition is of plucked type, using a raised cosine distribution centered at $x = 0.65$, and of amplitude 1.

Indeed, considering the special case of the linear spring alone (i.e., with $\omega_1 = \sigma_P = 0$), one could use, instead of (7.105),

$$F = -\omega_0^2 I_p(x_P)u \quad (7.111)$$

which leads to a completely update of (7.104). Indeed, such a direct discretization of a stiffness term is directly in line with the stable discretization of the stiffness term in the SHO, as per (3.12), and one might expect that the combined string/spring system should remain stable as well. In fact, this is not necessarily the case—as shown in Figure 7.13, the solution can indeed become unstable, with the instability manifesting itself, as expected, by high-frequency oscillations emanating from the connection point which grow and eventually obliterate the solution.

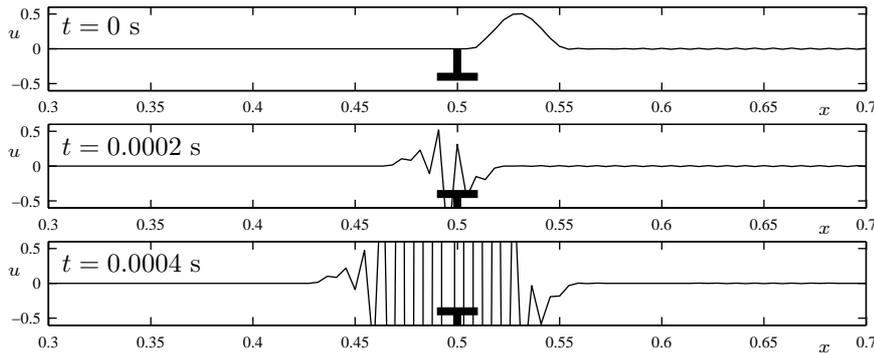


Figure 7.13: Development of instability in a connection between an ideal lossless string, with $\gamma = 200$, and a linear spring at $x_P = 0.5$, of stiffness $\omega_0 = 4000$, using finite difference scheme (7.104) along with (7.111). The sample rate is 44100 Hz.

In order to examine this, suppose, for the sake of simplicity, that the interpolation function is of zeroth order, so that the spring acts directly on the value of the grid function u_l at the grid point located at $l = P$. Under lossless conditions, and for conservative boundary conditions on the string, the total conserved energy for this combined system will be

$$\mathfrak{h} = \underbrace{\frac{1}{2} \|\delta_{t-} u\|_{\underline{U}_N}^2}_{\mathfrak{h}_S} + \underbrace{\frac{\gamma^2}{2} \langle \delta_{x+} u, e_{t-} \delta_{x+} u \rangle_{\underline{U}_N}}_{\mathfrak{h}_P} + \underbrace{\frac{\omega_0^2}{2} u_P e_{t-} u_P}_{\mathfrak{h}_P} \quad (7.112)$$

where \mathfrak{h}_S indicates the energy of the string, and \mathfrak{h}_P the added potential energy of the spring connection. Clearly, neither of the contributions is necessarily non-negative. One may bound them as follows:

$$\mathfrak{h}_S \geq \frac{1}{2} (1 - \lambda^2) \|\delta_{t-} u\|_{\underline{U}_N}^2 = \quad \mathfrak{h}_P \geq \frac{k^2 \omega_0^2}{8} (\delta_{t-} u_P)^2 \quad (7.113)$$

which implies, for the sum, that

$$\mathfrak{h} \geq \sum_{l \neq P} \frac{h}{2} (1 - \lambda^2) (\delta_{t-} u)^2 + \left(\frac{h}{2} (1 - \lambda^2) - \frac{k^2 \omega_0^2}{8} \right) (\delta_{t-} u_P)^2 \quad (7.114)$$

At grid points other than $l = P$, the non-negativity condition is, as before, the CFL condition $\lambda < 1$. But at $l = P$, there is another, stronger condition that must be enforced:

$$\frac{h}{2} (1 - \lambda^2) - \frac{k^2 \omega_0^2}{8} > 0 \quad (7.115)$$

In this case, the stability condition of the combination *interferes* with that of the systems in isolation, which is very different from the case of the discretization from (7.105). The lesson here is that the semi-implicit character of a scheme such as (7.105) allows for a modular connection without the additional worry of inducing instability—at the expense, of course, of a slightly more involved update. This is typical of many numerical methods for combined systems, and, in fact, is exactly the principle behind scattering based numerical approaches—see §1.2.4, and the recent publication by Rabenstein et al. [196]. Indeed, in the present case of a string/spring connection, if one were to use a digital waveguide for the string (corresponding to exactly scheme (7.104)), and a wave digital filter for the spring (which is also semi-implicit, like (7.105)), modular stability is also obtained. But, as illustrated here, it is not necessary to make use of scattering principles in order to arrive at such a result. When connecting a lumped object to a distributed object, a good rule of thumb is the following:

Rule of Thumb # 3

Efficient modular numerical behaviour is obtained when a semi-implicit method is used for lumped objects, and when an explicit method is used for distributed objects.

7.7.2 Masses

The extension of preparation of strings to elements with mass is straightforward, but leads to a slightly more involved formulation—one now must keep track of a separate position for the connected object. In other words, more state is required to represent the combined system. Supposing the position of the mass is $u_P(t)$, then, when connected through a linear spring of stiffness parameter

ω_0 to the ideal string, the force acting on the string is given by

$$F = -\mathcal{M} \frac{d^2 u_P}{dt^2} = \mathcal{M} \omega_0^2 (u_P - I(x_P)u) \quad (7.116)$$

where \mathcal{M} is the mass ratio of the lumped mass to the string. Now, the combined energy balance is

$$\frac{d\mathfrak{H}}{dt} = -2\sigma_0 \|u_t\|_{\mathbb{U}}^2 \quad \text{with} \quad \mathfrak{H} = \mathfrak{H}_S + \mathfrak{H}_P \quad \text{and} \quad \mathfrak{H}_P = \frac{\mathcal{M} \omega_0^2}{2} (u(x_P) - u_P)^2 + \frac{\mathcal{M}}{2} \left(\frac{du_P}{dt} \right)^2 \quad (7.117)$$

7.7.3 Rattling Elements

7.8 Coupled Strings and Bars

Part of the fun of physical modeling (and indeed, maybe the very point of it all) is to go beyond what can be done with an acoustic instrument. There are eminently physical ways of modifying a synthesis routine which would be difficult, if not impossible to carry out in the real world—witness the caution with which an orchestra director will allow a Steinway grand to be prepared. Coupling of string and bar models is one such example, and in the virtual world, no blowtorch is necessary.

Coupling between distributed systems and lumped objects has been discussed extensively in the last few sections. The same principles of analysis may be applied when distributed objects are coupled. For the sake of variety, consider the case of two ideal bars, coupled through some as yet unspecified mechanism, as illustrated in Figure 7.14.



Figure 7.14: (a) A pair of bars, pointwise coupled by an arbitrary mechanism relating the resulting forces. (b) A rigid connection, (c) a damped connection, and (d) a spring-like connection.

Supposing that the bars are of differing material properties, the system may be written, in dimensional form, as

$$\rho^{(1)} A^{(1)} u_{tt}^{(1)} = -E^{(1)} A^{(1)} u_{x^{(1)} x^{(1)} x^{(1)} x^{(1)}} + \delta(x_i^{(1)}) f^{(1)} \quad (7.118)$$

$$\rho^{(2)} A^{(2)} u_{tt}^{(2)} = -E^{(2)} A^{(2)} u_{x^{(2)} x^{(2)} x^{(2)} x^{(2)}} + \delta(x_i^{(2)}) f^{(2)} \quad (7.119)$$

Here, the superscripts (1) and (2) refer to the first and second bar, respectively—notice that different spatial coordinates $x^{(1)}$ and $x^{(2)}$ are used for each. The bars are assumed to lie over the spatial intervals $x^{(1)} \in [0, L^{(1)}]$ and $x^{(2)} \in [0, L^{(2)}]$, respectively, and the connection points are $x_i^{(1)}$ and $x_i^{(2)}$, respectively. Using the techniques discussed earlier in this chapter, one could easily extend this model to the case of a stiff string, with frequency-dependent loss, or to the case of a distributed connection, or using methods to be introduced in the next section, to the case of bars of variable cross-sectional area or density. This model behaves somewhat like a tuning fork, depending on the type of connection (i.e., on the definition of the mechanism giving rise to the forces $f^{(1)}$ and $f^{(2)}$), and could serve as a string point for physical models of certain electromechanical instruments such as the Fender-Rhodes electric piano. See Figure 7.15 for a picture of typical behaviour for a simple

coupled bar system.

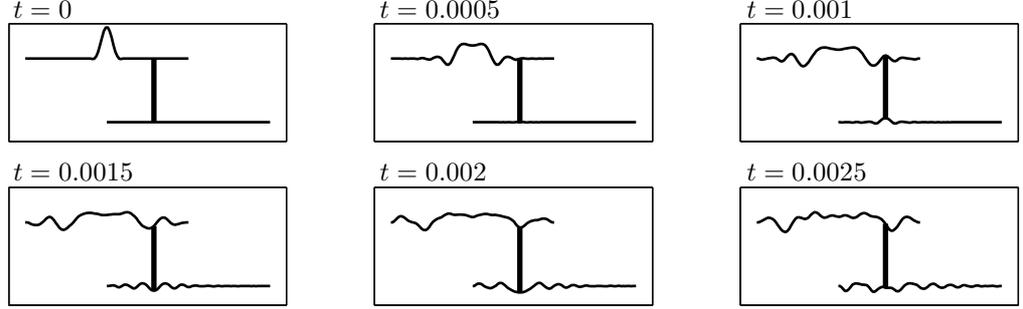


Figure 7.15: Time evolution of displacement profiles for a coupled bar system, at times as indicated. Here, the bars are of stiffnesses $\kappa^{(1)} = 2$ (bottom bar) and $\kappa^{(2)} = 3$ (top bar), and are coupled via a rigid connection occurring at $x_i^{(1)} = 0.3$, and $x_i^{(2)} = 0.8$. Boundary conditions are of clamped type, and the second bar is initialized with a plucked raised cosine distribution at the bar center.

Using the first bar as a reference, non-dimensionalization may be carried out, using coordinates $x^{(1)'} = x^{(1)}/L^{(1)}$ and $x^{(2)'} = x^{(2)}/L^{(2)}$, and subsequently removing primes, as

$$u_{tt}^{(1)} = -\left(\kappa^{(1)}\right)^2 u_{x^{(1)}x^{(1)}x^{(1)}x^{(1)}}^{(1)} + \delta(x_i^{(1)})F^{(1)} \quad (7.120)$$

$$\mathcal{M}u_{tt}^{(2)} = -\mathcal{M}\left(\kappa^{(2)}\right)^2 u_{x^{(2)}x^{(2)}x^{(2)}x^{(2)}}^{(2)} + \delta(x_i^{(2)})F^{(2)} \quad (7.121)$$

where

$$\kappa^{(1)} = \sqrt{\frac{E^{(1)}}{\rho^{(1)}(L^{(1)})^4}} \quad \kappa^{(2)} = \sqrt{\frac{E^{(2)}}{\rho^{(2)}(L^{(2)})^4}} \quad \mathcal{M} = \frac{\rho^{(2)}A^{(2)}}{\rho^{(1)}A^{(1)}} \quad F^{(1)} = \frac{f^{(1)}}{\rho^{(1)}A^{(1)}L^{(1)}} \quad F^{(2)} = \frac{f^{(2)}}{\rho^{(2)}A^{(2)}L^{(2)}} \quad (7.122)$$

Now, both bars are defined over the unit interval $\mathcal{D} = \mathbb{U}$.

The energy balance for this system, assuming conservative boundary termination (see §7.1.2 for the case of the ideal bar) is

$$\frac{d\mathfrak{H}}{dt} = F^{(1)}u_t^{(1)}(x_i^{(1)}) + F^{(2)}u_t^{(2)}(x_i^{(2)}) \quad \text{with} \quad \mathfrak{H} = \mathfrak{H}^{(1)} + \mathfrak{H}^{(2)} \quad (7.123)$$

and

$$\mathfrak{H}^{(1)} = \frac{1}{2}\|u_t^{(1)}\|_{\mathbb{U}}^2 + \frac{(\kappa^{(1)})^2}{2}\|u_{x^{(1)}x^{(1)}}\|_{\mathbb{U}}^2 \quad \mathfrak{H}^{(2)} = \frac{1}{2}\|u_t^{(2)}\|_{\mathbb{U}}^2 + \frac{(\kappa^{(2)})^2}{2}\|u_{x^{(2)}x^{(2)}}\|_{\mathbb{U}}^2 \quad (7.124)$$

Clearly, the system as a whole will be dissipative when

$$F^{(1)}u_t^{(1)}(x_i^{(1)}) + F^{(2)}u_t^{(2)}(x_i^{(2)}) \leq 0 \quad (7.125)$$

or lossless when the above inequality is satisfied with equality. If the connection itself can store energy (i.e., if there are masses or springs involved), then dissipative behaviour will occur if

$$F^{(1)}u_t^{(1)}(x_i^{(1)}) + F^{(2)}u_t^{(2)}(x_i^{(2)}) = -\frac{d\mathfrak{H}_C}{dt} - Q \quad (7.126)$$

where $Q \geq 0$, because in this case, one will then have

$$\frac{d}{dt}(\mathfrak{H} + \mathfrak{H}_C) = -Q \quad (7.127)$$

7.8.1 Connection Types

The simplest type of connection is that of rigid type:

$$F^{(1)} = -F^{(2)} \quad \text{and} \quad u_t^{(1)}(x_i^{(2)}) = u_t^{(2)}(x_i^{(2)}) \quad (7.128)$$

satisfying (7.125) with equality. Forces are equal and opposite, and the bars are constrained to move at the same velocity at the connection point. The resulting behaviour of such a system is, from a spectral point of view, enormously complex—in general, the modal frequencies will depend on the relative stiffnesses of the bars, the mass ratio, as well as the connection points. See Figure 7.16. Even for such a simple connection, there is a huge variety of timbres available to the instrument designer.

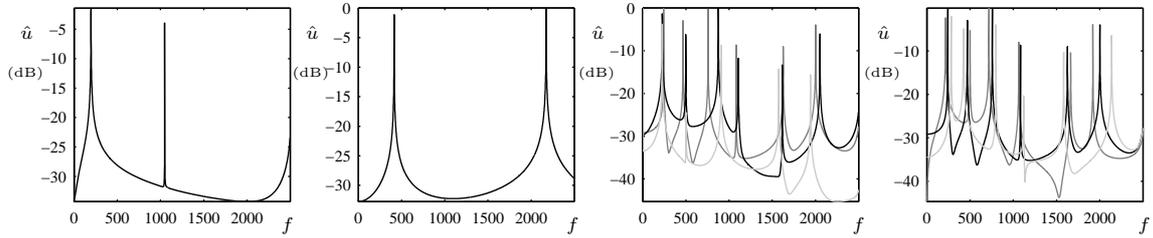


Figure 7.16: *Output spectra for a coupled bar system. Typical output spectra for bars in isolation, under clamped conditions, for stiffness parameters (a) $\kappa^{(1)} = 50$, and (b) $\kappa^{(2)} = 100$. In (c) is shown output spectra under variations in the connection point, for two bars of mass ratio $M = 1$. The connection point on the first bar is at $x_i^{(1)} = 0.3$, and on the second bar at $x_i^{(2)} = 0.8$ (in dark grey), 0.7 (in black) and 0.6 (in light grey). In (d), a similar comparison of spectra is shown, under variations in the mass ratio, for fixed connection points $x_i^{(1)} = 0.3$ and $x_i^{(2)} = 0.8$, where $M = 0.3$ (dark grey), 1 (black) and 10 (light grey).*

A device involving both damping and a mixed linear-cubic spring is described by

$$F^{(1)} = -F^{(2)} = -\omega_0^2 \eta - \omega_1^4 \eta^3 - 2\sigma_C \frac{d}{dt} \eta \quad \text{with} \quad \eta = u^{(1)}(x_i^{(1)}) - u^{(2)}(x_i^{(2)}) \quad (7.129)$$

Such a spring obviously possesses a potential energy \mathfrak{H}_C , given by

$$\mathfrak{H}_C = \frac{\omega_0^2}{2} \eta^2 + \frac{\omega_1^4}{4} \eta^4 \quad (7.130)$$

which renders the system dissipative as a whole, with an energy balance of the form (7.127), with $Q = -2\sigma_C (d\eta/dt)^2$. Connections with mass are also a possibility—see Problem ???. Even more variability in timbre is now possible—see Figure 7.17. The linear spring action serves to further alter modal frequencies, as does the damping term, which also generally introduces frequency dependent decay. Interestingly, depending on the connection points, and the strength of the damper, it can allow for energy to be traded back and forth between the bars, at potentially a very slow rate, leading to interesting sub-audio rate variations in amplitude, as is visible in Figure 7.17(a) and (b). Finally, the nonlinear spring term in conjunction with damping yields pitch changes and wide-band sound output, dependent on the strength of the excitation.

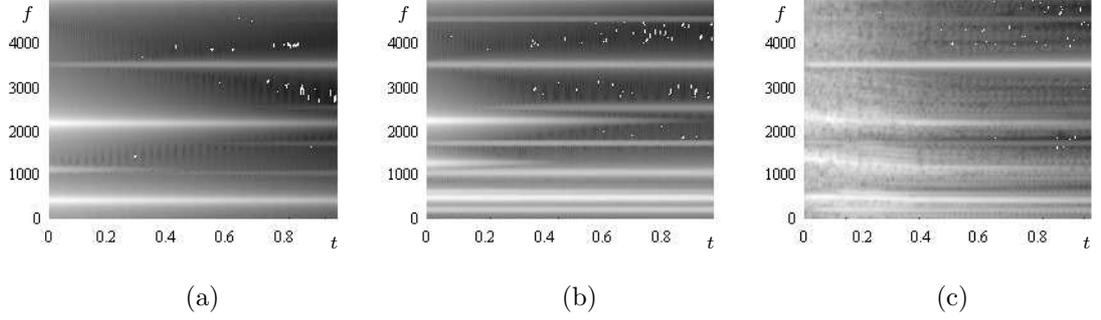


Figure 7.17: Spectrograms of audio output for two bars under a damped spring connection. The two bars are of stiffness coefficients $\kappa^{(1)} = 50$ and $\kappa^{(2)} = 100$, and of mass ratio $\mathcal{M} = 0.2$, and the connection points are $x_i^{(1)} = 0.3$ and $x_i^{(2)} = 0.8$. In (a), a pure damper, of damping coefficient $\sigma_P = 1.4$ is employed, in (b), the same damper with a linear spring with $\omega_0 = 1000$, and in (c), the same damper again with a nonlinear spring with $\omega_1 = 70$. Output is taken from the second bar, at location $x^{(2)} = 0.5$.

7.8.2 Finite Difference Schemes

Finite difference schemes may be developed in the usual way for the coupled bar system—the one new feature here is that, in general, because the bars may be of differing stiffnesses (i.e., $\kappa^{(1)} \neq \kappa^{(2)}$), it is not necessarily a good idea to use the same grid spacing for each. The reason for this is simple—if a global sample rate is chosen, then a distinct stability condition will arise for each bar in isolation. If the spacing is chosen the same for both bars, one will necessarily induce heavy numerical dispersion. Recall, from §??, that if one wants to use a simple explicit scheme, numerical dispersion effects are very strong, especially in the case of the bar. Thus, choose grid spacings $h^{(1)}$ and $h^{(2)}$ for the first and second bars, respectively.

The simplest scheme is of the following form:

$$\delta_{tt}u^{(1)} = -\left(\kappa^{(1)}\right)^2 \delta_{x^{(1)}x^{(1)}x^{(1)}x^{(1)}}u^{(1)} + J_p(x_i^{(1)})F^{(1)} \quad (7.131a)$$

$$\mathcal{M}\delta_{tt}u^{(2)} = -\mathcal{M}\left(\kappa^{(2)}\right)^2 \delta_{x^{(2)}x^{(2)}x^{(2)}x^{(2)}}u^{(2)} + J_p(x_i^{(2)})F^{(2)} \quad (7.131b)$$

where as before, J_p represents a spreading operator of order p . Notice that the two instances of J_p will be acting over distinct grids. The discrete energy balance, under numerical conservative boundary conditions will be

$$\delta_{t+}\mathfrak{h} = \delta_t.(I_p(x_i^{(1)}))F^{(1)} + \delta_t.(I_p(x_i^{(2)}))F^{(2)} \quad \text{with} \quad \mathfrak{h} = \mathfrak{h}^{(1)} + \mathfrak{h}^{(2)} \quad (7.132)$$

and where the discrete energies for the two bars are given for bars (1) and (2) by

$$\mathfrak{h}^{(1),(2)} = \frac{1}{2}\|\delta_t u\|_{\underline{U}_N}^2 + \frac{(\kappa^{(1),(2)})^2}{2}\langle \delta_{x^{(1),(2)}x^{(1),(2)}}u^{(1),(2)}, e_{t-}\delta_{x^{(1),(2)}x^{(1),(2)}}u^{(1),(2)} \rangle_{\underline{U}_N} \quad (7.133)$$

These are the same expressions as in the case of uncoupled bars (see §??), and are non-negative under the conditions

$$\mu^{(1)} = \frac{k\kappa^{(1)}}{h^{(1)}} \leq \frac{1}{2} \quad \mu^{(2)} = \frac{k\kappa^{(2)}}{h^{(2)}} \leq \frac{1}{2} \quad (7.134)$$

As mentioned above, the grid spacings $h^{(1)}$ and $h^{(2)}$ should be chosen to satisfy the above bounds as close to equality as possible.

The rigid connections is the simplest to deal with. One may use

$$F^{(1)} = -F^{(2)} \quad I_p(x_i^{(1)})u^{(1)} = I_p(x_i^{(2)})u^{(2)} \quad (7.135)$$

Here, the positions of the bars at the connection point are set equal—this is a particularly easy way to enforce a condition of equal velocity (instead of adding an extra parameter representing the distance between the bars, which is of no consequence). Under these conditions, there is exact numerical energy conservation from (7.132), and thus numerical stability follows if conditions (7.134) are respected.

The damper and spring connection is more complex, but may be dealt with in a manner similar to the lumped spring connection discussed in §7.7.1. A semi-implicit discretization leads to

$$F^{(1)} = -F^{(2)} = -2\sigma_C\delta_t \left(I_p(x_i^{(1)})u^{(1)} - I_p(x_i^{(2)})u^{(2)} \right) - \omega_0^2\mu_t\eta - \omega_1^4\eta^2\mu_t\eta \quad (7.136)$$

with

$$\eta = I(x_i^{(1)})u^{(1)} - I(x_i^{(2)})u^{(2)} \quad (7.137)$$

Explicit Updating

Due to the implicit nature of the approximation to $F^{(1)}$ and $F^{(2)}$, it is not immediately clear how to go about performing an update. The forces depend on unknown values of the grid functions, and conversely—a complicating factor is that due to the spreading functions, there is in fact a coupled region of interdependence of forces and displacements. An explicit update is indeed possible—as a first step, apply the interpolation operators $I_p(x_i^{(1)})$ and $I_p(x_i^{(2)})$ to Eqs. (7.131a) and (7.131b), respectively, to get:

$$\delta_{tt}I_p(x_i^{(1)})u^{(1)} = -\zeta^{(1)} + h^{(1)}\|J_p(x_i^{(1)})\|_{\mathbb{U}_{N_1}}^2 F^{(1)} \quad (7.138)$$

$$\mathcal{M}\delta_{tt}I_p(x_i^{(2)})u^{(2)} = -\mathcal{M}\zeta^{(2)} + h^{(2)}\|J_p(x_i^{(2)})\|_{\mathbb{U}_{N_2}}^2 F^{(2)} \quad (7.139)$$

where $\zeta^{(1)}$ and $\zeta^{(2)}$ are known, and given by

$$\zeta^{(1)} = \left(\kappa^{(1)}\right)^2 \delta_{x^{(1)}x^{(1)}x^{(1)}x^{(1)}} u^{(1)} \quad \zeta^{(2)} = \left(\kappa^{(2)}\right)^2 \delta_{x^{(2)}x^{(2)}x^{(2)}x^{(2)}} u^{(2)} \quad (7.140)$$

Given $F^{(1)} = -F^{(2)}$, one may solve for the unknown forces (say, $F^{(1)}$), as

$$F^{(1)} = \frac{\zeta^{(1)} - \zeta^{(2)} + \delta_{tt}\eta}{h^{(1)}\|J_p(x_i^{(1)})\|_{\mathbb{U}_{N_1}}^2 + h^{(2)}\|J_p(x_i^{(2)})\|_{\mathbb{U}_{N_2}}^2 / \mathcal{M}} \quad (7.141)$$

In the case of a rigid connection (i.e., when $F^{(1)} = -F^{(2)}$ and $I_p(x_i^{(1)}) = I_p(x_i^{(2)})$ implying $\eta = 0$, as per (7.135)), $F^{(1)}$ may be calculated immediately from known values of the grid functions. Once the forces are known, they may be inserted directly into the updates (7.131a) and (7.131b), which are fully explicit.

For spring/damper connections, the situation is slightly more complicated: η is as yet unknown in the expression for the force above. Note, however, that using (7.136), one may write

$$\frac{\zeta^{(1)} - \zeta^{(2)} + \delta_{tt}\eta}{h^{(1)}\|J_p(x_i^{(1)})\|_{\mathbb{U}_{N_1}}^2 + h^{(2)}\|J_p(x_i^{(2)})\|_{\mathbb{U}_{N_2}}^2 / \mathcal{M}} = -\omega_0^2\mu_t\eta - \omega_1^4\eta^2\mu_t\eta - 2\sigma_P\delta_t\eta \quad (7.142)$$

which is a linear equation in the unknown value of η at time step $n + 1$, and which may be solved directly. Once η is determined, the forces may be determined, and, as before, inserted into the updates (7.131a) and (7.131b).

7.9 Spatial Variation and Stretched Coordinates

All the systems discussed up to this point have been linear and shift invariant, or LSI (see §5.1.1). Some objects which occur in musical acoustics exhibit variations in material properties with position—in 1D, the most important example is certainly the case of the acoustic tube of variable cross-section, which features prominently in wind instruments and the voice. Simulation for the acoustic tube will be discussed in Chapter 9. In the arena of string and bar vibration, the main example is the bar of varying cross-section, which appears in percussion instruments such as xylophones [116], and the string of variable density is a useful preliminary test problem.

In such cases, some of the analysis tools used up to this point lose their utility—there is no longer a well-defined notion of a phase or group velocity, and von Neumann analysis becomes unwieldy², though the notions of a modal expansion and characteristic frequencies persist. A good treatment of energy analysis of finite difference schemes for systems similar to those which appear in this section is that of Cohen and Joly [58]

7.9.1 Strings of Varying Density

As a first example, consider the equation of motion of a string, length L , under tension T_0 , with a linear mass density $\rho_l = \rho_0 \epsilon^2(x)$, in dimensional form:

$$\rho_0 \epsilon^2(x) u_{tt} = T_0 u_{xx} \quad \text{over} \quad x \in [0, L] \quad (7.143)$$

Here, ρ_0 is the average density over the length of the string, and $\epsilon(x)$ represents the variations. This can be non-dimensionalized as

$$\epsilon^2(x) u_{tt} = \gamma_0^2 u_{xx} \quad \text{over} \quad x \in \mathbb{U} \quad (7.144)$$

where $\gamma_0 = \sqrt{T_0/\rho_0 L^2}$. One can think of this equation, in a very rough sense, as a wave equation with a wave speed $\gamma = \gamma_0/\epsilon(x)$ which varies with position. It should be clear that now, the modal shapes will depend wholly on the density profile, and, in general, can only be computed numerically.

Energy analysis may be applied just as in previous examples. One arrives at

$$\frac{d\mathfrak{H}}{dt} = \mathfrak{B} \quad \text{with} \quad \mathfrak{H} = \mathfrak{T} + \mathfrak{V} \quad (7.145)$$

and where now, one has

$$\mathfrak{T} = \frac{1}{2} \|\epsilon u_t\|_{\mathbb{U}}^2 \quad \mathfrak{V} = \frac{\gamma_0^2}{2} \|u_x\|_{\mathbb{U}}^2 \quad \mathfrak{B} = \gamma_0^2 (u_t(1, t) u_x(1, t) - u_t(0, t) u_x(0, t)) \quad (7.146)$$

Thus one may come to the same conclusions regarding conservative boundary conditions as in the case of the 1D wave equation. In particular, the Dirichlet and Neumann conditions discussed in §6.1.9 continue to lead to energy conservation.

Consider now the obvious choice for a finite difference scheme for (7.144):

$$\epsilon^2 \delta_{tt} u = \gamma_0^2 \delta_{xx} u \quad (7.147)$$

Here, $\epsilon = \epsilon_l$ now represents the grid function obtained through direct sampling of the given function $\epsilon(x)$ at the grid locations $x = lh$. Energy analysis, i.e., through an inner product with $\delta_t u$, leads to the energy balance

$$\delta_{t+} \mathfrak{h} = \mathfrak{b} \quad \text{with} \quad \mathfrak{h} = \mathfrak{t} + \mathfrak{v} \quad (7.148)$$

²von Neumann stability analysis may be carried out using so-called frozen-coefficient analysis [244], but energy analysis is an elegant and informative alternative.

and

$$\mathbf{t} = \frac{1}{2} \|\epsilon \delta_{t-} u\|_{\mathbb{U}_N}^2 \quad \mathbf{v} = \frac{\gamma_0^2}{2} \langle \delta_{x-} u, e_{t-} \delta_{x-} u \rangle_{\mathbb{U}_N} \quad \mathbf{b} = \gamma_0^2 (\delta_t \cdot u_N \delta_{x+} u_N - \delta_t \cdot u_0 \delta_{x-} u_0) \quad (7.149)$$

Under what conditions is this quantity non-negative? Assuming conservative boundary conditions, so that $\mathbf{b} = 0$, one has, using the same techniques as in the case of the 1D wave equation,

$$\mathfrak{h} \geq \frac{1}{2} \|\epsilon \delta_{t-} u\|_{\mathbb{U}_N}^2 - \frac{\gamma_0^2 k^2}{8} \|\delta_{x-} \delta_{t-} u\|_{\mathbb{U}_N}^2 \geq \frac{1}{2} \|\epsilon \delta_{t-} u\|_{\mathbb{U}_N}^2 - \frac{\gamma_0^2 k^2}{2h^2} \|\delta_{t-} u\|_{\mathbb{U}_N}^2 \quad (7.150)$$

$$= \frac{1}{2} \sum_{l=0}^N h (\epsilon_l^2 - \lambda^2) (\delta_{t-} u_l)^2 \quad (7.151)$$

where $\lambda = \gamma_0 k/h$. This expression can only be non-negative under the condition

$$\lambda \leq \epsilon_{min} = \min_{l \in \mathbb{U}_N} \epsilon_l \quad (7.152)$$

Under such conditions, one can, as before, go further and find bounds on solution growth, and thus the above serves as a stability condition for scheme (7.147).

Frequency Domain Behaviour

The scheme (7.147), even when condition (7.152) is satisfied with equality, behaves rather poorly, in that bandwidth is wasted, depending on the variation of the density—the larger the variation, the less of the spectrum is used. See Figure 7.18, for some plots of typical output spectra, under different choices of the function $\epsilon(x)$. Such behaviour is definitely audible, leading to a low-passing effect on resulting sound output, and is to be avoided at all costs. One way of approaching this problem is to design implicit schemes; a better idea, however, given that the problem itself exhibits spatial variation, is to introduce coordinate changes.

Using Stretched Coordinates

In essence, the low-passing effect which occurs for the scheme (7.147) is another example of the same behaviour which occurs in explicit schemes for LSI systems when the stability condition is not satisfied with equality—see, e.g., the case of schemes for the wave equation operating away from the CFL condition, as discussed in §6.2.3. It is possible to make sense of this geometrically in terms of the “regions of dependence” arguments used in this case, and as illustrated in Figure 7.19(a). (The reader may wish to refer to Figure 6.9 in the previous chapter, which deals with similar issues for schemes for the wave equation.)

Considering the model problem (7.144), one may say, roughly, that at a given point x , the wave speed is, locally, $\gamma = \gamma_0/\epsilon(x)$, and thus the region of dependence varies from point to point. At the point x at which ϵ takes its minimum value, the region of dependence is thus largest, and, as the numerical region of dependence of the scheme must include that of the model problem (for convergence, and stability), on a uniform grid, the minimum grid spacing is determined by ϵ at this point—energy analysis leads to exactly this result, as given in (7.152). At other grid points, however, the grid spacing is larger than this, and thus the scheme exhibits dispersive behaviour over the majority of the domain.

The natural means of attacking this problem is to employ stretched coordinates which match the variation in the wave speed—see §5.3 for a discussion of some of the relevant algebraic machinery.

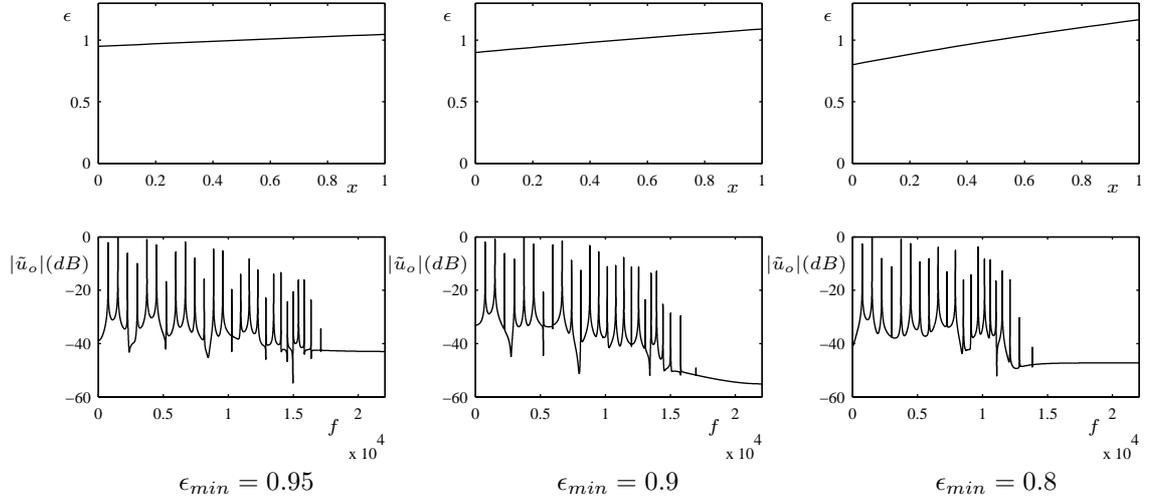


Figure 7.18: Output spectra $|\tilde{u}_o(f)|$, in dB, for finite difference scheme (7.147) for the string of variable density, under different density profiles, as indicated in plots of $\epsilon(x)$, related to the density $\rho(x)$ and its average value ρ_0 through $\rho_l = \rho_0 \epsilon^2(x)$. The sample rate was chosen as $f_s = 44100$ Hz, and $\gamma_0 = 1500$, and the bound (7.152) is satisfied as close to equality as possible. The initial condition is of plucked type, using a raised cosine of amplitude 1, centered at $x = 0.3$, of half-width 0.1, and the readout position is taken at $x = 0.8$.

Consider a dimensionless coordinate $\alpha(x)$, defined in terms of the dimensional coordinate x as

$$\alpha(x) = \frac{\int_0^x \epsilon(\eta) d\eta}{\int_0^L \epsilon(\eta) d\eta} \quad (7.153)$$

The spatially non-dimensionalized form of (7.144) then becomes

$$\epsilon u_{tt} = \gamma_0^2 (\epsilon u_\alpha)_\alpha \quad (7.154)$$

A simple finite difference scheme for the above equation is

$$[\epsilon] \delta_{tt} u = \gamma_0^2 \delta_{\alpha+} (\mu_{\alpha-} \epsilon \delta_{\alpha-} u) \quad (7.155)$$

Here, the exact form of the discrete approximation to ϵ on the left is left unspecified for the moment, and indicated as the grid function $[\epsilon] = [\epsilon]_l$.

A discrete energy follows, as before, through an inner product with δ_t , now in α coordinates, i.e.,

$$\delta_{t+} \mathfrak{h} = \mathfrak{b} \quad \text{with} \quad \mathfrak{h} = \mathfrak{t} + \mathfrak{v} \quad (7.156)$$

and

$$\mathfrak{t} = \frac{1}{2} \|\sqrt{[\epsilon]} \delta_t u\|_{\mathbb{U}_N}^2 \quad \mathfrak{v} = \frac{\gamma_0^2}{2} \langle \mu_{\alpha-} \epsilon \delta_x u, e_{t-} \delta_x u \rangle_{\overline{\mathbb{U}}_N} \quad (7.157)$$

It then follows, after manipulations similar to those performed previously, that under conservative conditions,

$$\mathfrak{h} \geq \frac{1}{2} \|\sqrt{[\epsilon]} \delta_t u\|_{\mathbb{U}_N}^2 - \frac{\lambda^2}{2} \|\sqrt{\mu_{\alpha\alpha}} \epsilon \delta_t u\|_{\mathbb{U}_N}^2 = \frac{1}{2} \sum_{l=0}^N h ([\epsilon] - \lambda^2 \mu_{\alpha\alpha} \epsilon) (\delta_{t-} u_l)^2 \quad (7.158)$$

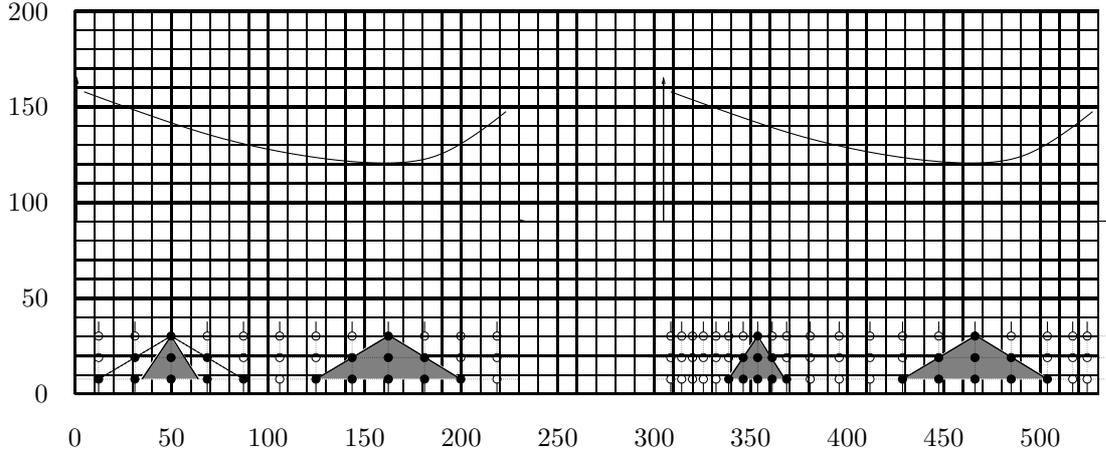


Figure 7.19: For a string of variable density (as indicated by the function $\epsilon(x)$, at top), the local wave speed, and thus the region of dependence will be variable. For an explicit scheme over a uniformly sampled grid, as shown in (a), at bottom, the numerical region of dependence of the scheme only matches that of the model system (indicated in grey) at the location at which ϵ is a minimum. Coordinate transformation and a regions of dependence in a scheme for a string of spatially-varying density (top). In (b), a coordinate transformation is applied such that the numerical and model regions of dependence coincide.

here, then, the natural choice of $[\epsilon] = \mu_{\alpha\alpha}\epsilon$ leads to the stability condition

$$\lambda \leq 1 \quad (7.159)$$

which is independent of ϵ —the coordinate transformation has removed such a dependence.

This scheme now exhibits much better behaviour in the frequency domain—see Figure 7.20. The audio bandwidth is filled, barring the slight loss incurred near the Nyquist, due to quantization of the number of grid points to an integer (i.e., the condition $\lambda \leq 1$ is satisfied close to equality).

7.9.2 Bars of Varying Cross-sectional Area

Much more relevant, at least in musical acoustics, is the case of the bar of variable cross-sectional area. For some percussion instruments, such as, e.g., the marimba, bars are tuned, generally by cutting an arch-like shape into the bars. See, e.g., [96] for an overview. The effects of variations in cross sectional area on the modal frequencies are complex, to say the least.

In the case of the uniform ideal bar, there was no need, from a simulation point of view, to enter into the details of various physical parameters (in particular the moment of inertia)—all the relevant parameters, including bar length, could be bundled into a single stiffness parameter κ . When the cross section is varying, more care needs to be taken, and it is useful to begin again from a dimensional form, which is:

$$\rho A(x)u_{tt} = -(EI(x)u_{xx})_{xx} \quad (7.160)$$

In the spatially-varying case, both A and the moment of inertia I will be functions of some characteristic thickness, which is dependent on x . It is assumed, here, that the material density ρ and Young's modulus E are not variable (though one could permit such variations as well).

It is perhaps most useful to concentrate on the case of a bar of rectangular cross section, with width b and height $H(x) = H_0\phi(x)$. H_0 is a reference thickness, and $\phi(x)$ represents the variations

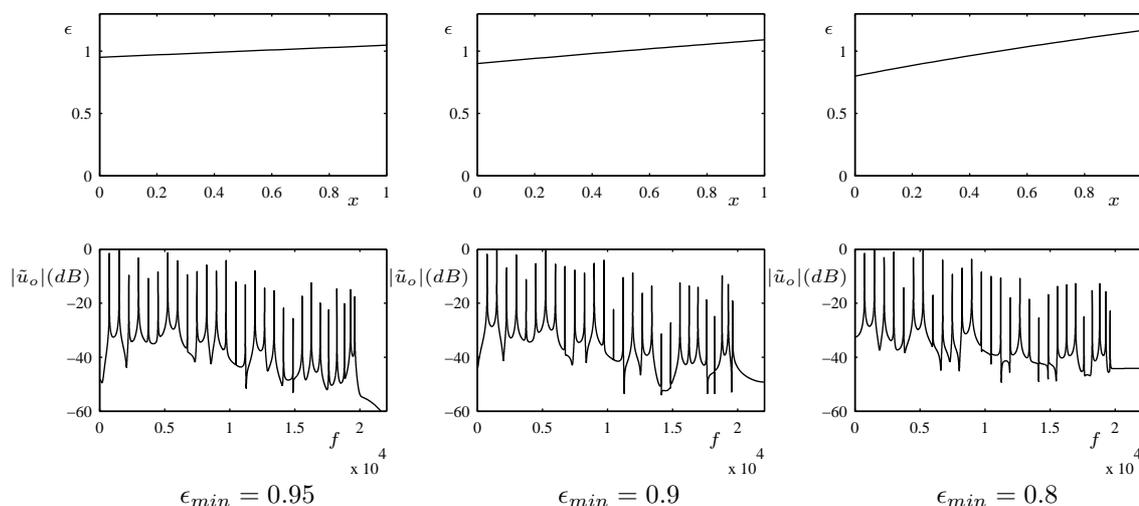


Figure 7.20: Output spectra for finite difference scheme (7.147) for the string of variable density, under different density profiles. The other conditions are otherwise the same as in the caption to Figure 7.18.

about this value. For such a bar, the area and moment of inertia are given by

$$A = bH_0\phi(x) \quad I = \frac{1}{12}bH_0^3\phi^3 \quad (7.161)$$

Under these conditions, when nondimensionalized, (7.160) becomes

$$\phi u_{tt} = -\kappa_0^2 (\phi^3 u_{xx})_{xx} \quad (7.162)$$

where

$$\kappa_0^2 = \frac{EH_0^2}{12\rho L^4} \quad (7.163)$$

where, as previously, L is the length of the bar.

Arched Bars

As an example of the effect of the variation in cross-sectional area, consider the effect of variation in the depth of an arch in a bar of rectangular cross-section, as shown in Figure 7.21. The gross effect of removing material from the bar is to decrease the modal frequencies—thus pitch is decreased. The ratios of the various modal frequencies do not stay the same however, and in particular, the first two non-zero modal frequencies, which are primarily responsible for the perception of pitch, may be adjusted in this way such that their ratio becomes consonant (such as, say, 3:1).

Energy

The variable bar system possesses a conserved energy as well. Over the real line, $\mathcal{D} = \mathbb{R}$, one has

$$\frac{d\mathfrak{H}}{dt} = 0 \quad \text{with} \quad \mathfrak{H} = \mathfrak{T} + \mathfrak{V} \quad (7.164)$$

with

$$\mathfrak{T} = \frac{1}{2} \|\sqrt{\epsilon} u_t\|_{\mathbb{R}}^2 \quad \mathfrak{V} = \frac{\kappa_0^2}{2} \|\phi^{3/2} u_{xx}\|_{\mathbb{R}}^2 \quad (7.165)$$

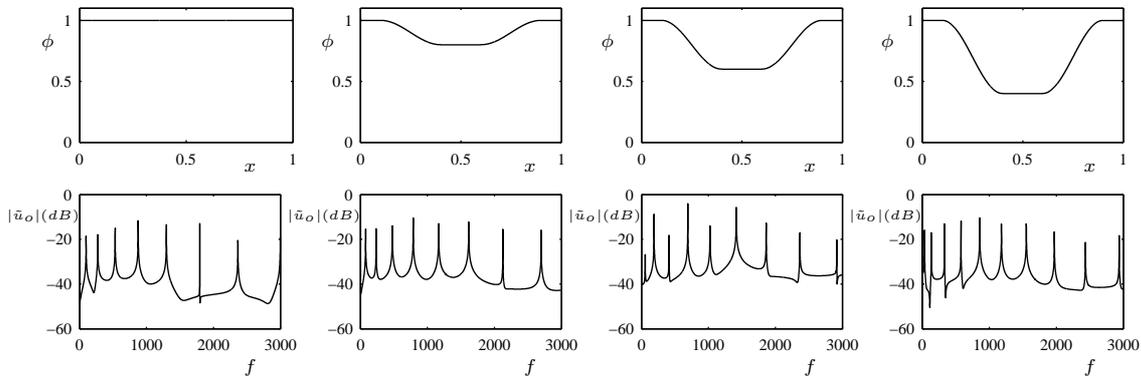


Figure 7.21: Output spectra for the bar of variable rectangular cross-sectional area, under different thickness profiles, as shown. Here, $\kappa_0 = 30$, and boundary conditions are of free type.

The fixed and clamped boundary conditions (7.13a) and (7.13b) continue to lead to exact conservation. The free condition, of great importance in mallet percussion instruments, must be modified to

$$u_{xx} = (\phi^3 u_{xx})_{xx} = 0 \quad (7.166)$$

at an endpoint—the condition reduces to (7.13c) when ϕ is constant at an endpoint (also generally the case for percussion).

Loss in the spatially-varying case will not be discussed here, though its effect on the energy balance above leads to strict dissipation—this effect carries over to the discrete case as well, and has no effect on numerical stability. See Problem 7.11.

Finite Difference Schemes

One can proceed immediately from the nondimensionalized form (7.162) above to a finite difference scheme:

$$[\phi] \delta_{tt} u = -\delta_{xx} (\phi^3 \delta_{xx} u) \quad (7.167)$$

Again, some unspecified second-order approximation $[\phi]$ has been employed. This scheme also may be shown to be stable, using energy techniques, under the condition

$$\mu = \frac{k\kappa_0}{h^2} \leq \frac{1}{2} \min \left(\sqrt{\frac{[\phi]}{\mu_{xx}(\phi^3)}} \right) \approx \frac{1}{2\phi_{max}} \quad (7.168)$$

The approximate bound above may be made exact through the proper choice of the approximation $[\phi]$ —see Problem ??.

As might be gathered from the discussion of the string of variable density in the previous section, this scheme performs terribly! For even very small variations in thickness, the spectrum of the output can be bandlimited to below one-quarter the sample rate. As such, this scheme is not attractive for synthesis—indeed, for larger variations in cross-sectional area, dispersion is so extreme that the scheme is practically useless—see Figure 7.22. Again, one may introduce a stretched coordinate $\alpha(x)$, in order to accommodate the variations in cross-section. Under the coordinate transformation

defined by

$$\alpha(x) = \frac{1}{\alpha_{av}} \int_0^x \frac{1}{\sqrt{\phi(\eta)}} d\eta \quad \text{with} \quad \alpha_{av} = \int_0^1 \frac{1}{\sqrt{\phi(\eta)}} d\eta \quad (7.169)$$

the system (??) becomes, using the transformed derivatives as discussed in §5.3,

$$\phi^{3/2} u_{tt} = -\frac{\kappa_0^2}{\alpha_{av}^4} \left(\phi^{-1/2} \left(\phi^{5/2} \left(\phi^{-1/2} u_\alpha \right)_\alpha \right)_\alpha \right) \quad (7.170)$$

(Notice, in particular, that the effect of the variation is now balanced on both side of the equation, in that an equal power of ϕ appears in both terms.) The related finite difference scheme is now

$$[\phi^{3/2}] \delta_{tt} u = -\frac{\kappa_0^2}{\alpha_{av}^4} \delta_{\alpha+} \left(\mu_{\alpha-} \phi^{-1/2} \delta_{\alpha-} \left(\phi^{5/2} \delta_{\alpha+} \left(\mu_{\alpha-} \phi^{-1/2} \delta_{\alpha-} u \right) \right) \right) \quad (7.171)$$

and the stability condition, again following from energy analysis, is

$$\mu \leq \frac{\alpha_{av}^2}{2} \min \left(\sqrt{\frac{[\phi^{3/2}]}{\mu_{\alpha+} ((\mu_{\alpha+} \phi^{5/2})(\mu_{\alpha-} \phi^{-1/2})^2)}} \right) \approx \frac{\alpha_{av}^2}{2} \quad (7.172)$$

Again, with a proper choice of $[\phi^{3/2}]$, the approximate bound above may be made exact. See Problem ??.

The improvement, relative to scheme (7.167) is dramatic—see Figure 7.22. The bandwidth of the computed solution is nearly full, regardless of the degree if variation in cross-sectional area. The problem of numerical dispersion, i.e., modal mistuning, remains, however, just as in the case of the uniform bar (see page 168). For finer modelling, one could well employ an implicit generalization of either of schemes (7.167) or (7.171), as has been done by Chaigne and Doutaut [116].

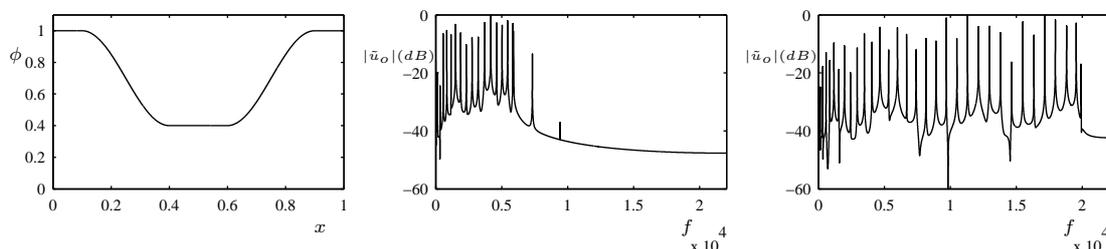


Figure 7.22: Output spectra for the bar of variable rectangular cross-sectional area, under the arched thickness profile, at left. Here, $\kappa_0 = 30$, boundary conditions are of free type, and the sample rate is $f_s = 44\ 100$ Hz. In the center panel is shown the spectrum of sound output for scheme (7.167), and in the right panel, that of output from scheme (7.171).

7.10 Problems

Problem 7.1 An explicit formula for the modal frequencies of an ideal bar under simply supported conditions at each end is given in (7.17), and an implicit formula for those the bar under clamped/simply supported conditions in (7.18). Derive implicit formulae for the remaining combinations of boundary conditions given in (7.13), namely: clamped/free, clamped/clamped, free/free, and simply supported/free. Show that in all cases, the distribution of modal frequencies ω_p approaches $\omega_p = \kappa \pi^2 p^2$ in the limit as p becomes large. Deduce that the number of degrees of freedom of the bar, as given in (7.19), is approximately independent of the choice of boundary condition.

Problem 7.2 Given the energy balance for the semi-infinite ideal bar from (7.12), show that the following conditions also imply losslessness. In all cases, the total conserved energy will be of the form $\mathfrak{H} + \mathfrak{H}_B$, where \mathfrak{H} is as defined in (7.10) (limited to the domain $\mathcal{D} = \mathbb{R}^+$, and where \mathfrak{H}_B is a term representing stored energy at the boundary point $x = 0$). The parameter α is constrained to be non-negative.

(a) $u_{xx}(0, t) = 0$ and $u_{xxx}(0, t) + \alpha u(0, t) = 0$.

(b) $u(0, t) = 0$ and $u_{xx}(0, t) - \alpha u_x(0, t) = 0$.

(c) $u_{xx}(0, t) = 0$ and $u_{xxx}(0, t) + \alpha u_{tt}(0, t) = 0$.

Furthermore, show that the following termination is strictly dissipative:

(d) $u_{xx}(0, t) = 0$ and $u_{xxx}(0, t) + \alpha u_t(0, t) = 0$.

To what physical combinations of masses, springs and dashpots do these terminations correspond? Attempt to illustrate each termination with a diagram.

Problem 7.3 Apply von Neumann stability analysis to the implicit scheme (7.42) for the ideal bar equation. Show that, by introducing the ansatz $u_l^n = z^n e^{j\beta l h}$, a characteristic polynomial of the form

$$z + \left(-2 + \frac{16\mu^2 p^2}{1 - 2(1 - \theta)p^2} \right) + z^{-1} = 0 \quad (7.173)$$

results. Here, as in the case of the simpler scheme (7.21), $p = \sin^2(\beta h/2)$, and takes on values between 0 and 1 only. Using the stability condition for second order polynomials given in (2.14), show the stability conditions (7.44). Try to prove the same result using energy methods.

Problem 7.4 Consider the action of the operator δ_{xxxx} applied to the grid function u_l , defined over \mathbb{Z}^+ , under the free boundary condition (7.36c), and write $\delta_{xxxx}u_0$ and $\delta_{xxxx}u_1$ purely in terms of values of u_l over the domain interior.

Problem 7.5 Consider the implicit θ scheme (7.62) for the stiff string equation. By equating the number of degrees of freedom of the model, from (7.58), and the number of degrees of freedom of the scheme, from (7.64), find an optimal setting for θ in terms of γ , κ , and the sample rate f_s . (You will need such an expression for the optimal θ in order to complete Programming Exercise (7.1) below.)

Problem 7.6 Some physical models employ an even more refined model of frequency-dependent loss. Consider the following model of the stiff string, with three-parameter frequency-dependent loss:

$$u_{tt} = \gamma^2 u_{xx} - \kappa^2 u_{xxxx} - 2\sigma_0 u_t + 2\sigma_1 u_{txx} - 2\sigma_2 u_{txxx} \quad (7.174)$$

(a) By inserting a test solution of the form $u = e^{st+j\beta x}$, find the characteristic equation, and find the roots $s_{\pm}(\beta)$.

(b) Using $s_{\pm} = \sigma \pm j\omega$, find expressions for $\sigma(\beta)$ and $\omega(\beta)$, under the assumption that the parameters σ_0 , σ_1 and σ_2 are small.

(c) The conditions

$$\sigma_0 \geq 0 \quad \sigma_1 \geq 0 \quad \sigma_2 \geq 0 \quad (7.175)$$

are sufficient for your expression for loss $\sigma(\beta)$ to be non-positive everywhere (i.e., so that all wavenumbers are damped). Derive necessary conditions. Hint: start by evaluating the expression $\sigma(\beta)$ for the limiting cases $\beta = 0$ and $\beta = \infty$. Sketch $\sigma(\beta)$ for a representative case where the sufficient condition above is violated, but the necessary condition is satisfied; you should be able to produce loss curves which are not-monotonic as a function of wavenumber.

(d) Supposing that the equation above is defined over the infinite interval $\mathcal{D} = \mathbb{Z}$, find an expression for the energy balance of the string of the form

$$\frac{d\mathfrak{H}}{dt} = -Q \quad (7.176)$$

and show that, under the sufficient conditions above, Q is non-negative. Can you show that Q is non-negative under the necessary conditions you derived in (c) above?

Problem 7.7 Consider the following implicit scheme for the stiff string model with three-parameter frequency-dependent loss described in the last problem:

$$\delta_{tt}u = \gamma^2 \delta_{xx}u - \kappa^2 \delta_{xxxx}u - 2\sigma_0 \delta_t u + 2\sigma_1 \delta_t \delta_{xx}u - 2\sigma_2 \delta_t \delta_{xxxx}u \quad (7.177)$$

- (a) Using von Neumann analysis (i.e., by inserting a test solution of the form $u_l^n = z^n e^{jl\beta h}$, find the characteristic polynomial for the scheme, which will be a quadratic in z .
 (b) Show that, if the conditions (7.175) are satisfied, the stability condition for the scheme is unchanged from (7.60) for the stiff string without loss.
 (c) Write the scheme in vector matrix form

$$\mathbf{A}u^{n+1} + \mathbf{B}u^n + \mathbf{C}u^{n-1} = 0 \quad (7.178)$$

where \mathbf{u}^n is a vector containing values of the grid function u_l^n . You may use the shorthand notations \mathbf{D}_{xx} and \mathbf{D}_{xxxx} to represent the matrix form of the operators δ_{xx} and δ_{xxxx} , thus assuming that boundary conditions are taken into account.

Problem 7.8 Consider the following variant of the stiff string model:

$$u_{tt} = \gamma^2 u_{xx} - \kappa^2 u_{xxxx} - 2\sigma_0 u_t - b u_{ttt} \quad (7.179)$$

which differs from the model (7.67) in the treatment of the frequency-dependent loss term.

- a) By inserting a test solution of the form $u = e^{st+j\beta x}$, show that the characteristic equation for this model is

$$bs^3 + s^2 + 2\sigma s + \omega_0^2 \beta^2 + \kappa^2 \beta^4 = 0 \quad (7.180)$$

- b) Show that, for large values of the wavenumber β , the roots of the characteristic equation must approach the roots of

$$s^3 + \frac{\kappa^2}{b} \beta^4 = 0 \quad (7.181)$$

- c) Prove that, given that b and β are real, at least one of the roots of the characteristic equation must have a positive real part in the limit as β becomes large. How does your analysis depend on the sign of b ? What can you conclude about the validity of such a model?

Problem 7.9 Prove condition stability condition (7.60) for the difference scheme (7.59) for the stiff string, using von Neumann analysis.

Problem 7.10 Reconsider the stiff string model given in (7.179) above, defined over $\mathcal{D} = \mathbb{R}$. By taking an inner product with u_t , find an energy balance for the equation, of the form

$$\frac{d\mathfrak{H}}{dt} = -Q \quad (7.182)$$

Are \mathfrak{H} and Q non-negative?

Problem 7.11 Consider the spatially-varying bar model, in the nondimensionalized form given by

$$\phi u_{tt} = -\kappa_0^2 (\phi^3 (u_{xx} + \sigma u_{txx}))_{xx} \quad (7.183)$$

defined over the entire real line, $\mathcal{D} = \mathbb{R}$, where $\sigma \geq 0$ is a loss parameter. (Such a model has been used to describe the vibration of non-uniform xylophone bars [116].)

- (a) Show that the energy balance (7.164) may be generalized to

$$\frac{d\mathfrak{H}}{dt} = -Q \quad (7.184)$$

and determine Q , showing that it is non-negative, and thus that the system is strictly dissipative.

- (b) Design a scheme for the lossy model above, using a centered difference approximation δ_t for the extra time derivative in the loss term. Show that this scheme will be dissipative, and thus numerically stable, in the same way as for the model problem.

7.11 Programming Exercises

Exercise 7.1 Create a matlab script which calculates and plots the phase velocity of the stiff string system (7.48), as well as the numerical phase velocity of the implicit scheme (?), for any value of

κ , γ and θ , over the band $f \in [0, f_s/2]$. You should assume that the grid spacing h_{min} is chosen at the stability limit, from (??). Assuming simply supported conditions, your code should also generate a list of the modal frequencies of the model system over the same band, as well as the modes of scheme (??). Verify, using the value for θ as a function of γ , κ and f_s (see Problem 7.5 above) that deviations in both the numerical phase velocity and scheme modal frequencies from those of the model system are minimized under these conditions.

Chapter 8

Nonlinear String Vibration

In Chapter 7, a general model of linear string motion was presented. This model, which holds under low-amplitude vibration conditions, is sufficient in many cases of musical interest, but not all. If vibration amplitude becomes large (under, say, high amplitude striking or plucking conditions), various nonlinear effects begin to appear, and can become perceptually dominant. The effect which is perhaps the most familiar to the reader will be that of the pitch glide, which is common across many instruments, and not merely strings (it occurs in struck percussion instruments as well—see §13.1.4). Under a high amplitude strike or pluck, there is often a downward change in the pitch of the string, due to increased tension in string, (or, equivalently, to the increased length of the deformed string). Such an effect, often called tension modulation [255, 269, 81], cannot be captured by a linear model, for which modal frequencies are, by linear system theory, fixed. Other more subtle phenomena also play an important perceptual role. The generation of audible so-called phantom partials [59, 15] in piano strings under high striking velocities is due to coupling between longitudinal and transverse vibration, and beating can result from the instability of motion of a nonlinear string in a single polarization, which is a purely three-dimensional effect.

The development here, proceeding from simpler models to the more general, is the reverse of the usual treatment in acoustics. A useful starting point, and perhaps the simplest nonlinear partial differential equation in musical acoustics, is the Kirchhoff-Carrier model of string vibration, introduced in §8.1, which models the effect of tension modulation, and thus gives rise to pitch glide phenomena. Finite difference schemes and modal methods are discussed. In order to deal with more realistic cases of string vibration, more general models in both one and two polarizations are introduced in §8.2 and §8.3, respectively. Finite difference schemes are developed, and various effects of musical interest such as phantom partials and whirling are examined.

References for this chapter include: [159, 282, ?, 270, 269, 255, 151, 160, 5, 59, 186]

8.1 The Kirchhoff-Carrier String Model

Perhaps the simplest generalization of the dynamics of a string beyond the linear variants that were detailed in the last chapter is that of Kirchhoff and Carrier [146, 49]:

$$\rho A u_{tt} = \left(T_0 + \frac{EA}{2L} \int_0^L u_x^2 dx \right) u_{xx} \quad (8.1)$$

Here, as for the case of the linear string, ρ and T_0 are the material density and tension, respectively, and u represents string displacement in a single polarization. In contrast to the simple linear model,

stiffness effects play a role, and thus Young's modulus E and the cross-sectional area A intervene as coefficients. The nonlinearity itself is of a very special form, involving an average of the square string slope over the length of the string—as one might guess, this is a form arrived at through quite a number of simplifying assumptions. The nonlinearity: the entire It also should be clear that the string is, in this case, well-defined only over a finite interval, in this case, $x \in [0, L]$. The equation (8.1) results from a simplification of a more complex system (to be discussed in §8.2), under various assumptions, the most important of which is that $EA \gg T_0$, which is true for nearly any string of musical interest. The full derivation is readily available in the literature—see, e.g., the classic papers by Anand [4], Narasimha [177]. Due to the simplicity of the nonlinear term, is heavily used for the analysis of nonlinear phenomena [73, 74, 128, 217]. Such an equation has also been used as a starting point in many studies in musical acoustics—see, e.g., [110, 156], and, in sound synthesis [82, 34].

It is useful to non-dimensionalize the system, again spatially, using a coordinate $x' = x/L$, but, because the system is nonlinear (i.e., its response is amplitude dependent), a non-dimensionalization of the dependent variable itself, through a choice of $u' = u/L$ is also advisable. The resulting system, now defined over the unit interval \mathbb{U} is

$$u_{tt} = \gamma^2 \left(1 + \frac{\alpha^2}{2} \|u_x\|_{\mathbb{U}}^2 \right) u_{xx} \quad (8.2)$$

where the parameters γ and α are defined by

$$\gamma = \frac{1}{L} \sqrt{\frac{T_0}{\rho}} \quad \alpha = \sqrt{\frac{EA}{T}} \quad (8.3)$$

The parameter γ should be familiar from the discussions in the previous two chapters, and α is a measure of the strength of stiffness versus tension effects in the string. Notice that the norm notation $\|\cdot\|$ has been introduced—see §5.1.3.

Given that effects of stiffness have appear in this model (notice the presence of Young's modulus above), one might wonder what has happened to the linear fourth order stiffness term which occurs, e.g., in the model of the ideal bar, as discussed in §7.1. Such a term can indeed be included, to yield a basic model of nonlinear bar vibration—see §8.1.7.

8.1.1 Amplitude-dependent Pitch

For nonlinear systems in musical acoustics, it is rare to be able to come to any conclusions about perceptual effects through simple inspection, but the Kirchhoff-Carrier model, due to its extreme simplicity, does permit some observations to be made. As a first step, rewrite (8.2) as

$$u_{tt} = \gamma^2 \mathfrak{G} u_{xx} \quad \text{with} \quad \mathfrak{G} = 1 + \frac{\alpha^2}{2} \|u_x\|_{\mathbb{U}}^2 \quad (8.4)$$

Here, the entire effect of the nonlinearity is represented by the scalar factor $\mathfrak{G} \geq 1$. Assuming that the time variation of \mathfrak{G} is slow (and it is always dangerous to make such assumptions), it can be viewed as a scaling factor to the parameter γ , and, thus, to the pitch. For instance, under fixed end conditions, the lowest frequency of vibration will be approximately $f_0 = \gamma \sqrt{\mathfrak{G}_{av}}/2$, where \mathfrak{G}_{av} represents an average value of \mathfrak{G} over several cycles— \mathfrak{G} itself varies rapidly! Note also that \mathfrak{G} is always greater than one, and approaches this value in limit as $\|u_x\|_{\mathbb{U}}$ becomes small, or, in other words, when vibration amplitude is small (i.e., the linear case). Thus, the greater the energy of the string, the greater the frequency of vibration—see Figure 8.1 for some examples of this behaviour.

One might expect, then, that loss will play an even more important role for such a nonlinear

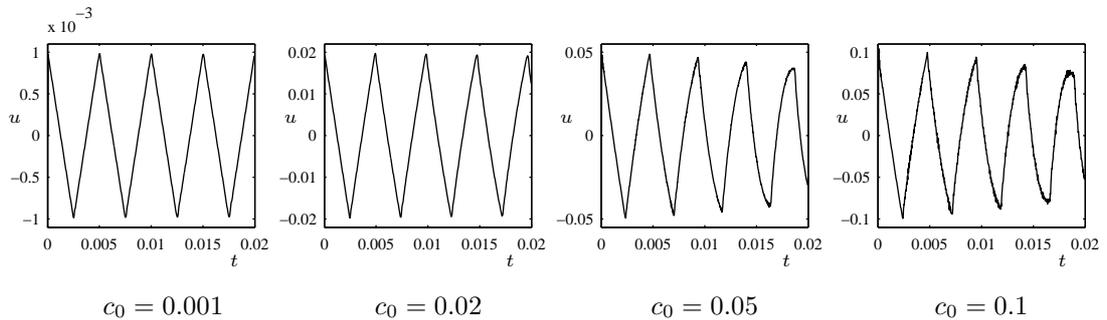


Figure 8.1: Plots of displacement of a string modelled by the Kirchhoff-Carrier equation (8.1), with $\gamma = 400$ and $\alpha = 10$, as a function of time t . In each case, the initial condition is a triangular function $c_{tri}(x)$, with $x_0 = 0.5$, and where the maximum displacements take on increasing values c_0 as indicated. The output displacement at the string center is plotted. Notice in particular that the gross rate of oscillation increases with the initial displacement, and also the deformation of the waveform away from the pure triangular form, which is characteristic of lossless linear strings.

system than for linear strings, as discussed in the previous chapter. Not only will it determine the rate of decay of vibration, but in this case, it will have the additional effect of altering the pitch over time, as the total energy of the system (and thus \mathfrak{G} decreases). See §8.1.5.

8.1.2 Energy Analysis and Boundary Conditions

Even though the Kirchhoff-Carrier equation (8.2) is the result of several approximations, it still possesses a conserved energy. It is possible to proceed as for the wave equation, and take an inner product with u_t , the velocity, over $\mathcal{D} = \mathbb{U}$, to get the energy balance

$$\frac{d\mathfrak{H}}{dt} = \gamma^2 \left(1 + \frac{\alpha^2}{2} \|u_x\|_{\mathbb{U}}^2 \right) u_t u_x \Big|_0^1 \tag{8.5}$$

with

$$\mathfrak{H} = \mathfrak{T} + \mathfrak{V} \quad \text{and} \quad \mathfrak{T} = \frac{1}{2} \|u_t\|_{\mathbb{U}}^2 \quad \mathfrak{V} = \frac{\gamma^2}{2} \left(1 + \frac{\alpha^2}{4} \|u_x\|_{\mathbb{U}}^2 \right) \|u_x\|_{\mathbb{U}}^2 \tag{8.6}$$

Under Dirichlet or Neumann conditions at either end point, the system is again exactly conservative. Note that the expression for the potential energy \mathfrak{V} is no longer a quadratic form, reflecting the nonlinearity¹ of system (8.2). It remains true, however, that the energy is non-negative—notice, in particular, that it is *higher* than that of the associated linear system (the wave equation).

8.1.3 Finite Difference Schemes

A direct finite difference approximation to the form of the Kirchhoff-Carrier equation given in (8.4), where the factor \mathfrak{G} has been introduced, is as follows:

$$\delta_{tt} u = \gamma^2 [\mathfrak{g}] \delta_{xx} u \tag{8.7}$$

¹Though it is true that if the energy of a system is not expressed as a quadratic form of the state, then the system must be nonlinear, the converse is not necessarily true. The von Karman system, for instance, which describes high-amplitude vibration of a thin plate, is highly nonlinear, yet the expression for the energy remains a quadratic form. See §??.

Here, $[\mathbf{g}] = [\mathbf{g}]^n$ is a scalar time series, intended as an approximation to \mathfrak{G} ; its form is left unspecified for the moment (this is indicated, as before, with the use of square brackets). There are obviously many ways of forming such an approximation. Perhaps the simplest is given by

$$[\mathbf{g}] = 1 + \frac{\alpha^2}{2} \|\delta_{x+}u\|_{\mathbb{U}}^2 \quad (8.8)$$

As it turns out, however, such a choice does not lead to an energy conservation property for the finite difference scheme, and it is difficult to deduce anything about numerical stability. In order to find an appropriate setting for \mathbf{g} , take the inner product of (8.7) with $\delta_t u$ over \mathbb{U} , to get

$$\langle \delta_t u, \delta_{tt}u \rangle_{\mathbb{U}} = \gamma^2 [\mathbf{g}] \langle \delta_t u, \delta_{xx}u \rangle_{\mathbb{U}} \quad (8.9)$$

$$= -\gamma^2 [\mathbf{g}] \langle \delta_t \delta_{x+}u, \delta_{x+}u \rangle_{\mathbb{U}} + \mathbf{b} = -\gamma^2 [\mathbf{g}] \delta_{t+} \frac{1}{2} \langle \delta_{x+}u, e_{t-} \delta_{x+}u \rangle_{\mathbb{U}} + \mathbf{b} \quad (8.10)$$

where here, the boundary term is given by

$$\mathbf{b} = \gamma^2 [\mathbf{g}] (\delta_{x+}u_N \delta_t u_N - \delta_{x-}u_0 \delta_t u_0) \quad (8.11)$$

Clearly, regardless of the choice of $[\mathbf{g}]$, the boundary term vanishes when Dirichlet or Neumann conditions are applied at either end of the string.

An energy conservation property, as mentioned earlier, does not follow for a choice of $[\mathbf{g}]$ such as the simple form given in (8.8). But consider the following choice:

$$[\mathbf{g}] = \mathbf{g} = 1 + \frac{\alpha^2}{2} \mu_{t+} \langle \delta_{x+}u, e_{t-} \delta_{x+}u \rangle_{\mathbb{U}} \quad (8.12)$$

where the square braces are removed, now that a choice on an approximation has been decided. This leads immediately to an energy conservation property, i.e.,

$$\delta_{t+} \mathfrak{h} = 0 \quad \text{with} \quad \mathfrak{h} = \mathfrak{t} + \mathfrak{v} \quad (8.13)$$

and

$$\mathfrak{t} = \frac{1}{2} \|\delta_{t-}u\|^2 \quad \mathfrak{v} = \frac{\gamma^2}{2} \left(\langle \delta_{x+}u, e_{t-} \delta_{x+}u \rangle_{\mathbb{U}} + \frac{\alpha^2}{4} \langle \delta_{x+}u, e_{t-} \delta_{x+}u \rangle_{\mathbb{U}}^2 \right) \quad (8.14)$$

which is directly analogous to the energy of the Kirchhoff-Carrier system, from (8.6). This energy will again be conserved, to machine accuracy, in a simulation—see Figure 8.2.

An Explicit Form

The quantity \mathbf{g} defined above in (8.12), when used in scheme (8.7), leads, apparently, to an explicit scheme— \mathbf{g} at time step n depends on values of u at time steps $n-1$, n and $n+1$. This may be remedied in the following way: First rewrite \mathbf{g} , using identity (2.7a), as

$$\mathbf{g} = 1 + \frac{\alpha^2}{2} \langle \delta_{x+}u, \mu_t \delta_{x+}u \rangle_{\mathbb{U}} = 1 + \frac{\alpha^2}{2} \|\delta_{x+}u\|_{\mathbb{U}}^2 + \frac{k^2 \alpha^2}{4} \langle \delta_{x+}u, \delta_{tt} \delta_{x+}u \rangle_{\mathbb{U}} \quad (8.15)$$

$$= 1 + \frac{\alpha^2}{2} \|\delta_{x+}u\|_{\mathbb{U}}^2 + \frac{\mathbf{g} k^2 \alpha^2}{4} \langle \delta_{x+}u, \delta_{xx} \delta_{x+}u \rangle_{\mathbb{U}} \quad (8.16)$$

$$= 1 + \frac{\alpha^2}{2} \|\delta_{x+}u\|_{\mathbb{U}}^2 - \frac{\mathbf{g} k^2 \alpha^2}{4} \langle \delta_{xx}u, \delta_{xx}u \rangle_{\mathbb{U}} + \mathbf{b}' \quad (8.17)$$

$$(8.18)$$

When the boundary term is zero (it will be under fixed conditions), \mathbf{g} can then be written as

$$\mathbf{g} = \frac{1 + \frac{\alpha^2}{2} \|\delta_{x+}u\|_{\mathbb{U}}^2}{1 + \frac{\mathbf{g} k^2 \alpha^2}{4} \langle \delta_{xx}u, \delta_{xx}u \rangle_{\mathbb{U}}} \quad (8.19)$$

This is something of a rarity among numerical methods—an explicit, conservative scheme for a

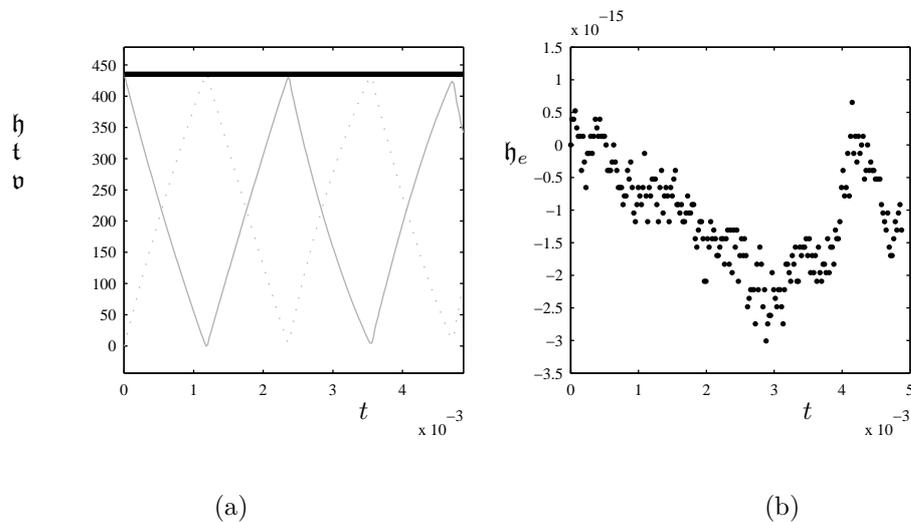


Figure 8.2: (a) Variation of the discrete potential energy v (solid grey line), discrete kinetic energy t (dotted grey line) and total discrete energy h (solid black line), plotted against time t , for the output of scheme (8.7) using fixed boundary termination. In this case, the values $\gamma = 400$, $k = 1/44100$ and $\lambda = 0.9$ were used, and the scheme was initialized with the a triangular distribution $c_{tri}(x)$, with $x_0 = 0.5$ and $c_0 = 0.035$. (b) Variation of the error in energy, defined, at time step n , as $h_e = (h - h^0) / h^0$, plotted as black points.

nonlinear distributed system. It is one of the few such examples to be discussed in this book. It is possible, in this case, due to the very simple form of the nonlinearity, which is scalar, putting it in line with explicit numerical methods for some lumped systems, particularly those where the nonlinearity is of third order—see §4.2.1.

Stability

Given these forms of the kinetic and potential energy for scheme (8.7) with the choice of \mathbf{g} from (8.12), one important observation that one can make is that the potential energy v consists of two terms: one, due to linear effects, and a second due to nonlinear effects. The second term is always non-negative; this has a simplifying effect on stability analysis, which becomes nearly trivial, once one has examined the related scheme for the wave equation. .

The potential energy v in (8.14) above differs from that which appears in the analysis of the related finite difference scheme for the wave equation by a single term (the final one)—this term, which is not quadratic in the state variables, indicates the presence of a nonlinearity. It is, however, non-negative. Recalling the discussion of such terms in §2.4.3, it is not hard to see that if the energy for the linear scheme is non-negative under condition (6.63), then the addition of an extra non-negative term to the energy cannot affect this result. Thus condition (6.63) serves as a sufficient condition for non-negativity of energy, and, as one can show, a numerical stability condition

Numerical Oscillations

8.1.4 A Quasi-modal Form

Though modal techniques are, as a rule, strictly applicable only to linear and time-invariant systems, in this case, because of the scalar nature of the nonlinearity, one might suspect that an extension of modal techniques might be possible.

Suppose that fixed (Dirichlet) boundary conditions have been applied at either end of the string. It is valid, from Fourier theory arguments, to expand the solution $u(x, t)$ in the following form:

$$u(x, t) = \sum_{p=1}^{\infty} U_p(t) \sin(p\pi x) \quad (8.20)$$

Upon inserting this form in (8.4), it then results that

$$\sum_{p=1}^{\infty} \frac{d^2 U_p}{dt^2} \sin(p\pi x) = -\gamma^2 \mathfrak{G} \sum_{p=1}^{\infty} p^2 \pi^2 U_p \sin(p\pi x) \quad (8.21)$$

and, after multiplying through individually by the functions $\sin(p\pi x)$ and integrating over the domain $\mathcal{D} = \mathbb{U}$, one arrives at

$$\frac{d^2 U_p}{dt^2} = -\gamma^2 \mathfrak{G} p^2 \pi^2 U_p \quad \text{for } p = 1, 2, \dots \quad (8.22)$$

As far as \mathfrak{G} goes, through Parseval's relation, one may demonstrate that

$$\mathfrak{G} = 1 + \frac{\alpha^2}{2} \sum_{p=1}^{\infty} \frac{\pi^2 p^2}{2} U_p^2 \quad (8.23)$$

To this point, the modal description (8.21), accompanied by the alternative formulation of \mathfrak{G} above, is equivalent to the time domain description (8.4); no approximations have been made. The PDE, defined over the unit interval \mathbb{U} has been replaced by an infinite set of ODEs. In order to arrive at a formulation suitable for simulation, one may truncate the infinite set of ODEs to a finite number, say M , and apply a second time difference to approximate the second time derivative, i.e.,

$$\delta_{tt} U_p = -\gamma^2 \mathfrak{g} p^2 \pi^2 U_p \quad \text{for } p = 1, 2, \dots, M \quad (8.24)$$

Here, $U_p = U_p^n$ is to be seen as a time series, and $\mathfrak{g} = \mathfrak{g}^n$ is a time series which depends only on the U_p^n , $p = 1, \dots, M$. It is worth noting that here, only the time derivative has been approximated; spatial derivatives have been approximated, through Fourier series to what is sometimes called "spectral accuracy," which is, for all practical purposes, exact. In fact, this is an example of a spectral method—these methods, and their potential in sound synthesis, will be discussed briefly in Chapter 15.

A suitable choice for the \mathfrak{g} , again the discrete counterpart to \mathfrak{G} may be arrived at, as before, through energetic principles. Multiplying through each of the difference equations in (8.24) by $\delta_t U_p$, one arrives at

$$\delta_{t+} \frac{1}{2} (\delta_t U_p)^2 = -\frac{\gamma^2}{2} \mathfrak{G} p^2 \pi^2 \delta_{t+} (U_p e_{t-} U_p) \quad (8.25)$$

A choice of

$$\mathfrak{g} = 1 + \frac{\alpha^2}{2} \sum_{p=1}^M \frac{\pi^2 p^2}{2} \mu_{t+} (U_p e_{t-} U_p) \quad (8.26)$$

8.1.5 Loss and Pitch Glides

If a loss term is added to (??), i.e., if the equation is altered to

$$u_{tt} = \gamma^2 \mathfrak{G} u_{xx} - 2\sigma_0 u_t \quad (8.27)$$

with \mathfrak{G} defined as before, then, under conservative boundary conditions, the energy \mathfrak{H} will be monotonically decreasing, and the amplitude of vibration will thus decay just as in the linear case. What is more interesting, sonically, is the audible change in pitch, which is downwards, from the moment of a strike or pluck.

If one accepts the factor $\sqrt{\mathfrak{G}}\gamma/2$, time-averaged, as a crude estimate of instantaneous pitch, then it is not hard to deduce the following bound on \mathfrak{G} , purely from energetic considerations:

$$\mathfrak{G} \leq \sqrt{1 + \frac{2\alpha^2 \mathfrak{H}}{\gamma^2}} \quad (8.28)$$

Thus, \mathfrak{G} is bounded by a function of \mathfrak{H} , which itself decreases monotonically. Even though \mathfrak{G} itself is rapidly varying, its average value will decrease smoothly, leading to the observed pitch glide. See Problem 8.1.

8.1.6 A Digital Waveguide Form

Because tension is

8.1.7 A Nonlinear Bar Model

8.2 General Planar Nonlinear String Motion

The Kirchhoff-Carrier model (8.1) is certainly the simplest model of the nonlinear distributed string—it adequately reproduces the important pitch glide effect, and can be simulated with ease. It is worth noting, however, that in this model, string motion is still purely transverse. Coupling between longitudinal and transverse motion does, in fact, lead to perceptually important effects such as so-called phantom partials (see §8.2.1), and a more complex model is necessary.

A quite general model of string vibration (see, e.g., [174]), including both longitudinal and transverse motion in a single plane (in dimensional form) is the following:

$$\rho A u_{tt} = E A u_{xx} - (EA - T_0) \left(\frac{\partial \Phi}{\partial u_x} \right)_x \quad \rho A \zeta_{tt} = E A \zeta_{xx} - (EA - T_0) \left(\frac{\partial \Phi}{\partial \zeta_x} \right)_x \quad (8.29)$$

Here, $u(x, t)$ is again the transverse displacement of the string, and $\zeta(x, t)$ is the longitudinal displacement from its rest—see Figure 8.3(a) for a graphical representation of these quantities. The constants ρ , A , E and T_0 are as for the Kirchhoff-Carrier model, and the function Φ , which nonlinearly couples the two equations, is

$$\Phi = \sqrt{(1 + \zeta_x)^2 + u_x^2} - 1 - \zeta_x \quad (8.30)$$

Notice that the final term $-1 - \zeta_x$ has no influence on the dynamics of system (8.29)—it is included here so as to adjust the zero-point energy of the system as a whole. This will be discussed further in §8.2.2.

Using the same non-dimensionalization strategy as for the Kirchhoff-Carrier system (i.e., substituting $x' = x/L$, $u' = u/L$, and $\zeta' = \zeta/L$ in the above system, and removing primes), one arrives at

$$u_{tt} = \gamma^2 \alpha^2 u_{xx} - \gamma^2 (\alpha^2 - 1) \left(\frac{\partial \Phi}{\partial q} \right)_x \quad \zeta_{tt} = \gamma^2 \alpha^2 \zeta_{xx} - \gamma^2 (\alpha^2 - 1) \left(\frac{\partial \Phi}{\partial p} \right)_x \quad (8.31)$$

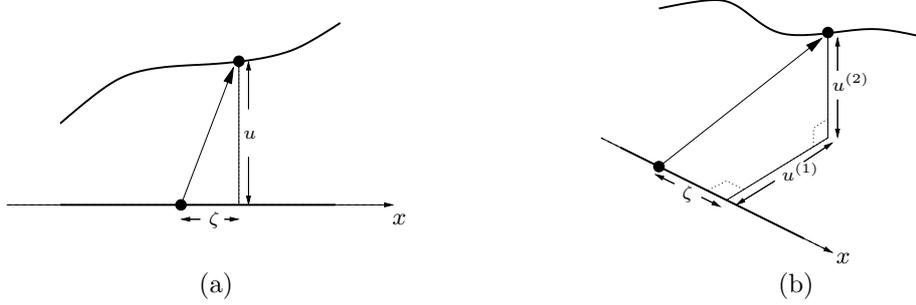


Figure 8.3: (a) Representation of the transverse displacement $u(x,t)$ and longitudinal displacement $\zeta(x,t)$ in the case of planar string motion, and (b) a similar diagram in the non-planar case, where the two transverse displacements are written as $u^{(1)}(x,t)$ and $u^{(2)}(x,t)$.

where, for simplicity, the auxiliary variables

$$p = \zeta_x \quad q = u_x \quad (8.32)$$

have been introduced, and, for Φ , one has

$$\Phi(p, q) = \sqrt{(1+p)^2 + q^2} - 1 - p \quad (8.33)$$

8.2.1 Coupling Between Transverse and Longitudinal Modes and Phantom Partials

8.2.2 Energy

Finding an expression for the energy of system (8.31) is straightforward—considering the problem defined over the infinite domain $\mathcal{D} = \mathbb{R}$, taking the inner product of the first of Eqs. (8.31) with u_t , and the second with ζ_t and adding the result leads to:

$$\mathfrak{H} = \mathfrak{T} + \mathfrak{V} = \text{constant} \quad (8.34)$$

with

$$\mathfrak{T} = \frac{1}{2} \|u_t\|_{\mathbb{R}}^2 + \frac{1}{2} \|\zeta_t\|_{\mathbb{R}}^2 \quad \mathfrak{V} = \frac{\gamma^2 \alpha^2}{2} \|q\|_{\mathbb{R}}^2 + \frac{\gamma^2 \alpha^2}{2} \|p\|_{\mathbb{R}}^2 - \gamma^2 (\alpha^2 - 1) \langle \Phi, 1 \rangle_{\mathbb{R}} \quad (8.35)$$

The energy \mathfrak{H} is non-negative (though this is slightly tricky to show—see Problem 8.2).

8.2.3 Series Approximations

Series approximated forms have played a significant role in the analysis of nonlinear systems such as the string model presented above; approximations to third or fourth order are commonly employed [174]. Recall also from the early discussion of the nonlinear oscillator in §??, that such approximated forms can lead to efficient and provably stable numerical methods. Such is the case here as well, but one must take great care in the type of approximation to be employed.

The function to be approximated in $\Phi(p, q)$. Consider three approximations, to second, third, and fourth order in p and q :

$$\Phi_2 = \frac{1}{2} q^2 \quad \Phi_3 = \frac{1}{2} q^2 - \frac{1}{2} p q^2 \quad \Phi_4 = \frac{1}{2} q^2 - \frac{1}{2} p q^2 + \frac{1}{2} q^2 p^2 - \frac{1}{8} q^4 \quad (8.36)$$

Approximation Φ_2 , to second order, leads immediately to an uncoupling of longitudinal and transverse motion— u and ζ individually satisfy the 1D wave equation, with speeds of γ and $\gamma\alpha$, respectively.

More interesting, in the present context, are the approximations Φ_3 and Φ_4 . It should be obvious that system (8.31), under the series approximations above, will also lead to a conservative system, with a modified energy function $\mathfrak{H}_3 = \mathfrak{T} + \mathfrak{V}_3$ or $\mathfrak{H}_4 = \mathfrak{T} + \mathfrak{V}_4$, where the modified expressions for the potential energy will be, in terms of p and q ,

$$\mathfrak{V}_3 = \frac{\gamma^2}{2} \|q\|_{\mathbb{R}}^2 + \frac{\gamma^2 \alpha^2}{2} \|p\|_{\mathbb{R}}^2 + \frac{\gamma^2 (\alpha^2 - 1)}{2} \langle p, q^2 \rangle_{\mathbb{R}} \quad (8.37)$$

$$\mathfrak{V}_4 = \frac{\gamma^2}{2} \|q\|_{\mathbb{R}}^2 + \frac{\gamma^2}{2} \|p\|_{\mathbb{R}}^2 + \frac{\gamma^2 (\alpha^2 - 1)}{8} \|q^2 + 2p\|_{\mathbb{R}}^2 - \frac{\gamma^2 (\alpha^2 - 1)}{2} \|qp\|_{\mathbb{R}}^2 \quad (8.38)$$

The third order approximation in particular has been used in finite difference piano sound synthesis applications [13], and the fourth order approximation (or variants—see below) is often used in studies of mode interaction [174].

Though useful for analysis purposes, the third and fourth order approximations are quite unphysical when it comes to the underlying Hamiltonian—in fact, due to series truncation, the potential energy terms can become negative, and unboundedly so! See Problem 8.3. This is obviously bad news if one is interested in developing a robust simulation routine, for if one can not bound the behaviour of the model system, it certainly will not be possible to do so for any derived numerical method.

A Special Series Approximation

The problem above is one of Hamiltonian truncation—as an alternative, consider the following approximation to $\Phi(p, q)$, similar to Φ_4 , but lacking one of the fourth order terms:

$$\Phi_4^* = \frac{1}{2} q^2 - \frac{1}{2} p q^2 - \frac{1}{8} q^4 \quad (8.39)$$

Now, the associated potential energy is

$$\mathfrak{V}_4^* = \frac{\gamma^2}{2} \|q\|_{\mathbb{R}}^2 + \frac{\gamma^2}{2} \|p\|_{\mathbb{R}}^2 + \frac{\gamma^2 (\alpha^2 - 1)}{8} \|q^2 + 2p\|_{\mathbb{R}}^2 \quad (8.40)$$

which is clearly non-negative if $\alpha \geq 1$ (which, again, is the case of most interest in the musical context). One might well ask in what sense the use of the approximation Φ_4^* is justified. One answer follows from the energetic properties to be discussed in the next section; another relates to the relative orders of magnitude of q_ξ and q_η in system (8.31). As noted by Anand [4] and Morse [174], under some conditions, p can be considered to be of the same order of magnitude as q^2 , and it is perhaps more natural, then, to use a homogeneous approximation, truncated to powers of, say, q . Given that the term $p^2 q^2$ in \mathfrak{V}_4 is clearly of sixth order in q , it is then justified to neglect it with respect to the term in q^4 . See [25] for more comments on this topic.

The system (8.31), under the choice Φ_4 reduces to

$$u_{tt} = \gamma^2 u_{xx} + \gamma^2 \frac{\alpha^2 - 1}{2} (q^3 + 2pq)_x \quad \zeta_{tt} = \gamma^2 \alpha^2 \zeta_{xx} + \gamma^2 \frac{\alpha^2 - 1}{2} (q^2)_x \quad (8.41)$$

where p and q are as defined in (8.32). This will be used as the starting point for the finite difference methods to be developed in the next section.

Wave Speeds and Computational Implications

A fundamental difficulty with system (8.41) arises from the differing speeds of the two coupled equations—in the absence of the nonlinear terms, the system reduces to two 1D linear wave equations, one of speed γ , and the other of speed $\gamma\alpha$ —for applications in musical acoustics, α can be of the order of 10 or larger. Such a system with differing wave speeds, and thus operating over distinct ranges of frequencies, is often referred to (in a very vague way) as stiff []. From the interpretation of the CFL condition in terms of a region of dependence (see Figure 6.9), an explicit scheme will clearly be strongly impacted by this. Consider, for example, *any* scheme for system (8.41) of the form

$$\delta_{tt}u = \gamma^2\delta_{xx}u + \dots \quad \delta_{tt}\zeta = \gamma^2\alpha^2\delta_{xx}\zeta + \dots \quad (8.42)$$

Regardless of the type of discretization applied to the nonlinear part, the scheme must still be capable of integrating the underlying linear system in a stable manner (because, under low amplitude conditions, the system (8.41) reduces to two linear wave equations). Thus two CFL-type conditions on the time step k and grid spacing h must be enforced:

$$h \geq \gamma k \quad h \geq \gamma\alpha k \quad (8.43)$$

The second condition is more strict than the first—it is clear that both conditions cannot be satisfied near their respective stability limits, and thus numerical dispersion, and, potentially a very severe numerical cutoff will result—see §6.2.3 for a discussion of the effects of operation of such schemes away from the CFL limit. The ramifications of such issues, as well as potential solutions will be discussed in the remainder of this section.

8.2.4 A Conservative Finite Difference Scheme

Disregarding the dire warning above about schemes of the form (8.42), such a scheme does form a good starting point for the development and analysis of schemes which are stable in the fully nonlinear case. The discretization of the nonlinear terms in (8.41) is extremely delicate, but, one may note that the nonlinear terms are at most cubic. One may suspect a link with the schemes for the particularly simple stable scheme for the cubic nonlinear oscillator, as discussed in §4.2. This is in fact the case, and, indeed, the extension to distributed systems such as the present case of the nonlinear string was one of the main motivations for the earlier preoccupation with such forms.

There are many possibilities for conservative schemes for system (8.41). Here is a particularly simple one:

$$\delta_{tt}u = \gamma^2\delta_{xx}u + \gamma^2\frac{\alpha^2-1}{2}\delta_{x+}(q^2\mu_t.q + 2q\mu_{tt}p) \quad (8.44a)$$

$$\delta_{tt}\zeta = \gamma^2\alpha^2\delta_{xx}\zeta + \gamma^2\frac{\alpha^2-1}{2}\delta_{x+}(q\mu_t.q) \quad (8.44b)$$

where, in analogy with the continuous case, the quantities p and q are defined as

$$p = \delta_{x-}\zeta \quad q = \delta_{x-}u \quad (8.45)$$

Notice in particular the use of time averaging operators on the right hand side of both updates—such a scheme is implicit, but, notice that just as in the case of schemes such as (??) for the cubic nonlinear oscillator, the unknowns appear only linearly, and thus there is no issue of existence or uniqueness of solutions. The full matrix update form will be given shortly.

Energy Analysis

To see the conservative property of this scheme, it is useful to perform the calculation explicitly. Taking the inner products of the two schemes (8.44) with $\delta_t u$ and $\delta_t \zeta$ respectively, over $\mathcal{D} = \mathbb{Z}$, and using summation by parts leads to:

$$\langle \delta_t u, \delta_{tt} u \rangle_{\mathbb{Z}} = \gamma^2 \langle \delta_t u, \delta_{xx} u \rangle_{\mathbb{Z}} + \gamma^2 \frac{\alpha^2 - 1}{2} \langle \delta_t u, \delta_{x+} (q^2 \mu_t q + 2q \mu_{tt} p) \rangle_{\mathbb{Z}} \quad (8.46)$$

$$= -\gamma^2 \langle \delta_t q, q \rangle_{\mathbb{Z}} - \gamma^2 \frac{\alpha^2 - 1}{2} \langle \delta_t q, q^2 \mu_t q + 2q \mu_{tt} p \rangle_{\mathbb{Z}} \quad (8.47)$$

$$\langle \delta_t \zeta, \delta_{tt} \zeta \rangle_{\mathbb{Z}} = \gamma^2 \alpha^2 \langle \delta_t \zeta, \delta_{xx} \zeta \rangle_{\mathbb{Z}} + \gamma^2 \frac{\alpha^2 - 1}{2} \langle \delta_t \zeta, \delta_{x+} (q \mu_t q) \rangle_{\mathbb{Z}} \quad (8.48)$$

$$= -\gamma^2 \alpha^2 \langle \delta_t p, p \rangle_{\mathbb{Z}} - \gamma^2 \frac{\alpha^2 - 1}{2} \langle \delta_t p, q \mu_t q \rangle_{\mathbb{Z}} \quad (8.49)$$

Adding the two equations, one arrives at

$$\delta_{t+} (\mathfrak{t} + \mathbf{v}_{lin}) + \gamma^2 \frac{\alpha^2 - 1}{2} \underbrace{(\langle \delta_t q, q^2 \mu_t q + 2q \mu_{tt} p \rangle_{\mathbb{Z}} + \langle \delta_t p, q \mu_t q \rangle_{\mathbb{Z}})}_{\Theta} = 0 \quad (8.50)$$

where the kinetic energy \mathfrak{t} , and the part of the potential energy corresponding to the underlying system \mathbf{v}_{lin} (i.e., the second order terms above), have been extracted as

$$\mathfrak{t} = \frac{1}{2} \|\delta_t u\|_{\mathbb{Z}}^2 + \|\delta_t \zeta\|_{\mathbb{Z}}^2 \quad \mathbf{v}_{lin} = \frac{\gamma^2}{2} \langle q, e_t - q \rangle_{\mathbb{Z}} + \frac{\gamma^2 \alpha^2}{2} \langle p, e_t - p \rangle_{\mathbb{Z}} \quad (8.51)$$

Notice that the terms on the right hand side depend only on the grid functions q and p , under the application of time difference and averaging operators—one may thus make liberal use of the identities (??), in order to write the term Θ as indicated in (8.50) as the difference of a nonlinear potential energy contribution, as

$$\Theta = \langle q \delta_t q, q \mu_t q \rangle_{\mathbb{Z}} + \langle q \delta_t q, 2 \mu_{tt} p \rangle_{\mathbb{Z}} + \langle \delta_t p, q \mu_t q \rangle_{\mathbb{Z}} \quad (8.52)$$

$$\stackrel{(2.22b), (2.22d)}{=} \frac{1}{2} \langle \delta_{t+} (q e_t - q), \mu_{t+} (q e_t - q) \rangle_{\mathbb{Z}} + \langle \delta_{t+} (q e_t - q), \mu_{t+} \mu_t - p \rangle_{\mathbb{Z}} + \langle \mu_{t+} (q e_t - q), \delta_{t+} \mu_t p \rangle_{\mathbb{Z}} \quad (8.53)$$

$$\stackrel{(2.23), (2.22c)}{=} \delta_{t+} \left(\frac{1}{4} \|q e_t - q\|_{\mathbb{Z}}^2 + \langle q e_t - q, \mu_t - p \rangle_{\mathbb{Z}} \right) \quad (8.54)$$

Thus, finally, one has numerical energy conservation, i.e.,

$$\delta_+ \mathfrak{h} = 0 \quad \text{with} \quad \mathfrak{h} = \mathfrak{t} + \mathbf{v}_{lin} + \mathbf{v}_{nonlin} \quad (8.55)$$

where \mathfrak{t} and \mathbf{v}_{lin} are as given in (8.51) above, and where \mathbf{v}_{nonlin} is defined as

$$\mathbf{v}_{nonlin} = \gamma^2 \frac{\alpha^2 - 1}{2} \left(\frac{1}{4} \|q e_t - q\|_{\mathbb{Z}}^2 + \langle q e_t - q, \mu_t - p \rangle_{\mathbb{Z}} \right) \quad (8.56)$$

Scheme (8.44) is but one among many in a family of conservative methods for the system (8.41)—see [25]. Stability properties vary greatly, and, one may lose the linear updating property. See Problems (??).

Non-negativity of Energy and Stability

As in the case of the systems examined in the previous chapters, once one has a numerical conservation property, the route to a stability condition is to find conditions under which the energy remains non-negative. To this end, note that by using identity (??), the total potential energy may

be written as

$$\mathbf{v} = \mathbf{v}_{lin} + \mathbf{v}_{nonlin} = \frac{\gamma^2}{2} \langle q, e_{t-q} \rangle_{\mathbb{Z}} \frac{\gamma^2 \alpha^2 k^2}{8} \|\delta_{t-p}\|_{\mathbb{Z}}^2 + \frac{\gamma^2}{2} \|\mu_{t-p}\|_{\mathbb{Z}}^2 + \gamma^2 \frac{\alpha^2 - 1}{2} \|\mu_{t-p} + \frac{1}{2} q e_{t-q}\|_{\mathbb{Z}}^2 \quad (8.57)$$

When $\alpha \geq 1$, one may go further and bound the potential energy from below, using inequality (??), as

$$\mathbf{v} \geq \frac{\gamma^2}{2} \langle q, e_{t-q} \rangle_{\mathbb{Z}} - \frac{\gamma^2 \alpha^2 k^2}{8} \|\delta_{t-p}\|_{\mathbb{Z}}^2 \quad (8.58)$$

$$\geq -\frac{\gamma^2 k^2}{8} \|\delta_{t-q}\|_{\mathbb{Z}}^2 - \frac{\gamma^2 \alpha^2 k^2}{8} \|\delta_{t-p}\|_{\mathbb{Z}}^2 \quad (8.59)$$

$$\geq -\frac{\gamma^2 k^2}{2h^2} \|\delta_{t-u}\|_{\mathbb{Z}}^2 - \frac{\gamma^2 \alpha^2 k^2}{2h^2} \|\delta_{t-\zeta}\|_{\mathbb{Z}}^2 \quad (8.60)$$

Finally, from the expression for total energy \mathfrak{h} , one arrives at

$$\mathfrak{h} \geq \frac{1}{2} \left(1 - \frac{\gamma^2 k^2}{h^2}\right) \|\delta_{t-u}\|_{\mathbb{Z}}^2 + \frac{1}{2} \left(1 - \frac{\gamma^2 \alpha^2 k^2}{h^2}\right) \|\delta_{t-\zeta}\|_{\mathbb{Z}}^2 \quad (8.61)$$

which is non-negative under conditions (8.43).

Numerical Behaviour

The stability conditions (8.43) are, as mentioned, rather restrictive. Using a typical audio sample rate f_s such as 44 100 Hz leads to trouble— $k = 1/f_s$ must be chosen so as to satisfy the second condition, leading normally to a very large grid spacing. As the grid spacing is used for both the transverse displacement as well as the longitudinal displacement, the number of degrees of freedom necessary for the transverse displacement (the modes of which in the audio range are far more numerous than for the longitudinal displacement) is grossly insufficient, and a very severe numerical cutoff is evident. See Figure ???. In fact, for most values of α which occur in practice, such a scheme is unusable at a typical audio sample rate. One obvious approach, and one taken by Bank [], is to use a very high audio sample rate (often on the order of 400 000 + Hz). But given that computational complexity scales with the square of the sample rate, this is obviously a technique of limited utility in synthesis. As a way around this difficulty, Bank has proposed using a modal representation for the longitudinal displacement, in conjunction with a finite difference scheme for the transverse displacement, which appears to work quite well.

On the other hand, various more elaborate schemes for the wave equation and the ideal bar have been discussed in §6.3 and §7.1.5, respectively, and one may put such techniques to use here as well.

8.2.5 An Improved Finite Difference Scheme

The problem with scheme (8.44), as mentioned above, is that the stability conditions (8.43) conflict. In addition, as it is the transverse displacement which is of primary importance in string vibration, one would like to be able to use a scheme which is of high accuracy for at least this part of the problem. To this end, consider the scheme

$$\delta_{tt}u = \gamma^2 \delta_{xx}u + \gamma^2 \frac{\alpha^2 - 1}{2} \delta_{x+} (q^2 \mu_t \cdot q + 2q \mu_{tt} p) \quad (8.62a)$$

$$(\theta + (1 - \theta) \mu_{x \cdot}) \delta_{tt} \zeta = \gamma^2 \alpha^2 \delta_{xx} \zeta + \gamma^2 \frac{\alpha^2 - 1}{2} \delta_{x+} (q \mu_t \cdot q) \quad (8.62b)$$

which differs from scheme (8.44) through an added implicit treatment of the equation for longitudinal displacement, parameterized by θ . Under low-amplitude (or linear conditions), this scheme decouples

into two independent schemes in u and ζ , the second of which is exactly the compact implicit method for the 1D wave equation as discussed in §6.3.2.

From the earlier treatment of this scheme, one may deduce that the stability conditions for scheme (8.62) are, in addition to the requirement $\theta \geq 1/2$, that

$$h \geq \gamma k \quad h \geq \frac{\gamma \alpha k}{\sqrt{2\theta - 1}} \quad (8.63)$$

Though this condition was derived using von Neumann methods, it may be shown to hold in the case of the nonlinear coupled scheme above. See Problem 8.4. The free parameter θ may now be set so that the bounds are identical, as

$$\theta = \frac{1 + \alpha^2}{2} \quad (8.64)$$

If the common bound is satisfied with equality, the result is very low dispersion in the “transverse” wave equation approximation (8.62a), and considerable dispersion in the longitudinal approximation (8.62b); in particular, there is not a severe numerical cutoff, as in the case of (8.44)—see Figure ??.

8.3 Nonplanar String Motion

In the previous sections, nonlinear string vibration models have been examined, where the motion is assumed to lie in a single plane. The extension to the case of non-planar motion is direct, at least on paper, but entirely new phenomena appear, related to the inherent instability of planar motion of a string—“instability” is here used not in the dynamic or numerical sense (referring to the explosive behaviour), but rather to mean that a string motion which is initially planar will become largely nonplanar under very small nonplanar perturbation. The resulting effect is often referred to as “whirling,” and has been studied by various authors [128, 219, 184], including in musical contexts [110].

8.3.1 Model Equations

The Kirchhoff-Carrier system (8.2) may be extended easily to the nonplanar case as

$$\mathbf{u}_{tt} = \gamma^2 \left(1 + \frac{\alpha^2}{2} \|\mathbf{u}_x\|_{\mathbb{U}}^2 \right) u_{xx} \quad (8.65)$$

which is again in spatially nondimensionalized form and defined over the unit interval $\mathcal{D} = \mathbb{U}$, where γ is again the wave speed, as defined in (??), and α is as for the planar case, i.e., as given by (??). The transverse displacement $\mathbf{u} = [u^{(1)}, u^{(2)}]$ is now a vector quantity with components $u^{(1)}$ and $u^{(2)}$ in two directions orthogonal to the string axis at rest.

The series-approximated nonlinear system which generalizes (8.41) is

$$\mathbf{u}_{tt} = \gamma^2 \mathbf{u}_{xx} + \gamma^2 \frac{\alpha^2 - 1}{2} (\mathbf{q} (|\mathbf{q}|^2 + 2p))_x \quad \zeta_{tt} = \gamma^2 \alpha^2 \zeta_{xx} + \gamma^2 \frac{\alpha^2 - 1}{2} (|\mathbf{q}|^2)_x \quad (8.66)$$

where now, $\mathbf{q} = \mathbf{u}_x$. See Figure 8.3(b) for a representation of the various displacements in the nonplanar case.

Energy

Energy analysis of the nonplanar systems above is very similar to the planar case—

8.3.2 Finite Difference Schemes

8.3.3 Instability of Planar Motion and Whirling Behaviour

8.4 Problems

Problem 8.1 Prove the bound (8.28) on the \mathfrak{G} , which describes the pitch glide for the Kirchhoff-Carrier system with a linear loss term in the following way:

(a) Given that $0 \leq \mathfrak{V} \leq \mathfrak{H}$, find a bound on $\|u_x\|_{\mathbb{U}}^2$ in terms of \mathfrak{H} .

(b) Given the bound on $\|u_x\|_{\mathbb{U}}^2$, and the fact that \mathfrak{G} is a strictly increasing function of $\|u_x\|_{\mathbb{U}}^2$

Problem 8.2 To show non-negativity of the energy \mathfrak{H} for the nonlinear string system (8.31), it suffices to show that the expression for the potential energy \mathfrak{V} , given in (8.35), is non-negative. Show that \mathfrak{V} may be rewritten as

$$\mathfrak{V} = \gamma^2 \langle \Phi, 1 \rangle_{\mathbb{R}} + \frac{\alpha^2}{2} \|\Phi + p\|_{\mathbb{R}}^2 \quad (8.67)$$

and argue that both terms are individually non-negative.

Problem 8.3 Show that the potential energy terms corresponding to the truncation of the nonlinearity in the string system (8.31) to third and fourth order can take on negative unbounded values, under some or all choices of α .

Problem 8.4 Consider the scheme (8.62) for the nonlinear string system (8.41); recall that this is a variant of scheme (8.44), in which the linear part of the scheme for longitudinal displacement is treated implicitly, with a free parameter θ . Show that the energetic analysis performed in pages 213 to 214 for scheme (8.44) extends to this case as well, with the difference that the linear part of the potential energy v_{lin} must now be written as

$$v_{lin} = \frac{\gamma^2}{2} \langle q, e_t - q \rangle_{\mathbb{Z}} + \frac{\gamma^2 \alpha^2}{2} \langle p, e_t - p \rangle_{\mathbb{Z}} - \frac{h^2(1-\theta)}{4} \|\delta_t - p\|_{\mathbb{Z}}^2 \quad (8.68)$$

Deduce conditions on non-negativity of conserved energy in this case, and show that they are in fact those given in (8.63).

8.5 Programming Exercises

Chapter 9

Acoustic Tubes

Wave propagation in acoustic tubes forms the basis of musical sound production in wind and brass instruments, as well as in the case of the human voice. There are various ways of defining what is meant by the word “tube.” In general, it can be described as an enclosure for which the length scale in one coordinate is significantly greater than in the others. Generally, in linear problems, the dynamics of the material filling the tube (air in musical applications) will satisfy the wave equation, in three dimensions. If it is true that the wavelengths of interest are longer than the length scale in the two “short” coordinate directions, then it is possible to simplify the dynamics to one dimension.

The simplest case of numerical methods for lossless cylindrical tubes is described by the 1D wave equation, which was covered in some depth in Chapter 6; it will thus be presented in an abbreviated form in §9.1. The more interesting case of the conical tube, of great interest wind instrument modelling, will be approached in §9.2. The coupling with models of toneholes and reeds is approached in §9.3 and §9.4, respectively. Finally, a general numerical model of the dynamics of acoustic tubes, described by Webster’s equation is presented in §9.5.

References for this chapter include: [226, 60, 277, 22, 164, 78, 140, 213, 62, 245, 141, 280, 279, 278, 175]

9.1 Cylindrical Tubes

9.2 Conical Tubes

9.3 Tonehole Models

9.4 Coupling with Reed Models

9.5 Webster’s Equation and the Vocal Tract

A good model of the dynamics of an acoustic tube of variable cross-sectional area $S(x)$, often referred to as Webster’s equation, is the following:

$$Su_{tt} = \gamma^2 (Su_x)_x \tag{9.1}$$

This equation is again assumed spatially non-dimensionalized, where again $u(x, t)$ represents a one-dimensional pressure field. Webster’s equation reduces to the 1D wave equation when S is a constant. The underlying assumptions are discussed in detail by Benade and Jansson [?]. Webster’s equation

appears in slightly different (equivalent) forms in the literature, namely

$$u_{tt} = \gamma^2 \left(u_{xx} + \frac{S_x}{S} u_x \right) \quad \text{and} \quad (9.2)$$

Like the equation of motion of the bar of variable cross-section, this is an example of a linear and time-invariant (LTI), but not generally shift-invariant system—see §5.1.1. As such, the usual frequency domain techniques, including von Neumann analysis and the extraction of phase and group velocity information are much more difficult, though a modal description persists—see §9.5.2

An energy balance follows for Webster's equation. Over the real line, one has

$$\frac{d\mathfrak{H}}{dt} = 0 \quad \text{with} \quad \mathfrak{H} = \mathfrak{T} + \mathfrak{V} \quad \mathfrak{T} = \frac{1}{2} \|\sqrt{S}u_t\|_{\mathbb{R}}^2 \quad \mathfrak{V} = \frac{\gamma^2}{2} \|\sqrt{S}u_x\|_{\mathbb{R}}^2 \quad (9.3)$$

which is identical to that of the wave equation, but in a norm weighted by S .

9.5.1 Boundary Conditions

9.5.2 Modes

9.5.3 Finite Difference Schemes

As in the case of the bar of variable cross-section, the number of choices of even simple varieties of finite difference schemes increases enormously, due to the spatial variation of the area function $S(x)$. It is useful to distinguish between the grid function S_l , which consists of sampled values of the continuous function $S(x)$ at locations $x = lh$, and a grid function $[S]_l$, which is some approximation to the continuous function $S(x)$, of an exact form yet to be determined—it is assumed that $[S]_l$ is a second-order accurate approximation to $S(x)$. A general finite difference scheme for Webster's equation can then be written as

$$[S]\delta_{tt}u = \gamma^2\delta_{x+}((\mu_{x-}S)(\delta_{x-}u)) \quad (9.4)$$

which may be expanded to give

$$u_l^{n+1} = \frac{\lambda^2(S_{l+1} + S_l)}{2[S]_l} u_{l+1}^n + \frac{\lambda^2(S_l + S_{l-1})}{2[S]_l} u_{l-1}^n + \left(2 - \frac{\lambda^2(S_{l+1} + 2S_l + S_{l-1})}{2[S]_l} \right) u_l^n - u_l^{n-1} \quad (9.5)$$

where $\lambda = \gamma k/h$ is again the Courant number for the scheme. It is straightforward to show that (9.4) is a second-order accurate approximation to Webster's equation. See Problem 9.1.

Energy analysis is direct, and similar to that carried out for the spatially-varying systems in §???. Over the real line, the numerical energy balance is

$$\delta_{t+}\mathfrak{h} = 0 \quad \text{with} \quad \mathfrak{h} = \mathfrak{t} + \mathfrak{v} \quad \mathfrak{t} = \frac{1}{2} \|\sqrt{[S]}\delta_{t-}u\|_{\mathbb{Z}}^2 \quad \mathfrak{v} = \frac{\gamma^2}{2} \langle (\mu_{x+}S)\delta_{x+}u, e_{t-}\delta_{x+}u \rangle_{\mathbb{Z}} \quad (9.6)$$

The stability condition, which follows immediately, is

$$\lambda \leq \min \left(\sqrt{\frac{[S]}{\mu_{xx}S}} \right) \approx 1 \quad (9.7)$$

Two things are to be noted here: First, the stability condition is approximately independent of the area function $S(x)$; this is quite distinct from the cases of the string of variable density, or the bar of variable cross-sectional area. The reason is that the gross wave speed itself remains γ everywhere within the tube, though dispersion does intervene. Second, the approximate stability condition above may be made exact under the choice of $[S] = \mu_{xx}S$. In this case, it is worth looking at the special form of the recursion when $\lambda = 1$:

$$u_l^{n+1} = \frac{2(S_{l+1} + S_l)}{S_{l-1} + 2S_l + S_{l+1}} u_{l+1}^n + \frac{2(S_l + S_{l-1})}{S_{l-1} + 2S_l + S_{l+1}} u_{l-1}^n - u_l^{n-1} \quad (9.8)$$

This special form, for which the value at the grid point l, n has disappeared from the recursion, should hint at the special case of the digital waveguide, as discussed in §???. And indeed it should, for this scheme may be viewed in terms of the Kelly-Lochbaum speech synthesis model, probably the first example of a scattering-based numerical method.

9.5.4 The Kelly-Lochbaum Speech Synthesis Model

As was mentioned early on, the Kelly-Lochbaum

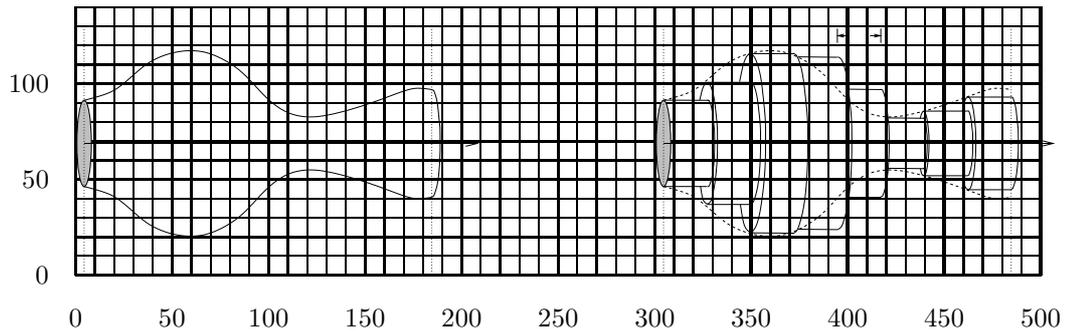


Figure 9.1: .

9.5.5 Coupling with a Glottis Model

9.6 Problems

Problem 9.1 Consider scheme (9.4) for Webster’s equation, and recall that the approximation $[S]$ is second-order accurate. Using the identity (??), show that it may be rewritten as

$$[S]\delta_{tt}u = \gamma^2 (\delta_x \cdot S \delta_x \cdot u + \mu_{x+} \mu_{x-} - S \delta_{xx} u) \tag{9.9}$$

Show that every approximation to either S or u in the above scheme is second-order accurate in space or time, and deduce that the scheme is second-order accurate as a whole.

9.7 Programming Exercises

Chapter 10

Grid Functions and Finite Difference Operators in 2D

This chapter is concerned with the extension of the difference operators introduced in Chapter 5 to two spatial dimensions. The 2D case is one of great interest in musical acoustics, given that many key components of musical instruments may be well described as such—for various percussion instruments such as drums, cymbals and gongs, it is the main resonating element, whereas in keyboard instruments and some stringed instruments, it serves as an auxiliary radiating element which imparts its own characteristic to the resulting sound. Perceptually speaking, the sound output from a 2D simulation is far richer than that of a 1D simulation. Part of this is due to the number of degrees of freedom, or modes which, in the linear case, is considerably larger, and of less regular a distribution than in 1D—sounds generated by 2D objects are generally inharmonic by nature. Beyond this, there are mechanisms at work, in particular in the nonlinear case, which lead to perceptual phenomena which have no real analogue in 1D; cymbal crashes are an excellent example of such behaviour. Most exciting of all, 2D physical modeling, long deemed to be an offline activity, is now coming within range of real time performance on commercially available hardware, thus all the more reason to spend some time on looking at the tools for simulation. That being said, much of the material presented in Chapter 5 extends in a natural way to 2D, and thus the presentation here will be as brief as possible, except when it comes to certain features which are particular to 2D.

10.1 Partial Differential Operators and PDEs in Two Space Variables

The single largest headache in 2D, both at the algorithm design stage, and in programming a working synthesis routine is, of course, problem geometry. Whereas 1D problems are defined over a domain which may always be non-dimensionalized down to the unit interval, in 2D, no such simplification is possible. As such, the choice of coordinates becomes important. Here, to keep the emphasis on basic principles, only two such choices, namely Cartesian and radial coordinates (certainly the most useful in musical acoustics) will be discussed. Despite this, it is worth keeping in mind that numerical simulation methods are by no means limited to such coordinate choices, though as the choice of coordinate system (generally governed by geometry) becomes more complex, finite difference methods lose a good deal of their appeal, and finite element methods (see Chapter 14 become an attractive option.

10.1.1 Cartesian and Radial Coordinates

Certainly the simplest coordinate system, and one which is ideal for working with problems defined over square or rectangular regions, is the Cartesian coordinate system, where a position is defined by the pair (x, y) . For problems defined over circles, radial coordinates (r, θ) , defined in terms of Cartesian coordinates by

$$r = \sqrt{x^2 + y^2} \quad \theta = \arctan(y/x) \tag{10.1}$$

may be more appropriate. See Figure 10.1(a) for an illustration of such coordinate systems.

In finite difference applications, Cartesian coordinates are undeniably much simpler to deal with, due to the symmetry between the x and y coordinate directions. Radial coordinates are trickier in some respects, especially due to the existence of a coordinate center, and also because differential operators exhibit a dependence on radius r .

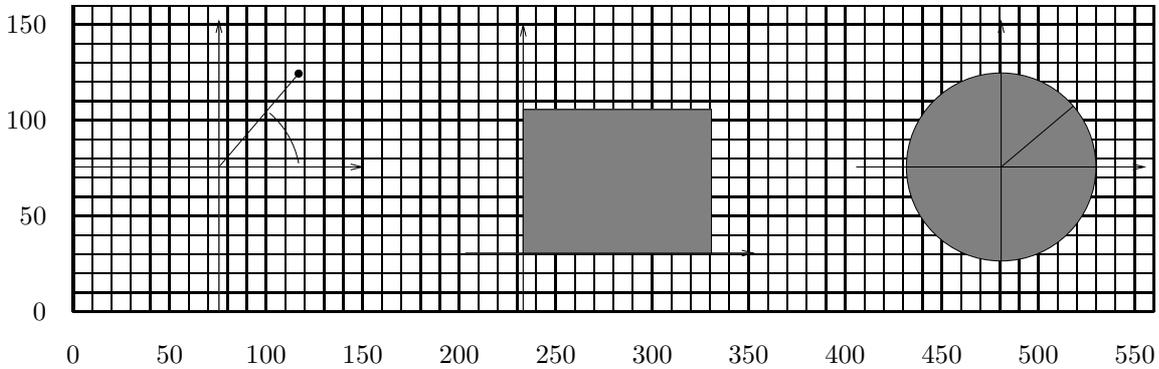


Figure 10.1: (a) Coordinates (x, y) , and (r, θ) , (b) the region \mathbb{U}_ϵ^2 , and (c) the region \mathbb{U}_\circ^2 .

10.1.2 Spatial Domains

As in 1D, a 2D problem is defined over a given domain \mathcal{D} , a subset of the plane \mathbb{R}^2 —see Figure ?? for an illustration of some of the regions to be discussed here. For analysis purposes, it is often convenient to work over the entire plane, or with $\mathcal{D} = \mathbb{R}^2$. In Cartesian coordinates, sometimes, for the analysis of boundary conditions, it is useful to work with a semi-infinite domain of the form $\mathbb{R}^{2,x+} = \{(x, y) \in \mathbb{R}^2, x \geq 0\}$, or $\mathbb{R}^{2,y+} = \{(x, y) \in \mathbb{R}^2, y \geq 0\}$, and in order to deal with corner conditions, the quarter plane $\mathbb{R}^{2,x+y+} = \{(x, y) \in \mathbb{R}^2, x \geq 0, y \geq 0\}$.

In practice, however, at least in Cartesian coordinates, it is the $L_x \times L_y$ rectangular region, of the form $\{(x, y) \in \mathbb{R}^2, 0 \leq x \leq L_x, 0 \leq y \leq L_y\}$ which are of most interest. Through nondimensionalization of spatial variables, i.e., setting coordinates

$$x' = x/L_x \quad y' = y/L_y \tag{10.2}$$

this region may always be reduced to the unit area rectangle of dimensions $\epsilon \times 1/\epsilon$, where $\epsilon^2 = L_x/L_y$ is the aspect ratio for the rectangle. This region will henceforth be called \mathbb{U}_ϵ^2 , and nondimensionalized coordinates will always be assumed (with the primed notation dropped).

In radial coordinates, the main region of interest is the circle of radius R , i.e., $\{(r, \theta) \in \mathbb{R}^2, 0 \leq r \leq R\}$. Again, through the introduction of a nondimensionalized coordinate $r' = r/R$, this region may be reduced to the unit circle, \mathbb{U}_\circ^2 .

10.1.3 Partial Differential Operators

In Cartesian coordinates, the differential operators which appear, beyond partial time derivatives, which have already been discussed in Chapter 5, are of the form $\frac{\partial}{\partial x}$, $\frac{\partial}{\partial y}$, $\frac{\partial^2}{\partial x^2}$, $\frac{\partial^2}{\partial y^2}$, $\frac{\partial^2}{\partial x \partial y}$, etc. When applied to a function $u(x, y, t)$, the following notation will be used:

$$\frac{\partial u}{\partial x} = u_x, \quad \frac{\partial u}{\partial y} = u_y, \quad \frac{\partial^2 u}{\partial x^2} = u_{xx}, \quad \frac{\partial^2 u}{\partial y^2} = u_{yy}, \quad \frac{\partial^2 u}{\partial x \partial y} = u_{xy}, \quad \text{etc.} \quad (10.3)$$

Technical considerations having to do with interchanging the order of derivatives, while important, will be neglected here, so it may be assumed, e.g., that $u_{xy} = u_{yx}$.

In radial coordinates, similar operators and accompanying notation are used, i.e.,

$$\frac{\partial u}{\partial r} = u_r, \quad \frac{\partial u}{\partial \theta} = u_\theta, \quad \frac{\partial^2 u}{\partial r^2} = u_{rr}, \quad \frac{\partial^2 u}{\partial \theta^2} = u_{\theta\theta}, \quad \frac{\partial^2 u}{\partial r \partial \theta} = u_{r\theta}, \quad \text{etc.} \quad (10.4)$$

For isotropic systems in musical acoustics, the most commonly occurring differential operator is not any of the above operators in isolation, but rather the 2D Laplacian Δ , defined in terms of its action on a function u as

$$\Delta u = u_{xx} + u_{yy} \quad \Delta u = \frac{1}{r} (ru_r)_r + \frac{1}{r^2} u_{\theta\theta} \quad (10.5)$$

in Cartesian and radial coordinates, respectively. Also important, especially in problems in plate dynamics, is the fourth order operator known as bi-Laplacian, or biharmonic operator Δ^2 , defined simply as

$$\Delta^2 = \Delta \Delta \quad (10.6)$$

10.1.4 Differential Operators in the Spatial Frequency Domain

Just as in 1D, it is possible to view differential operators in terms of their behaviour in the spatial frequency domain—in general, this is only simple on a Cartesian grid, where differential operators remain shift-invariant. (Notice that operators such as the Laplacian show an explicit dependence on the coordinate r .)

Consider the frequency domain ansatz, a test function of the form

$$u(x, y, t) = e^{st + j\beta_x x + j\beta_y y} \quad (10.7)$$

When $s = j\omega$, this corresponds to a traveling wave, traveling in direction (β_x, β_y) in the Cartesian plane, of wavelength $2\pi/|\boldsymbol{\beta}|$, where $|\boldsymbol{\beta}| = \sqrt{\beta_x^2 + \beta_y^2}$ is the wavenumber magnitude. For such a test function, the various differential operators above act as

$$\frac{\partial}{\partial x} \implies j\beta_x \quad \frac{\partial}{\partial y} \implies j\beta_y \quad (10.8)$$

$$\frac{\partial^2}{\partial x^2} \implies -\beta_x^2 \quad \frac{\partial^2}{\partial y^2} \implies -\beta_y^2 \quad \frac{\partial^2}{\partial x \partial y} \implies -\beta_x \beta_y \quad (10.9)$$

$$\Delta \implies -(\beta_x^2 + \beta_y^2) = -|\boldsymbol{\beta}|^2 \quad \Delta \Delta \implies |\boldsymbol{\beta}|^4 \quad (10.10)$$

Notice in particular that the effect of the operators Δ and $\Delta \Delta$ is dependent only on the wavenumber magnitude $|\boldsymbol{\beta}|$, and not on the individual components β_x and β_y —this is a reflection of the isotropic nature of such operations, which occur naturally in problems which do not exhibit any directional dependence. The same is not true, however, of the discrete operators which approximate them; see §10.2.1.

10.1.5 Inner Products

The definition of the L_2 inner product in 2D is a natural extension of that in 1D, For two functions f and g , dependent on (at least) two spatial coordinates, one may write

$$\langle f, g \rangle_{\mathcal{D}} = \iint_{\mathcal{D}} f g dx dy \qquad \langle f, g \rangle_{\mathcal{D}} = \iint_{\mathcal{D}} f g r dr d\theta \tag{10.11}$$

in Cartesian and radial coordinates, respectively; the same notation will be used for both coordinate systems, though one should note the presence of the factor r implicit in the definition in the radial case. The norm of a function f may be defined, as in the 1D case, as $\|f\|_{\mathcal{D}} = \langle f, f \rangle_{\mathcal{D}}$.

Two important identities also hold for inner products involving the Laplacian:

$$\langle f, \Delta g \rangle \tag{10.12}$$

10.2 Grid Functions and Difference Operators: Cartesian Coordinates

The extension of the definitions in §5.2.2 to two spatial coordinates is, in the Cartesian case, immediate. A grid function $u_{l,m}^n$, for $(l, m) \in \mathcal{D}$, and $n \geq 0$, represents an approximation to a continuous function $u(x, y, t)$, at coordinates $x = lh_x$, $y = lh_y$, $t = nk$. Here, \mathcal{D} is a subset of the set of pairs of integers, \mathbb{Z}^2 . The semi infinite domains corresponding to $\mathbb{R}^{2,x+}$ and $\mathbb{R}^{2,y+}$ are $\mathbb{Z}^{2,x+} = \{(l, m) \in \mathbb{Z}^2, l \geq 0\}$ and $\mathbb{Z}^{2,y+} = \{(l, m) \in \mathbb{Z}^2, m \geq 0\}$. For the quarter plane, one can define $\mathbb{Z}^{2,x+,y+} = \{(l, m) \in \mathbb{Z}^2, l, m \geq 0\}$. Most important, in real-world simulation, is the rectangular region $\mathbb{U}_{N_x, N_y} = \{(l, m) \in \mathbb{Z}^2, 0 \leq l \leq N_x, 0 \leq m \leq N_y\}$.

The temporal operators behave exactly as those defined in 1D, in §5.2.1, and it is not worth repeating these definitions here. Spatial shift operators, in the x and y directions may be defined as

$$e_{x+} u_{l,m}^n = u_{l+1,m}^n \qquad e_{x-} u_{l,m}^n = u_{l-1,m}^n \qquad e_{y+} u_{l,m}^n = u_{l,m+1}^n \qquad e_{y-} u_{l,m}^n = u_{l,m-1}^n$$

and forward, backward and centered difference operators as

$$\delta_{x+} \triangleq \frac{1}{h_x} (e_{x+} - 1) \approx \frac{\partial}{\partial x} \qquad \delta_{x-} \triangleq \frac{1}{h_x} (1 - e_{x-}) \approx \frac{\partial}{\partial x} \qquad \delta_x \triangleq \frac{1}{2h_x} (e_{x+} - e_{x-}) \approx \frac{\partial}{\partial x} \tag{10.13}$$

$$\delta_{y+} \triangleq \frac{1}{h_y} (e_{y+} - 1) \approx \frac{\partial}{\partial y} \qquad \delta_{y-} \triangleq \frac{1}{h_y} (1 - e_{y-}) \approx \frac{\partial}{\partial y} \qquad \delta_y \triangleq \frac{1}{2h_y} (e_{y+} - e_{y-}) \approx \frac{\partial}{\partial y} \tag{10.14}$$

Centered second derivative approximations follow immediately as

$$\delta_{xx} = \delta_{x+} \delta_{x-} \approx \frac{\partial^2}{\partial x^2} \tag{10.15}$$

$$\delta_{yy} = \delta_{y+} \delta_{y-} \approx \frac{\partial^2}{\partial y^2} \tag{10.16}$$

Equal Grid Spacings

Though, in general, one can choose unequal grid spacings h_x and h_y in the two Cartesian coordinates x and y , for simplicity of analysis and programming, it is easier set them equal to a single constant, i.e., $h_x = h_y = h$. This is natural for problems which are isotropic (i.e., for which wave propagation is independent of direction). This simplification is made more much of the remainder of this book. There are cases, though, for which such a choice can lead to errors which can become perceptually important when a very coarse grid is used (i.e., typically in simulating musical systems of high pitch). For an example relating to the 2D wave equation, see §??, and Problem ?? in the following chapter.

The Discrete Laplacian and Biharmonic Operators

Quite important in musical sound synthesis applications is the approximation to the Laplacian operator, as given in (??)—there are clearly many ways of doing this. The simplest, by far, is to make use of what is known as the five-point Laplacian. There are two possible forms of this operator, one making use of points adjacent to the center point, and another employing points diagonally adjacent:

$$\delta_{\Delta\boxplus} = \delta_{xx} + \delta_{yy} = \Delta + O(h^2) \quad \delta_{\Delta\boxtimes} = \delta_{xx} + \delta_{yy} + \frac{h^2}{2}\delta_{xx}\delta_{yy} = \Delta + O(h^2) \quad (10.17)$$

These two operators may be combined in a standard way (see §??) to yield a so-called nine-point Laplacian, depending on a free parameter α :

$$\alpha\delta_{\Delta\boxplus} + (1 - \alpha)\delta_{\Delta\boxtimes} = \Delta + O(h^2) \quad (10.18)$$

Though it involves more grid points, it may be used in order to render the approximation more isotropic—see §11.1.5 for an application of this in the case of the 2D wave equation.

Also important, in the case of the vibrating stiff plate, is the biharmonic operator, or bi-Laplacian, which consists of the composition of the Laplacian with itself, or $\Delta\Delta$. A simple approximation may be given as

$$\delta_{\Delta\boxplus}\delta_{\Delta\boxplus} = \Delta\Delta + O(h^2) \quad (10.19)$$

One could go further here and develop a family of approximations using a parameterized combination of the operators $\delta_{\Delta\boxplus}$ and $\delta_{\Delta\boxtimes}$ —see [31] for more on this topic.

The stencils or footprints of the above spatial difference operators are illustrated in Figure ??.

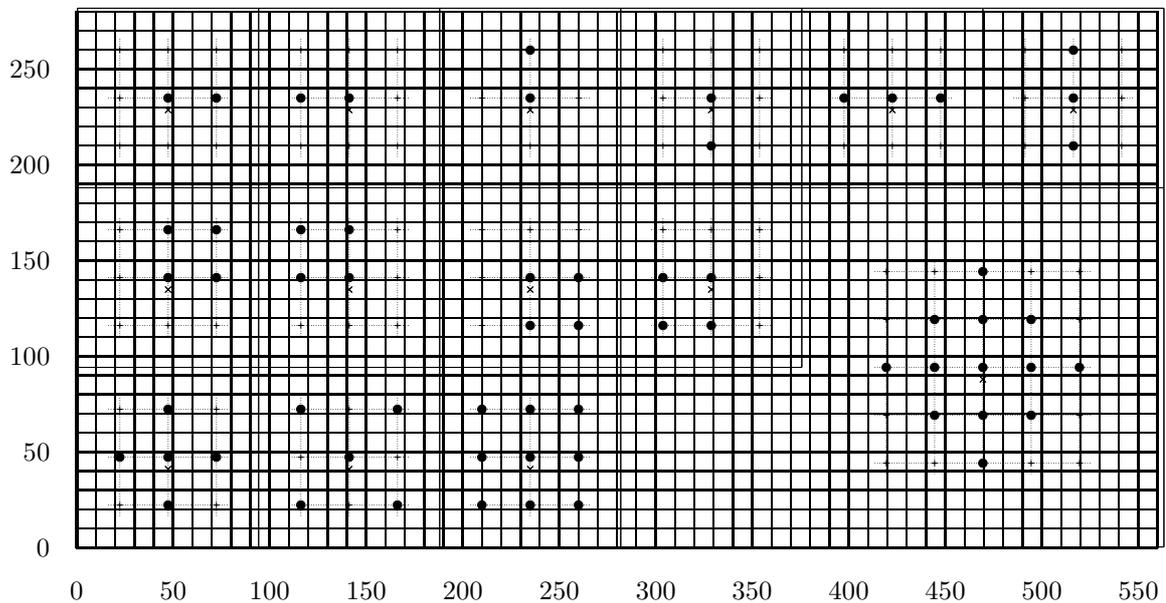


Figure 10.2:

10.2.1 Frequency Domain Analysis

Frequency domain analysis of difference operators in the Cartesian case is a straightforward generalization from 1D. Skipping over the definition of Fourier and z transforms, the ansatz is now

$$u_{l,m}^n = z^n e^{jh(l\beta_x + m\beta_y)} \tag{10.20}$$

The frequency domain behaviour of temporal operators is unchanged from the 1D case. Defining the variables p_x and p_y by

$$p_x = \sin^2(\beta_x h/2) \quad p_y = \sin^2(\beta_y h/2) \tag{10.21}$$

one arrives immediately at

$$\delta_{xx} \implies -\frac{4}{h^2} p_x \quad \delta_{yy} \implies -\frac{4}{h^2} p_y \tag{10.22}$$

$$\delta_{\Delta\oplus} \implies -\frac{4}{h^2} (p_x + p_y) \quad \delta_{\Delta\boxtimes} \implies -\frac{4}{h^2} (p_x + p_y - 2p_x p_y) \quad \delta_{\Delta\Delta} \implies \frac{16}{h^4} (p_x + p_y)^2 \tag{10.23}$$

Anisotropic Behaviour

One new facet of finite difference schemes in 2D is numerical anisotropy—waves travel at different speeds in different directions. This is wholly due to the directional asymmetry imposed on a problem by introducing a grid. This is a phenomenon which shows itself most prominently at high frequencies (or short wavelengths)—in the long-wavelength limit, numerical behaviour becomes approximately isotropic.

As a simple example of this, consider the operator $\delta_{\Delta\oplus}$, as defined in (10.17). It is perhaps easiest to examine the anisotropy in the frequency domain representation. Expanding from (10.23) in powers of the wavenumber components β_x and β_y , gives

$$\delta_{\Delta\oplus} \implies -\frac{4}{h^2} (p_x + p_y) = -(\beta_x^2 + \beta_y^2) + \frac{h^2}{12} (\beta_x^2 + \beta_y^2) + O(h^4) \tag{10.24}$$

Thus, as expected, the operator $\delta_{\Delta\oplus}$ approximates the Laplacian Δ to second order accuracy in the grid spacing h , but the higher order terms can not be grouped in terms of the wavenumber magnitude alone. Such numerical anisotropy, and ways of reducing it, will be discussed with regard to the 2D wave equation in §11.1.5. See also Problem 10.2.

Amplification Polynomials

Just as in the lumped and 1D cases, for LSI problems in 2D, in the analysis of difference schemes, one often arrives at amplification polynomials of the following form:

$$P(z) = \sum_{l=0}^M a_l(\beta_x, \beta_y) z^l = 0 \tag{10.25}$$

and as before, a stability condition is arrived at by finding conditions such that the roots are bounded by 1 in magnitude, now for all values of β_x and β_y supported on the grid.

10.2.2 An Inner Product and Energetic Manipulations

10.3 Grid Functions and Difference Operators in Two Dimensions: Radial Coordinates

Problems defined over a circular geometry play a large role in musical acoustics, in particular when it comes to the modeling of percussion instruments such as drums and gongs. Though finite element

models are often used under such conditions, a finite difference approach is still a wise choice, and can lead to very efficient, and easily programmed sound synthesis algorithm designs.

10.4 Problems

Problem 10.1 Given a general bilinear function $f(p, q)$ in the variables p and q

$$f(p, q) = a + bp + cq + dpq \quad (10.26)$$

defined over the square region $0 \leq p, q \leq 1$, for constants a, b, c and d , show that f takes on its maximum and minimum values at the corners of the domain.

Problem 10.2 Recalling the analysis of the anisotropy of the operator $\delta_{\Delta\boxplus}$ in §10.2.1, consider the nine-point Laplacian operator as given in (10.18), which depends on the free parameter α , and show that its expansion, in terms of wavenumber components β_x and β_y is

$$\alpha\delta_{\Delta\boxplus} + (1 - \alpha)\delta_{\Delta\boxtimes} \implies -(\beta_x^2 + \beta_y^2) + \frac{h^2}{12}(\beta_x^4 + 6(1 - \alpha)\beta_x^2\beta_y^2 + \beta_y^4) + O(h^4) \quad (10.27)$$

For an arbitrary choice of α , can the $O(h^2)$ term be written in terms of the wavenumber magnitude $|\beta|$ alone? If not, is there a value (or values) of α such that it can be? If you can find such a value of alpha, then the anisotropic behaviour of the parameterized operator will exhibit itself only at fourth order (though the operator remains a second order accurate approximation to the Laplacian).

10.5 Programming Exercises

Chapter 11

The 2D Wave Equation

The systems examined so far in this book have been one-dimensional. Certain musical instruments, in particular those of the percussion family, employ vibrating structures which can only be well-described in two dimensions. As one might expect, simulation complexity increases accordingly. At the time of writing, there has been, so far, relatively little work on two-dimensional problems in sound synthesis, partly because, until recently, real time synthesis from such systems on small computers was not possible. Another reason has been that percussion instruments have seen much less fundamental investigation from the point of view of musical acoustics than other instruments (though there is a growing body of work by Rossing [1] and others [2]), often concerned with experimental determination of modal frequencies, but not always [3]). On the other hand, such problems have a long research history in mainstream simulation, and, as a result, there is a wide expanse of literature and results which may be adapted to sound synthesis applications. For the linear problems discussed in this chapter, the systems may be directly generalized from their 1D counterparts, described in Chapter 7, and for this reason, the development here is somewhat abbreviated.

A good starting point is the 2D wave equation, introduced in §11.1, which serves as a useful test problem for the vibration of membranes, and also as another good point of comparison for the various physical modeling synthesis techniques, including finite difference schemes, modal methods, lumped networks, and digital waveguide meshes. The previous chapter serves as a reference for the techniques to be discussed here and in the following two chapters.

References include: [24, 32, 23, 290, 259, 31, 273, 225, 272, 98, 99, 224, 227, 107, 106, 30, 108, 150]

11.1 The 2D Wave Equation

The wave equation in one spatial dimension (6.1) may be directly generalized to two dimensions as

$$u_{tt} = c^2 \Delta u \quad (11.1)$$

Here, u is the dependent variable of interest, c is a wave speed, and Δ is the Laplacian operator as discussed in Chapter 10. When spatially non-dimensionalized using a characteristic length L , the 2D wave equation is of the form

$$u_{tt} = \gamma^2 \Delta u \quad (11.2)$$

The 2D wave equation, as given above, is a simple approximation to the behaviour of a vibrating membrane, and also can serve as a preliminary step towards the treatment of room acoustics problems (where, in general, a 3D wave equation would be employed). As previously, it will be assumed, for the moment, that the 2D wave equation is defined over \mathbb{R}^2 , so that a discussion of boundary conditions

may be postponed.

The wave equation, like all second-order in time PDEs, must be initialized with two functions (see §6.1.3); in Cartesian coordinates, for example, one would set, normally, $u(x, y, 0) = u_0(x, y)$ and $\frac{\partial u}{\partial t}(x, y, 0) = v_0(x, y)$. In the case of a membrane, the first condition corresponds to a pluck, and the second to a strike. A useful all-purpose initializing distribution is a 2D raised cosine, of the form

$$c_{rc}(x, y) = \begin{cases} \frac{c_0}{2} \left(1 + \cos(\pi \sqrt{(x-x_0)^2 + (y-y_0)^2} / x_{hw}) \right), & \sqrt{(x-x_0)^2 + (y-y_0)^2} \leq x_{hw} \\ 0, & (x-x_0)^2 + (y-y_0)^2 > x_{hw} \end{cases} \quad (11.3)$$

which has amplitude c_0 , and is centered at coordinates (x_0, y_0) , of half-width x_{hw} . Such a distribution can be used in order to model both plucks and strikes.

A pair of simulation results, under plucked and struck conditions, are illustrated in Figure 11.1. The main feature which distinguishes the behaviour of solutions to the 2D wave equation from the 1D case is the reduction in amplitude of the wave as it evolves, due to spreading effects—related to this is the presence of a “wake” behind the wavefront, which is absent in the 1D case, as well as the fact that, even when simple boundary conditions are employed, the solution is not periodic—see §11.1.3. One might guess, from these observations alone, that the efficiency of the 1D digital waveguide formulation for the 1D case, built around waves which travel without distortion, will disappear in this case—this is in fact true, as will be discussed in §11.1.6.

Plucked

Struck

$t = 0.0$ $t = 0.0005$ $t = 0.001$ $t = 0.0015$

Figure 11.1: *Time evolution of solutions to the 2D wave equation, defined over a square, with fixed boundary conditions. In this case, $\gamma = 400$, snapshots of the displacement are shown at times as indicated, under plucked conditions (top row) and struck conditions (bottom row). In the first case, the initial displacement condition is a raised cosine of maximum amplitude 0.001, and of half-width 0.15, centered at the center of the domain, and the second, the initial velocity profile is a raised cosine, of maximum amplitude 10, half-width 0.15, and also centered at the domain center.*

11.1.1 Phase and Group Velocity

Assuming, in Cartesian coordinates, a plane wave solution of the form

$$u(x, y, t) = e^{st+j(\beta_x x + \beta_y y)} \quad (11.4)$$

leads to the characteristic equation and dispersion relation

$$s^2 = -\gamma^2|\beta|^2 \quad \Rightarrow \quad \omega = \pm\gamma|\beta| \quad (11.5)$$

and, thus, to expressions for the phase and group velocity,

$$v_\phi = \gamma \quad v_g = \gamma \quad (11.6)$$

Thus, just as in 1D, all wave-like components travel at the same speed. Note that as the Laplacian operator is isotropic, wave speed is independent not only of frequency or wavenumber, but also of direction.

In 1D, for LSI systems, it is possible to rewrite the phase and group velocities as functions of frequency alone. While this continues to be possible for isotropic systems in 2D, it is not possible when the system exhibits anisotropy—most numerical methods exhibit such spurious anisotropy, and thus the only way to evaluate dispersive behaviour is through expressions of wavenumber, such as those given above.

11.1.2 Energy and Boundary Conditions

The 2D wave equation, like its 1D counterpart, is lossless. This can be seen, as usual, by writing the energy for the system, which is conserved. Assuming (11.1) to be defined over the infinite plane $\mathcal{D} = \mathbb{R}^2$, an inner product with u_t in Cartesian coordinates and using integration by parts leads to

$$\langle u_t, u_{tt} \rangle_{\mathbb{R}^2} = \gamma^2 \langle u_t, \Delta u \rangle_{\mathbb{R}^2} = \gamma^2 \langle u_t, u_{xx} \rangle_{\mathbb{R}^2} + \gamma^2 \langle u_t, u_{yy} \rangle_{\mathbb{R}^2} = -\gamma^2 \langle u_{xt}, u_x \rangle_{\mathbb{R}^2} - \gamma^2 \langle u_{yt}, u_y \rangle_{\mathbb{R}^2} \quad (11.7)$$

or,

$$\frac{d\mathfrak{H}}{dt} = 0 \quad \text{with} \quad \mathfrak{H} = \mathfrak{T} + \mathfrak{V} \quad (11.8)$$

and

$$\mathfrak{T} = \frac{1}{2} \|u_t\|_{\mathbb{R}^2}^2 \quad \mathfrak{V} = \frac{1}{2} (\|u_x\|_{\mathbb{R}^2}^2 + \|u_y\|_{\mathbb{R}^2}^2) = \frac{1}{2} \|\nabla u\|_{\mathbb{R}^2}^2 \quad (11.9)$$

where ∇ signifies the gradient operation. Such a result is obviously independent of the chosen coordinate system—see Problem ???. The expression is, again, non-negative, and leads immediately to bounds on the growth of the norm of the solution, by exactly the same methods as discussed in §???

Edges

In order to examine boundary conditions at, e.g., a straight edge, consider (11.1) defined over the semi-infinite region $\mathcal{D} = \mathbb{R}^{2,x+}$, as discussed in §10.1.2. Through the same manipulations above, one arrives at the energy balance

$$\frac{d\mathfrak{H}}{dt} = \mathfrak{B} \quad \text{with} \quad \mathfrak{H} = \mathfrak{T} + \mathfrak{V} \quad (11.10)$$

where \mathfrak{T} and \mathfrak{V} are as defined in (11.9) above, over the region $\mathfrak{D} = \mathbb{R}^{2,x+}$. The boundary term \mathfrak{B} , depending only on values of the solution over the boundary at $x = 0$, is

Corners

11.1.3 Modes

The modal decomposition of solutions to the 2D wave equation under certain simple geometries and boundary conditions is heavily covered in the musical acoustics literature—see, e.g., [96]. It is useful, however, to at least briefly review such a decomposition in the case of the rectangular membrane.

As in the case of the 1D wave equation (see §??) and the ideal bar equation (see §??), one may assume an oscillatory solution to the 2D wave equation of the form $u(x, y, t) = e^{j\omega t}U(x, y)$, leading to

$$-\omega^2 U = \gamma^2 \Delta U \quad (11.11)$$

The wave equation defined over a unit area rectangle, of aspect ratio α , and under fixed boundary conditions is separable, and the solutions to the above equation are

$$U_{p,q}(x, y) = \sin(p\pi\sqrt{\alpha}x) \sin(q\pi y/\alpha) \quad \omega_{p,q} = \pi\gamma\sqrt{\alpha p^2 + q^2/\alpha} \quad (11.12)$$

The modal functions are illustrated in Figure 11.2, in the case of a square domain.

$q = 1$

$q = 2$

$q = 3$

$p = 1$ $p = 2$ $p = 3$ $p = 4$ $p = 5$

Figure 11.2: Modal shapes $U_{p,q}(x, y)$ for a square membrane, under fixed boundary conditions.

Given the above expression for the modal frequencies $f_{p,q}$, it is not difficult to show that the number of modes of frequency less than or equal to a given frequency $f_s/2$ will be

$$N_m(f_s/2) = \frac{\pi f_s^2}{4\gamma^2} \quad (11.13)$$

See Problem 11.3. Twice this number is, as before, the number of degrees of freedom of the system when it is simulated at a sample rate f_s .

11.1.4 A Simple Finite Difference Scheme

Consider the wave equation in Cartesian coordinates. The simplest possible finite difference scheme employs a second difference in time, and a five point Laplacian approximation:

$$\delta_{tt}u = \gamma^2 \delta_{\Delta,+}u \quad (11.14)$$

where $u = u_{l,m}^n$ is a 2D grid function representing an approximation to the continuous solution $u(x, y, t)$ at $x = lh$, $y = mh$, $t = nk$, for integer l , m and n . Recall that h is the grid spacing in both the x and y directions, and as always, k is the time step.

The case $h_x = h_y$ is of most interest: expanding out the operator notation above leads to

$$u_{l,m}^{n+1} = \lambda^2 (u_{l+1,m}^n + u_{l-1,m}^n + u_{l,m+1}^n + u_{l,m-1}^n) + 2(1 - 2\lambda^2) u_{l,m}^n - u_{l,m}^{n-1} \quad (11.15)$$

where again, the Courant number λ is defined as

$$\lambda = \frac{k\gamma}{h} \quad (11.16)$$

Notice that under the special choice of $\lambda = 1/2$ (this case is of particular relevance to the so-called digital waveguide mesh, to be discussed in §11.1.6), the recursion above may be simplified to

$$u_{l,m}^{n+1} = \frac{1}{2} (u_{l+1,m}^n + u_{l-1,m}^n + u_{l,m+1}^n + u_{l,m-1}^n) - u_{l,m}^{n-1} \quad (11.17)$$

which involves multiplications by the factor $1/2$ only, and one fewer addition. It also has the interesting property that the scheme may be decomposed into two independent schemes operating over the black and white squares on a "checkerboard" grid. See Problem 11.7.

Scheme (11.14) continues to hold if the grid spacings are not the same, though the stability condition (discussed below) must be altered, and computational complexity increases. See Problem 11.5.

von Neumann analysis

Use of the ansatz $u_{l,m}^n = z^n e^{jh(l\beta_x + m\beta_y)}$ leads to the characteristic equation

$$z - 2 + 4\lambda^2 (p_x + p_y) + z^{-1} = 0 \quad (11.18)$$

in the two variables $p_x = \sin^2(\beta_x h/2)$ and $p_y = \sin^2(\beta_y h/2)$, as defined in §??. This equation is again of the form (??), and the solutions in z will be of unit modulus when

$$0 \leq \lambda^2 (p_x + p_y) \leq 1 \quad (11.19)$$

The left-hand inequality is clearly satisfied. Given that the variables p_x and p_y take on values between 0 and 1, the right hand inequality is satisfied for

$$\lambda \leq \frac{1}{\sqrt{2}} \quad (11.20)$$

which is the stability condition for scheme (11.14). Notice that just as in the case of scheme (??) for the 1D wave equation, at the stability limit (i.e., when $\lambda = 1/\sqrt{2}$), a simplified scheme results (namely (11.17)).

11.1.5 Other Finite Difference Schemes

In the case of the 1D wave equation, certain parameterized finite difference schemes were examined in §6.3, but given the good behaviour of the simplest explicit scheme, these variations were of virtually no use—the power of such parameterized methods was seen to some extent in the case of bar and stiff string simulation, in §?? and §??, and is even more useful in the present case of multidimensional systems such as the 2D wave equation (among others). The chief interest is in reducing numerical dispersion. The 2D wave equation, like its 1D counterpart, is a standard numerical test problem, and, as such, has seen extensive investigation—see, e.g., [262, 32, 23].

A Parameterized Explicit Scheme

Instead of a five-point Laplacian approximation, as employed in scheme (11.14), one might try a nine-point approximation, of the type discussed in §??. This leads to

$$\delta_{tt}u = \gamma^2 (\alpha\delta_{\Delta,+}u + (1 - \alpha)\delta_{\Delta,\times})u \quad (11.21)$$

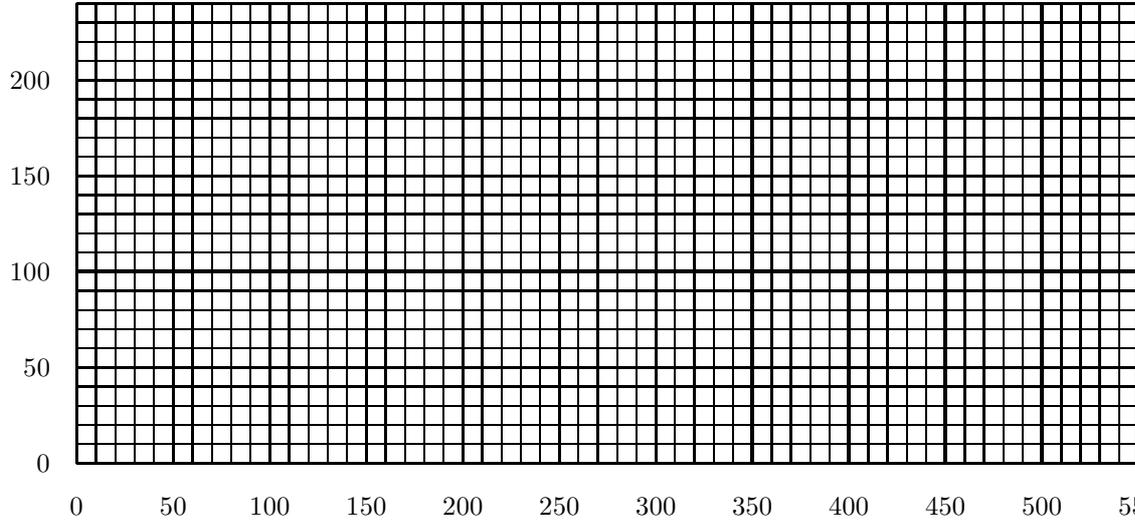


Figure 11.3: Computational grid (top) and relative numerical phase velocity (bottom) for (a) The explicit scheme (??), with $\lambda = 1/2$, (b) the parameterized scheme (??) with the special choice of $\alpha = 2/\pi$, and λ chosen to satisfy bound (??) with equality, and (c) the implicit scheme (??), with $\alpha = 2/\pi$, $\theta = 1.2$, and λ chosen to satisfy bound (??) with equality. Relative phase velocity deviations of 1% are plotted as contours in the wavenumber plane $0 \leq \beta_x, \beta_y \leq \frac{\pi}{h}$.

which reduces to scheme (11.14) when $\alpha = 1$. The computational stencil is, obviously, more dense in this case than for the simple scheme (??)—see Figure 11.3(b).

Using von Neumann analysis, it is straightforward to show that the stability conditions for this scheme are

$$\alpha \geq 0 \quad \lambda \leq \min\left(1, \frac{1}{\sqrt{2\alpha}}\right) \tag{11.22}$$

See Problem 11.2.

The dispersion characteristics of this scheme depend quite heavily on the choice of α , and, as one might expect, for intermediate values of this parameter over the allowable range $\alpha \in [0, 1]$, the dispersion is far closer to isotropic—a desirable characteristic, given that the wave equation itself exhibits such behaviour. Through trial and error, or perhaps optimization strategies, one may find particularly good behaviour when $\alpha \approx 0.64$, as illustrated in the plot of relative phase velocity plotted in Figure 11.3(b).

Such a choice may be justified, again, by looking at computational complexity (see §?? for similar analysis in the case of the ideal bar). For a surface of unit area, and when the stability bound above is satisfied with equality, the number of grid points necessary will be

$$N_{fd} = \frac{1}{h^2} = \frac{f_s^2}{\gamma^2 \min(1, 1/2\alpha)} \tag{11.23}$$

and equating this with the number of modes, one arrives at a choice of $\alpha = 2/\pi \approx 0.637$.

A Compact Implicit Scheme

A two-parameter family of implicit schemes for the 2D wave equation is given by

$$\delta_{tt}u = \gamma^2 (\theta + (1 - \theta)\mu_t) (\alpha\delta_{\Delta,+}u + (1 - \alpha)\delta_{\Delta,\times}) u \tag{11.24}$$

Table 11.1: Comparison among modal frequencies of the 2D wave equation, defined over a square, under fixed conditions, with $\gamma = 1000$, and modal frequencies (as well as their cent deviations from the exact frequencies) of the simple explicit scheme (??), the parameterized explicit scheme (11.21) with $\alpha = 2/\pi$, and the implicit scheme (11.24), with $\alpha = 2/\pi$ and $\theta = 1.2$, with a sample rate $f_s = 16000$ Hz, and where λ is chosen so as to satisfy the stability condition in each case as close to equality as possible.

Mode number	Exact	Explicit		Nine-pt. Explicit		Implicit	
	Freq.	Freq.	Cent Dev.	Freq.	Cent Dev.	Freq.	Cent Dev.
(1,1)	707.1	707.0	-0.3	706.3	-2.0	707.1	-0.1
(1,2)	1118.0	1114.0	-6.2	1114.9	-4.8	1118.0	0.0
(2,2)	1414.2	1413.1	-1.3	1407.6	-8.1	1413.4	-1.0
(1,4)	2061.6	2009.2	-44.5	2042.7	-15.9	2058.8	-2.2
(3,3)	2121.3	2117.5	-3.1	2098.7	-18.5	2115.7	-4.6

which now must be updated using linear system solution techniques.

von Neumann analysis, though now somewhat more involved, again allows fairly simple stability conditions on the free parameters and λ :

$$\alpha \geq 0 \quad \theta \text{ unconstrained} \quad \begin{cases} \lambda \leq \sqrt{\frac{\min(1, 1/(2\alpha))}{\sqrt{2(2\theta-1)}}}, & \theta > \frac{1}{2} \\ \lambda \text{ unconstrained} & \theta \leq \frac{1}{2} \end{cases} \quad (11.25)$$

See Problem 11.4. Under judicious choices of α and θ , an excellent match to the ideal phase velocity may be obtained over nearly the entire range of wavenumbers—see Figure 11.3(c). Such is the interest in compact implicit schemes—very good overall behaviour, at the additional cost of sparse linear system solutions, without unpleasant side-effects such as complex boundary termination.

Further Varieties

Beyond the two schemes presented here, there are of course many varieties of schemes for the 2D wave equation—further properties of parameterized schemes for the 2D wave equation are detailed elsewhere [23], and other schemes are discussed in standard texts [244].

11.1.6 Digital Waveguide Meshes

The extension of digital waveguides to multiple dimensions for sound synthesis applications was first undertaken by van Duyne and Smith in the mid 1990s [272, 273, 274]. It has continued to see a fair amount of activity in sound synthesis and artificial reverberation applications—see the comments and references in §1.2.3. Most interesting was the realization by van Duyne and others of the association with finite difference schemes.

A regular Cartesian mesh is shown in Figure 11.4(a). Here, each box labelled **S** represents a four-port parallel scattering junction. A scattering junction at location $x = lh, y = mh$ is connected to its four neighbors on the grid by four bidirectional delay lines, or waveguides, each of a single sample delay (of k seconds, where $f_s = 1/k$ is the sample rate). The signals impinging on a given scattering junction at grid location l, m at time step n from a waveguide from the north, south, east and west are written as $u_{l,m}^{n,(+),N}$, $u_{l,m}^{n,(+),S}$, $u_{l,m}^{n,(+),E}$, and $u_{l,m}^{n,(+),W}$, respectively, and those exiting

as $u_{l,m}^{n,(-),N}$, $u_{l,m}^{n,(-),S}$, $u_{l,m}^{n,(-),E}$ and $u_{l,m}^{n,(-),W}$. The scattering operation at a given junction may be written as

$$u_{l,m}^n = \frac{1}{2} \left(u_{l,m}^{n,(+),N} + u_{l,m}^{n,(+),S} + u_{l,m}^{n,(+),E} + u_{l,m}^{n,(+),W} \right) \tag{11.26}$$

$$u_{l,m}^{n,(-),\bullet} = -u_{l,m}^{n,(+),\bullet} + u_{l,m}^n \tag{11.27}$$

$$\tag{11.28}$$

Here, the $u_{l,m}^n$ is the junction variable (often referred to as a junction pressure). Shifting in digital waveguides themselves can be written as

$$u_{l,m}^{n,(+),N} = u_{l,m+1}^{n-1,(-),S} \quad u_{l,m}^{n,(+),S} = u_{l,m-1}^{n-1,(-),N} \quad u_{l,m}^{n,(+),E} = u_{l+1,m}^{n-1,(-),W} \quad u_{l,m}^{n,(+),W} = u_{l-1,m}^{n-1,(-),E} \tag{11.29}$$

The scattering and shifting operations are the basis of all wave-based numerical methods, including, in addition to digital waveguides, wave digital filter methods, as well as the transmission line matrix method (which are identical, in many respects, to waveguide networks). Numerical stability for an algorithm such as the above is obvious: the shifting operations clearly cannot increase any solution norm, and the scattering operation corresponds to an orthogonal (i.e., l_2 norm preserving) matrix multiplication.

On the other hand, by applying the scattering and shifting rules above, one may arrive at a recursion in the junction variables $u_{l,m}^n$, which is none other than scheme (??), with $\lambda = \gamma k/h = 1/\sqrt{2}$ —see Problem 11.1. Notice, however, that the finite difference scheme requires two units of memory per grid point, whereas in the wave implementation, five are necessary (i.e., the waves impinging on a junction at a given time step, as well as the junction variable itself). Furthermore, the mesh requires seven arithmetic operations (three in order to form the junction variable, and four in order to perform scattering), whereas the finite difference scheme requires four. Thus the efficiency advantage of the waveguide in 1D, based around the use of delay lines, does not carry over to the multidimensional case. Still, the structural robustness of the waveguide mesh is a very desirable property—indeed, termination of a mesh structure using passive filtering blocks

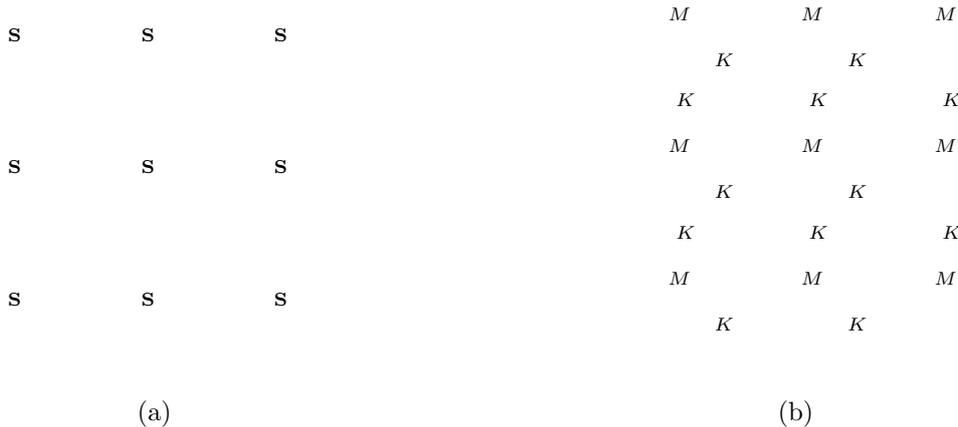


Figure 11.4: (a) A digital waveguide network corresponding to scheme (??) for the wave equation, under the special choice of $\lambda = 1/\sqrt{2}$, and (b), a lumped network corresponding to scheme (??), for any value of λ .

11.1.7 Lumped Networks

A regular lumped network of masses and springs also, not surprisingly, may be viewed in terms of the same underlying finite difference scheme. Suppose that, as shown in Figure 11.4(b), the network consists of a regularly spaced array of masses M , located at positions $x = lh, y = mh$, and constrained to move vertically, with displacement indicated by $u_{l,m}(t)$. Each is connected to neighboring masses by springs of stiffness K , and the force exerted on the mass l, m by its neighbors to the north, south, east and west are indicated by $f_{l,m+1/2}, f_{l,m-1/2}, f_{l+1/2,m}$ and $f_{l-1/2,m}$, respectively.

The equation of motion for a single mass will be

$$M \frac{d^2 u_{l,m}}{dt^2} = f_{l,m+1/2} + f_{l,m-1/2} + f_{l+1/2,m} + f_{l-1/2,m} \quad (11.30)$$

with

$$f_{l,m+1/2} \approx K(u_{l,m+1} - u_{l,m}) \quad f_{l+1/2,m} \approx K(u_{l+1,m} - u_{l,m}) \quad (11.31)$$

Just as before, this may be rewritten as

$$M \frac{d^2 u_{l,m}}{dt^2} = K(u_{l,m+1} + u_{l,m-1} + u_{l+1,m} + u_{l-1,m} - 4u_{l,m}) \quad (11.32)$$

Discretizing the above system of ODEs with a second time difference δ_{tt} leads immediately to scheme (??) for the 2D wave equation, with $\gamma = h\sqrt{K/M}$.

11.1.8 Modal Synthesis

11.2 Problems

Problem 11.1 Beginning from the definition of the grid variable $u_{l,m}^{n+1}$ in terms of wave variables, from (11.26), use the scattering operation (11.27) and the shifting operation (11.29) in order to show that the waveguide mesh calculates solutions to the finite difference scheme (??), under the special choice of $\lambda = \gamma k/h = 1/\sqrt{2}$.

Problem 11.2 Show that the characteristic polynomial corresponding to the “nine-point” explicit scheme (11.21) is given by

$$z - 2 + 4\lambda^2(p_x + p_y - 2(1 - \alpha)p_x p_y) + z^{-1} \quad (11.33)$$

where p_x and p_y are as defined in (??), and take on values between 0 and 1. The stability condition is thus

$$0 \leq \lambda^2(p_x + p_y - 2(1 - \alpha)p_x p_y) \leq 1 \quad (11.34)$$

Show that these conditions lead immediately to the stability conditions given in (??).

Hint: Begin with the left inequality, and derive a condition on α alone, then use the right inequality to find the condition on λ . Your results from problem (10.1) will be of great use here.

Problem 11.3 Considering the 2D wave equation defined over a square region of unit area, under fixed boundary conditions, the mode frequency of mode (p, q) is given by (??), or, in Hertz,

$$f_{p,q} = \frac{\gamma}{2} \sqrt{p^2 + q^2} \quad (11.35)$$

for $p, q > 0$.

(a) Show that $f_{p,q}$ corresponds to the distance between the origin and the point of coordinates $(p\gamma/2, q\gamma/2)$ in a plane.

(b) Find the density (per unit area) of such points in the plane.

(c) The total number $N_m(f_s/2)$ of modes of frequency less than or equal to $f_s/2$ may be represented as a quarter circle of radius $f_s/2$ in the plane of modal points. Given your expression for the density of modes above, find an approximate value for $N_m(f_s/2)$.

(d) Show that this expression remains unchanged if the 2D wave equation is defined instead over a arbitrary rectangle of unit area.

Problem 11.4 Show that the characteristic polynomial corresponding to the “nine-point” explicit scheme (11.21) is given by

$$z + \frac{-2 + 4\lambda^2(p_x + p_y - 2(1 - \alpha)p_x p_y)}{1 + 2(1 - \theta)\lambda^2(p_x + p_y - 2(1 - \alpha)p_x p_y)} + z^{-1} = 0 \quad (11.36)$$

where p_x and p_y are again as defined in (??), and take on values between 0 and 1. The stability condition is thus

$$0 \leq \frac{\lambda^2(p_x + p_y - 2(1 - \alpha)p_x p_y)}{1 + 2(1 - \theta)\lambda^2(p_x + p_y - 2(1 - \alpha)p_x p_y)} \leq 1 \quad (11.37)$$

Show that these conditions lead immediately to the stability conditions given in (??).

This is much trickier than the above problem, in that now the function to be bounded is rational rather than polynomial. There are essentially three steps: 1) Find conditions under which the numerator of the expression above is non-negative. 2) Given this condition, find conditions under which the denominator is non-negative. 3) Given the conditions derived in 1) and 2) above, find conditions under which the numerator is less than or equal to the denominator. Your bounds on λ may overlap, depending on the choice of θ and α .

Problem 11.5 Consider scheme (11.14), but where the grid spacings h_x and h_y are distinct.

(a) Show, from von Neumann analysis, that the stability condition relating k , h_x and h_y becomes

$$\gamma^2 k^2 \leq \frac{h_x^2 h_y^2}{h_x^2 + h_y^2} \quad (11.38)$$

(b) Write the update explicitly in terms of the grid function $u_{l,m}^n$, in a manner similar to that shown in (11.17). How many additions/multiplications are necessary in order to perform the update at a given grid point? Is there a particular choice of k , h_x , and h_y such that computational complexity may be reduced?

Problem 11.6 For the parameterized explicit scheme (??) for the 2D wave equation,

(a) Write the update explicitly in terms of the grid function $u_{l,m}^n$, in a manner similar to that shown in (11.17).

(b) For a given value of α , the scheme free parameter, what is the condition on λ such that the scheme update no longer depends on the value of the grid function $u_{l,m}^n$ at the “center point.” Does this condition conflict with the stability condition (??) for the scheme? (This case is important, as it corresponds to the so-called “interpolated digital waveguide mesh” []).

(c) Discuss the significance of a choice of λ away from the stability condition

Problem 11.7 Show that scheme (11.14), operating over the infinite domain $\mathcal{D} = \mathbb{Z}^2$ may be decomposed into two separate schemes, one operating over values of the grid function $u_{l,m}^n$ for even values of $n + l + m$, and one for odd values. If the scheme is restricted to operate over a finite domain, discuss the effects of various boundary conditions on the ability to obtain such a decomposition.

Problem 11.8 Construct a three parameter family of implicit finite difference schemes for the 2D wave equation, generalizing scheme (??).

11.3 Programming Exercises

Chapter 12

Linear Plate Vibration

The 2D wave equation was treated in the previous chapter as a test case for sound synthesis methods. But, aside from the case of the vibrating membrane, it is often materials with an inherent stiffness which are of interest. When these are flat, they are referred to as plates, and when curved, as shells. The physics of vibrating plates is far more complex than that of the membrane, described by the 2D wave equation, and this complexity shows itself in the resulting equations of motion, even in the simplest case of thin linear plate vibration.

Because the 2D wave equation is so simply expressed, it is often assumed that the increased model complexity for systems such as plates must translate to more computational work. But, as seen in Chapter 7, stiffness tends to *reduce* computational costs. The same is true of plates—whereas simulation of membrane vibration in real time is, even now, quite daunting, plate synthesis is not. This is good news, as the world of sounds produced by plates is a very rich one indeed.

The lossless Kirchhoff thin plate model, as well as several variants which serve to describe more realistic plate vibration in a musical context, are discussed in §12.1 and §12.2. As examples, the audio effect known as plate reverberation is simulated in §12.3, and piano soundboards in §12.4.

References include: [24, 32, 23, 290, 259, 31, 273, 225, 272, 98, 99, 224, 227, 107, 106, 30, 108, 150]

12.1 The Kirchhoff Thin Plate Model

The Kirchhoff model of a uniform thin isotropic plate is defined as

$$\rho H u_{tt} = -D \Delta \Delta u \quad (12.1)$$

Here, u is the plate deflection in a transverse direction, ρ is a material density, H is the plate thickness, and the constant D , defined by

$$D = \frac{EH^3}{12(1-\nu^2)} \quad (12.2)$$

is sometimes known as a stiffness parameter for the plate, depending on Young's modulus E , and Poisson's ratio ν . When spatially non-dimensionalized with respect to a length parameter L , the Kirchhoff model may be written as

$$u_{tt} = -\kappa^2 \Delta \Delta u \quad (12.3)$$

where

$$\kappa^2 = \frac{D}{\rho H L^4} \quad (12.4)$$

The model above is but the simplest representation of plate dynamics. It does not take into account effects that occur at high amplitudes, which will be covered in the next chapter, nor does

it hold when the plate becomes thick (i.e., when H/L becomes large in some sense), in which case a thick plate model (some examples of which are very well covered in the text by Graff [111]) is perhaps more appropriate. Most plates which occur in a musical setting, however, are quite thin, and the above model is sufficient for at least a preliminary foray into physical modeling synthesis.

12.1.1 Phase and Group Velocity

The characteristic equation for the thin plate equation (12.3) is

$$s^2 = -\kappa^2|\beta|^2 \quad \Rightarrow \quad \omega = \pm\kappa|\beta|^2 \quad (12.5)$$

The expressions for the phase and group velocity are

$$v_\phi = \kappa|\beta| \quad v_g = 2\kappa|\beta| \quad (12.6)$$

Again, just as in the case of the ideal bar in 1D, wave propagation is dispersive, with short wavelength components traveling faster than those of low frequency.

This system, like the wave equation, is isotropic—again, though it would be possible to write the expressions for phase and group velocity in terms of frequency ω alone, the associated difference schemes lose the isotropic property, and the easiest comparison is through the expressions above, in terms of wavenumber β .

12.1.2 Boundary Conditions

12.1.3 Finite Difference Schemes: Cartesian Coordinates

Probably the most straightforward finite difference method for the Kirchhoff plate employs the five-point approximation to the Laplacian twice:

$$\delta_{tt}u = -\kappa^2\delta_{\Delta,+}\delta_{\Delta,+}u \quad (12.7)$$

When written out in full in terms of the grid function $u_{l,m}^n$, assuming that the grid spacing is equal to h in both the x and y directions, the scheme is of the form

$$u_{l,m}^{n+1} = (2 - 20\mu^2)u_{l,m}^n + 8\mu^2(u_{l,m+1}^n + u_{l,m-1}^n + u_{l+1,m}^n + u_{l-1,m}^n) - 2\mu^2(u_{l+1,m+1}^n + u_{l+1,m-1}^n + u_{l-1,m+1}^n + u_{l-1,m-1}^n) - \mu^2(u_{l,m+2}^n + u_{l,m-2}^n + u_{l+2,m}^n + u_{l-2,m}^n) \quad (12.8)$$

and depends on the scheme parameter μ , defined as

$$\mu = \frac{\kappa k}{h^2} \quad (12.10)$$

As one might gather, there are many possibilities for the generalization of scheme (12.7), through parameterized approximations to $\Delta\Delta$ [31], and perhaps implicit constructions.

von Neumann Analysis and Stability

The characteristic polynomial for scheme (12.7) is

$$z + (16\mu^2(p_x + p_y)^2 - 2) + z^{-1} = 0 \quad (12.11)$$

in terms of the variables $p_x = \sin^2(\beta_x h/2)$, and $p_y = \sin^2(\beta_y h/2)$. The roots are of unit magnitude under the conditions

$$0 \leq 4\mu^2(p_x + p_y)^2 \leq 1 \quad (12.12)$$

which is satisfied when

$$\mu \leq \frac{1}{4} \quad (12.13)$$

12.2 Realistic Plate Models

12.2.1 Finite Difference Schemes

12.3 Plate Reverberation

Though this book is mainly concerned with sound synthesis based on physical models of musical instruments, one can indeed apply the same techniques to simulate certain audio effects as well. At present there is a good deal of work going into simulations of analog electrical circuits used in classic analog synthesis, both for filtering, and for direct synthesis—as one might expect, virtual analog (as it is called) is again always based on numerical methods for solving systems of differential equations []. Slightly more relevant here is the wide variety of electromechanical effects devices which have been notoriously difficult to emulate digitally. Physical modeling offers an approach to digital versions of effects such as the Leslie speaker, spring reverberation [1, 129], and, the granddaddy of all analog audio effects, plate reverberation [29, 30].

Plate reverb, as it is called, developed in the 1950s [150], and subsequently became the high-end reverberation device of choice for decades, until the advent of digital reverberation. Operation is relatively straightforward—a large metal plate, of varying dimensions, but sometimes as long as two metres, is fed with a dry input signal, and a pickup reads an output signal at a given position on the plate. The typical plate reverberation characteristic is very different from that of a room, mainly due to the absence of strong early reflections, and to the modal distribution, which is nearly uniform—both are results of the inherently dispersive character of the plate itself, as discussed in §??. In spite of this, the plate reverb sound has become one of the most sought after effects in digital audio.

12.4 Soundboards

12.5 Problems

12.6 Programming Exercises

Chapter 13

Nonlinear Plates

Perhaps the most dramatic example of nonlinear behaviour in all of musical acoustics is afforded by the vibration of a metal plate at high amplitudes. Various percussion instruments, and especially gongs and cymbals, exhibit this behaviour. The perceptual results are extremely strong, including effects such as the rapid buildup of high-frequency energy as heard in cymbal crashes, subharmonic generation, as well as pitch glides, which were discussed in some detail in the case of the string in Chapter 7. For sound synthesis purposes, a linear model is wholly insufficient.

There are a variety of models of nonlinear plate vibration; when the plate is thin, and vibration amplitudes are low, all of these reduce to the Kirchhoff model discussed in §12.1. (Plates which appear in a musical setting are generally thin, and so there is little reason to delve into the much more involved topic of thick plate vibration, which, even in the linear case, is orders of magnitude more involved than dealing with simple thin plate models.) Perhaps the simplest nonlinear thin plate model is that of Berger [21], which is discussed in §13.1; this system is a 2D analogue of the Kirchhoff-Carrier, or “tension-modulated” string discussed in §8.1, and the predominant perceptual result of employing such a model is the simulation of pitch glides. Though this model is sufficient in the first instance, and leads to computationally attractive finite difference schemes, it is not capable of rendering the more interesting effects mentioned above, which are defining characteristics of some percussion instruments. To this end, the more complex model of von Karman [178, 247] is introduced in §13.2, as are robust finite difference schemes based on energy conservation properties. Applications to cymbal and gong vibration appear in §13.3.

References for this chapter include: [111, 200, 18, 158, 247, 178, 21, 145, 287, 54, 256, 252, 227, 80, 144, 145, 26, 27, 253, 215]

13.1 The Berger Plate Model

13.1.1 Definition

13.1.2 Energy Analysis

13.1.3 A Finite Difference Scheme

13.1.4 Pitch Glides

13.2 The von Karman Plate Model

One form of the so-called dynamic analogue of the system of von Karman [247], [178] is given by the system

$$\rho H \ddot{u} = -D \Delta \Delta u + \mathcal{L}[\phi, u] \quad (13.1a)$$

$$\Delta \Delta \phi = -\frac{EH}{2} \mathcal{L}[u, u] \quad (13.1b)$$

Here $u(x, y, t)$ is the transverse plate deflection and $\phi(x, y, t)$ is the Airy stress function, both defined over region $(x, y) \in \mathcal{A} \subset \mathbb{R}^2$, and for time $t \geq 0$. Δ , as before, is the Laplacian, and $\Delta \Delta$ the biharmonic operator. E , H , and ρ are as defined for the Berger system in §13.1. The nonlinear operator $\mathcal{L}[\cdot, \cdot]$ is defined in Section 13.2.1. The system is initialized using the values $u(x, y, 0)$ and $\dot{u}(x, y, 0)$.

It is worth noting that there are several variants of the von Karman system; that given above is simplified from the so-called “full” or “complete” system, in which in-plane displacements appear explicitly [145], [178]; all such systems may themselves be derived from even more general forms [287]. Equation (13.1a) reduces, in the absence of the term involving the bracket operator, to the linear Kirchhoff model of plate vibration [111]; the Berger model also may be arrived at, under somewhat more subtle assumptions. All the analysis presented in this article extends, with ease, to the case in which a linear damping term is included—conserved quantities become dissipated, but all the resulting stability analysis remains virtually unchanged. This has been discussed earlier by this author, in the case of nonlinear strings [33], and is covered briefly in Section ??.

13.2.1 The Operator $\mathcal{L}[\cdot, \cdot]$

The operator $\mathcal{L}[\cdot, \cdot]$ is defined, in Cartesian coordinates, as

$$\mathcal{L}[\alpha, \beta] = \frac{\partial^2 \alpha}{\partial x^2} \frac{\partial^2 \beta}{\partial y^2} + \frac{\partial^2 \alpha}{\partial y^2} \frac{\partial^2 \beta}{\partial x^2} - 2 \frac{\partial^2 \alpha}{\partial x \partial y} \frac{\partial^2 \beta}{\partial x \partial y}$$

Clearly,

$$\mathcal{L}[\alpha, \beta] = \mathcal{L}[\beta, \alpha] \quad (13.2)$$

The operator $\mathcal{L}[\cdot, \cdot]$ is bilinear, or linear in either one of its arguments if the other is held fixed. Considering the first argument, for any constants c_1 and c_2 ,

$$\mathcal{L}[c_1 \alpha_1 + c_2 \alpha_2, \beta] = c_1 \mathcal{L}[\alpha_1, \beta] + c_2 \mathcal{L}[\alpha_2, \beta] \quad (13.3)$$

for any functions α_1 , α_2 and β . (It will be linear in the second argument as well by (13.2).) The bilinearity property also implies that

$$(\mathcal{L}[\dot{\alpha}, \alpha]) = 2\mathcal{L}[\dot{\alpha}, \alpha] \quad (13.4)$$

Another important property (for lack of a better term, it will be called “triple self-adjointness”

in this article) is that for any three functions α , β and γ defined over the quarter plane \mathcal{A} ,

$$\begin{aligned} \iint_{\mathcal{A}} \alpha \mathcal{L}[\beta, \gamma] d\sigma &= \iint_{\mathcal{A}} \mathcal{L}[\alpha, \beta] \gamma d\sigma \\ &+ \int_0^{\infty} \alpha_y \beta_{xx} \gamma - \alpha (\beta_{xx} \gamma)_y - \alpha_x \beta_{xy} \gamma + \alpha (\beta_{xy} \gamma)_x dx \\ &+ \int_0^{\infty} \alpha_x \beta_{yy} \gamma - \alpha (\beta_{yy} \gamma)_x - \alpha_y \beta_{xy} \gamma + \alpha (\beta_{xy} \gamma)_y dy \end{aligned} \quad (13.5)$$

In designing a conservative numerical scheme for (13.1), it is crucial that discrete analogues of the properties (13.4) and (13.5) be maintained. The bilinear property (13.3) is the key to efficient computer realizations, as will be discussed in Section 13.2.2.

13.2.2 Energy Analysis

13.2.3 A Finite Difference Scheme

13.3 Cymbals and Gongs

13.4 Problems

13.5 Programming Exercises

Chapter 14

Finite Element Methods

Finite difference methods are well suited to problems defined in either one spatial dimension, or problems defined over a simple geometry in two dimensions (such as rectangular or circular). This covers all string models, 1D tube models, bars, and rectangular or circular percussion instruments. In general, finite differences methods rely on the use of regular grids, and, for finer modeling in irregular geometries, (perhaps geared towards musical instrument design rather than sound synthesis), are not always the best choice. There are certain strategies which may be employed: the use of form-fitting coordinates is one, and domain decomposition [48] is another. On the other hand, for extremely complex geometries, such approaches become unwieldy.

Finite element methods are designed to deal with such problems in irregular geometries; they are based on a distinct point of view; instead of approximating an underlying differential equation, pointwise, over a grid. The solution is approximated using a collection of so-called shape-functions, which may or may not lie in a regular arrangement. In addition, the numerical solution is often approached using integral and variational methods, rather than differential approximations. In the end, however, finite element methods for problems in dynamics operate much as finite difference methods do, through matrix recursions at a given (or perhaps variable) sample rate. Needless to say, the literature on finite element methods is vast; only the briefest introduction can be supplied here. The downside is that writing finite element code is considerably more involved than in the case of finite differences—generally, one must rely on software packages, which can be good or bad, depending on one’s point of view.

In §14.1, the alternate variational form of the model equation is introduced, followed in §14.2 by the definition of several simple shape functions of local support, in both one and two spatial dimensions. Strategies for grid generation are discussed, briefly, in §14.3, and in §14.4 and §14.5, the stiff string and linear plate models are reexamined from the point of finite element simulations. Finally, in §14.6, some general trends in finite element modeling of musical instrument modeling are presented.

References for this chapter include: [11, 86, 66, 199, 180]

- 14.1 Variational Forms**
- 14.2 Shape Functions**
- 14.3 Grid Generation**
- 14.4 The Stiff String Revisited**
- 14.5 The Linear Plate Revisited**
- 14.6 Musical Instrument Modeling**
- 14.7 Problems**
- 14.8 Programming Exercises**

Chapter 15

Spectral Methods

One of the most important drawbacks to the use of time domain methods relates to the effect of numerical dispersion, which leads, at least in the linear case, to mistuning of modal frequencies. Numerical dispersion itself results from insufficient accuracy in a numerical method. While the perceptual importance of numerical dispersion in musical sound synthesis applications, which is generally limited to the upper end of the human hearing range, is a matter of debate, this is a good point to introduce other numerical techniques which have much better accuracy properties.

Finite difference methods and finite element methods are based on essentially local discretizations of a model system; generally, one models the various partial derivatives at a given grid point or node through combinations of values at neighboring points. As was mentioned early on, better accuracy may be achieved as more neighboring points are used in the calculation; this renders the calculations less local. In the most well-known manifestations of spectral methods¹ [48, 260, 100, 195] approximations are global; a derivative calculated at a given grid point will make use of information from the entire spatial domain of the problem. In fact, even the concept of a grid function is not entirely useful in this setting, as the solution itself is considered to be decomposed into an entire set of basis functions—modal synthesis[2], particularly in the form discovered independently by Garber[104], is an example of a very simple spectral method based on a Fourier decomposition.

There are many different forms of spectral methods, dependent chiefly of the types of expansions used to represent the solution (often Fourier or polynomial series, discussed in §15.1), and on the constraints one places on the solution: if the error is to be minimized at a finite collection of points in the spatial domain, one speaks of a collocation method (see §15.2), and if an error with respect to the expansion basis used to represent the solution is minimized, then a Galerkin method results (see §15.3). More general weighted residual methods have been developed, but in this short chapter, only these two forms will be discussed, particularly with reference to their suitability for musical sound synthesis. Spectrally accurate derivative approximations, as well as the use of non-uniform Chebyshev grids, and fast evaluation through the use of the fast Fourier transform are discussed in §15.4. As spectrally accurate approximations are generally applied to spatial derivatives, global accuracy may be achieved through the use of standard high-order time differencing strategies—these are briefly summarized in §15.5. Finally, musical applications to the stiff string and linear plate problems are presented, in §15.6 and 15.7, respectively.

¹It is worth being quite clear on the use of the word “spectral,” in this case, which is not used in the sense of spectral modeling. The word spectral refers to extremely high accuracy of numerical methods—though they can be based on the use of Fourier series, in most cases of interest they are rather generally based on more general polynomial series approximations.

References for this chapter include [260, 100, 48, ?, 195, 145, 144, 2, 104, 109]

15.1 Basis Functions

15.1.1 Fourier Series

15.1.2 Polynomials

15.2 Collocation Methods

15.3 Galerkin Methods

15.4 Spectral Derivatives

15.4.1 As Limiting Case of Finite Difference Approximations

15.4.2 Chebyshev Grids

15.4.3 Fast Fourier Transforms

15.5 High Order Time Stepping Methods

15.5.1 Forward and Backward Euler

15.5.2 Adams-Bashforth Methods

15.5.3 Adams-Moulton Methods

15.5.4 Runge-Kutta Methods

15.6 The Stiff String Again Revisited

15.7 The Linear Plate Again Revisited

15.8 Problems

15.9 Programming Exercises

Chapter 16

Concluding Remarks

16.1 Complexity of Musical Systems

After having now examined the simulation of a wide variety of systems in musical acoustics, it is useful to return to the subject of computational complexity, which was introduced briefly in §1.3.3.

16.1.1 A Family of LSI Musical Systems

A rather large class of musical systems is described by an equation of the form

$$u_{tt} + c^2 (-\nabla_d^2)^p u = 0 \tag{16.1}$$

Here, ∇_d is the d -dimensional Laplacian operator, and p is a constant equal to 1 for non-stiff systems, and equal to 2 for stiff systems. Eq. (16.1) is assumed spatially-nondimensionalized such that it is defined over a region of unit d -dimensional volume. When $p = 1$, Eq. 16.1 is the wave equation, and c can be thought of as a constant wave speed, and when $p = 2$, it is the equation of motion of an ideal bar (in 1D) and a thin plate (in 2D). Under different choices of d and p , the systems which are described to a very rough approximation are as given in Table ??.

p/d	1	2	3
1	strings, acoustic tubes	membranes	room acoustics
2	thin bars	thin plates	

Eq. (16.1) is studied here primarily because of simplicity of analysis, but strictly speaking, it only describes systems which are linear, shift-invariant, lossless, and isotropic. Most of the analysis which follows on computational complexity is affected relatively little by the addition of loss (which is normally quite small in most musical systems), spatial variation, and anisotropy. There are also, of course, other LSI systems which have been described in this book, such as strings with stiffness, which are not described by (16.1), but again, the analysis may be extended to cover these cases. When nonlinearities come in to the picture, however, it is not at all clear how to arrive at bounds on complexity—some comments on this subject appear in .

A starting point for analysis of complexity in both the case of modal methods, and those based on direct integration methods, is the dispersion relation for Eq. (16.1), which is

$$\omega = c|\beta_d|^p \tag{16.2}$$

where β_d is a d -dimensional wavenumber.

16.1.2 Sampling Theorems and Complexity Bounds

It is possible to arrive at estimates of complexity for direct numerical simulation methods for Eq. 16.1, through a dual application of Shannon's sampling theorem in time and space. It is useful to rewrite the dispersion relation above in terms of frequency $f = \omega/2\pi$ and wavelength $|\beta_d| = 2\pi/\Lambda$ as

$$f = \frac{c(2\pi)^{p-1}}{\Lambda^p} \quad (16.3)$$

Beginning from a choice f_s for the audio sample rate, it should be clear that one would expect that, at best, any numerical method should be capable of simulating frequencies up to the Nyquist, or $f_s/2$ —one might argue that one could perhaps save on computation by computing results over a smaller bandwidth, but, as has been shown at various points in this book, this is really a waste of audio bandwidth, and one could save far more work by simply reducing the sample rate, and then perhaps upsampling the resulting sound output in a post-processing step.

The question then turns to the range of wavelengths which must be resolved by the numerical method. At sampling frequency f_s , the shortest such wavelength will be given, by the dispersion relation (16.3), as

$$\Lambda_{min} = 2\pi^{1-1/p} \left(\frac{c}{f_s} \right)^{1/p} \quad (16.4)$$

For a numerical method operating over a grid, it should be clear that the spacing between adjacent points must be on the order of less than half the smallest wavelength to be rendered, or $\Lambda_{min}/2$ —this follows directly from sampling theorems in the frequency domain, and there will be slight variations in the required density depending on the type of grid to be used, especially in the multidimensional case.

For a unit volume, and given the requirement above on the spacing between grid points, the number of grid points needed to fill the space will be $1/(\Lambda_{min}/2)^d$, and, for a two-step difference scheme, the total number of memory locations required will be twice this, or approximately

$$N_{fd} \approx 2\pi^{d/p-d} \left(\frac{f_s}{c} \right)^{d/p} \quad (16.5)$$

For a sparse finite difference scheme, the number of operations O_{fd} per second required will be approximately

$$O_{fd} = \kappa f_s N_{fd} \approx 2\kappa f_s \pi^{d/p-d} \left(\frac{f_s}{c} \right)^{d/p} \quad (16.6)$$

where κ is a parameter which depends on the type of scheme—generally, it will lie in the range of between 1 and 20 for most of the systems encountered in musical acoustics, and for reasonably simple schemes.

It is worth noting the dependence on the parameters d and p , which behave in opposite respects with regard to complexity: memory and the operation count increase severely with the dimension, but are reduced with increasing stiffness; this is the reason that, say, full room acoustics simulation is a massive task, whereas a model of plate reverberation is manageable, and even possible in real time.

16.1.3 Modal Representations and Complexity Bounds

Again beginning from a choice f_s for the audio sampling rate, one may again make use of the dispersion relation (16.3) in order to arrive at a bound on computational complexity, this time

using principles of modal density—see, e.g., the chapter by Weinreich for more information [283]. Suppose, for simplicity, that (16.1) is defined over a unit hypercube in d dimensions, and that the boundary conditions are of fixed/simply supported type (i.e., $u = 0$ on the boundary, and additionally $\nabla_d^2 u = 0$ on the boundary if $p = 2$). The wavenumber β_d must then be of the form $\beta_d = \pi \mathbf{n}_d$, where $\mathbf{n}_d = [n_1, n_2, \dots, n_d]$ for (non-negative) integers n_1, n_2, \dots, n_d . The dispersion relation may then be rewritten as

$$|\mathbf{n}_d| = \pi^{1/p-1} \left(\frac{2f_{\mathbf{n}}}{c} \right)^{1/p} \quad (16.7)$$

where the frequencies $f = f_{\mathbf{n}}$ are now limited to a countably infinite set indexed by the vector \mathbf{n} . It now remains to determine the of frequencies $f_{\mathbf{n}}$ which lie below the Nyquist frequency $f_s/2$. As $|\mathbf{n}_d|$ is the Euclidean length of the vector \mathbf{n}_d in a d -dimensional space, the number of modes will then be that number which lies within the positive d -dimensional quadrant of the sphere of radius $\pi^{1/p-1} \left(\frac{f_s}{c} \right)^{1/p}$. Given that each mode possesses two degrees of freedom, the total number of degrees of freedom (or number of units of memory required) will be approximately

$$N_m \approx 2\pi^{d/p-d} \left(\frac{f_s}{c} \right)^{d/p} \quad (16.8)$$

for small values of d . This is the same as the bound obtained using principles of sampling theory in the previous section.

Though the above derivation appears to depend on several restrictive conditions, it is in fact more general than it appears to be. Addressing each in turn: The issue of degeneracy of modes is minor, in that a) the proportion of such modes which are degenerate under the conditions above vanishes in the limit of large $|\mathbf{n}|$, leading to the same expression for the number of modes, and furthermore b), degeneracy only occurs under very precise geometrical conditions which occur rarely if at all in practice. The issue of the special choice of geometry (i.e., a unit hypercube) in order to obtain a set of modes which may be easily ordered (and thus counted) is also minor: any unit volume will possess the same distribution of modes in the limit of high frequencies. Similarly, different choices of boundary conditions will affect only the frequencies of the lowest modes, and not the distribution in the limit of high frequencies.

16.2 Physical Modeling Sound Synthesis Methods: Strengths and Weaknesses

16.2.1 Modal Synthesis

While intuitively appealing, and perceptually relevant, modal synthesis has a number of serious shortcomings:

First, modal techniques are strictly applicable only to linear and time invariant (LTI) systems, i.e., those for which an eigenvalue problem may be derived. Though this does cover a number of useful cases in musical acoustics, there are many others, often among the most musically interesting, which are not LTI. Under weakly nonlinear conditions, it is possible to draw some mainly qualitative conclusions using perturbation analysis about modal solutions (though this has not, apparently, been done in the context of musical sound synthesis). Perceptual effects such as pitch glides in strings and struck plates may be analyzed in this way. But for stronger nonlinearities, such as those that occur in struck gongs or cymbals, such analysis becomes unwieldy, and of questionable validity.

Second, although for certain simple spatially-uniform structures (such as, e.g., an ideal string or

rectangular membrane under fixed or free termination, or an ideal plate under simply-supported conditions), the eigenfunctions and eigenfrequencies (modal shapes and frequencies) may be expressed in closed form, for almost all other geometries and boundary conditions, they can not be, and must be computed numerically, generally before run time. For large problems, this calculation can be enormous. For a static expansions onto (and projections of) the modal functions, it is sufficient to compute the weighting coefficients alone. This is often sufficient in mainstream applications, but definitely not in musical sound synthesis, in which case dynamic input and output parameters are a crucial feature (e.g., if a membrane is to be struck or its output taken at varying locations); the weighting coefficients must be recomputed with any such variation. (The alternative is to simply store all the eigenfunctions, but this can become prohibitively costly in terms of memory use.)

Third, it is important to point out that there is no efficiency advantage, either in terms of the operation count or memory use, in using modal expansions relative to time-domain methods; this follows from

In the cases in which they may be applied, modal methods do, however, possess two significant advantages relative to time-domain techniques. First, because the modal decomposition may be thought of as essentially breaking a system down into a parallel combination of a set of second order linear oscillators, stability is extremely easy to verify; such is not the case

All these difficulties stem from the complexity of the back-and-forth between the time and frequency domains.

16.2.2 Digital Waveguides

16.2.3 Lumped Mass Spring Networks

On the other hand, working with lumped elements in order to describe a distributed system is somewhat cumbersome, especially when an extremely elegant mathematical representation of the distributed system is available (this is often the case in musical acoustics). The distinction between the use of lumped and distributed modeling can be described as *physical*, in the former case, and *mathematical* in the latter. The relative advantages of the two approaches was well-expressed many years ago by Ames [3]:

...The specialist sometimes finds the physical approach useful in motivating further analyses. In such a modus operandi the discrete (physical) model is given the lumped physical characteristics of the continuous system. For example, a heat conducting slab could be replaced by a network of heat conducting rods. The governing equations are then developed by direct application of the physical laws to the discrete system. On the other hand, in the mathematical approach the continuous formulation is transformed to a discrete formulation by replacing derivatives by, say, finite difference approximations. When the continuous problem formulation is already available this procedure is more flexible...

Ames was not speaking about musical systems, but replacing “heat conducting slab” and “heat conducting rods” above by “membrane” and “masses and springs,” one can translate his point to the musical sound synthesis setting. As one might expect, when a lumped network is intended to behave as a distributed object, an equivalent finite difference scheme can be shown to exist—see §?? and §11.1.7 for two examples. One could proceed farther along these lines, but it quickly

becomes difficult to deal with important numerical issues such as accuracy, or global approximations to derivatives, or complex boundary conditions; a fully distributed model is indispensable in such cases. In many ways, however, it is not fair to compare the lumped network formalism with that of direct simulation, because the goals are somewhat different.

16.2.4 Direct Simulation

16.3 The Future

Appendix A

MATLAB Code Examples

In this appendix, various simple code fragments are provided. All can be viewed as prototypes for physical modeling sound synthesis. The coding style reflects something of a compromise between efficiency on the one hand, and brevity and intelligibility on the other. The choice of Matlab as a programming environment definitely reflects the latter sensibility, though the use of Matlab as an actual synthesis engine is not recommended. Some of these examples make use constructs and features which need not appear in a code fragment intended for synthesis, including various calls to plotting functions, as well as the demonstration of energy conservation in some cases. It should be clear, in all cases, which elements of these examples may be neglected in an actual implementation. For the sake of brevity, these examples are too crude for actual synthesis purposes, but many features, discussed at various points in the texts and exercises, may be added.

A.1 The Simple Harmonic Oscillator

```
% matlab script sho.m
% finite difference scheme for simple harmonic oscillator
%%%%%%%% begin global parameters
SR = 44100;                % sample rate (Hz)
f0 = 200;                 % fundamental frequency (Hz)
TF = 0.05;                % duration of simulation (s)
u0 = 0.1;                 % initial displacement
v0 = 1;                   % initial velocity
%%%%%%%% end global parameters
% check that stability condition is satisfied
if(SR<=pi*f0)
    error('Stability condition violated');
end
% derived parameters
k = 1/SR;                 % time step
u1 = u0+k*v0;             % derive value of time series at n=1;
coef = 2-k^2*(2*pi*f0)^2; % scheme update coefficient
NF = floor(TF*SR);        % duration of simulation (samples)
% initialize state of scheme
u=0;                      % current value of time series
u_last = u1;              % last value of time series
u_last2 = u0;             % one before last value of time series
% initialize readout
out = zeros(NF,1); out(1) = u0; out(2) = u1;
```

```

%%%%%%%% start main loop
for n=3:Nf
    u=coef*u_last-u_last2;           % difference scheme calculation
    out(n) = u;                       % read value to output vector
    u_last2 = u_last; u_last = u;    % update state of difference scheme
end
%%%%%%%% end main loop
% plot output waveform
plot([0:Nf-1]*k, out, 'k');
xlabel('t'); ylabel('u'); title('SH0: Difference Scheme Output');
axis tight
% play sound
soundsc(out,SR);

```

A.2 Hammer Collision with Mass-Spring System

```

% matlab script idealbarfd.m
% hammer collision with a mass-spring system
%%%%%%%% global parameters
SR = 44100;           % sample rate
xH0 = -0.01; vH0 = 10; % initial conditions of hammer
Tf = 0.05;           % duration of simulation (s)
w0 = 1000;          % angular frequency of mass-spring system
wH = 300;           % stiffness parameter for hammer
MR = 10;            % hammer/target mass ratio
alpha = 2;          % hammer stiffness nonlinearity exponent
%%%%%%%% derived parameters
Nf = floor(Tf*SR);
k = 1/SR;
%%%%%%%% initialization
uHlast2 = xH0; uHlast = xH0+k*vH0;
ulast2 = 0; ulast = 0;
out = zeros(Nf,1); f = zeros(Nf,1);
out(1) = ulast2; out(2) = ulast;
%%%%%%%% main loop
for n=3:Nf
    if(uHlast>ulast)
        f(n-1) = wH^(1+alpha)*(uHlast-ulast)^alpha;
    else f(n-1) = 0;
    end
    uH = 2*uHlast-uHlast2-k^2*f(n-1);
    u = 2*ulast-ulast2-w0^2*k^2*ulast+MR*k^2*f(n-1);
    out(n) = u;
    ulast2 = ulast; ulast = u;
    uHlast2 = uHlast; uHlast = uH;
end
%%%%%%%% plots of displacement of target mass and force
subplot(2,1,1)
plot([1:Nf]*k, out, 'k'); title('Position of Target Mass'); xlabel('time (s)');
axis tight
subplot(2,1,2)
plot([1:Nf]*k, f, 'k'); title('Hammer Force'); xlabel('time');
axis tight

```



```

lambda = 1; % Courant number
rp = 0.5; % position of readout (0-1)
%%%%%%%% end global parameters
%%%%%%%% begin derived parameters
gamma = 2*f0; % wave equation free parameter
k = 1/SR; % time step
NF = floor(SR*TF); % duration of simulation (samples)
h = gamma*k/lambda; % grid spacing
N = floor(1/h); % number of subdivisions of spatial domain
h = 1/N; % reset h
lambda = gamma*k/h; % reset Courant number
s0 = 2*(1-lambda^2); s1 = lambda^2; % scheme parameters
rp_int = 1+floor(N*rp); % rounded grid index for readout
rp_frac = 1+rp/h-rp_int; % fractional part of readout location
%%%%%%%% initialize grid functions and output
u = zeros(N+1,1); u1 = zeros(N+1,1); u2 = zeros(N+1,1);
out = zeros(NF,1);
%%%%%%%% create raised cosine
rc = zeros(N+1,1);
for qq=1:N+1
    pos = (qq-1)*h; dist = ctr-pos;
    if(abs(dist)<=wid/2)
        rc(qq) = 0.5*(1+cos(2*pi*dist/wid));
    end
end
%%%%%%%% set initial conditions
u2 = u0*rc; u1 = (u0+k*v0)*rc;
%%%%%%%% start main loop
for n=3:NF
    u(2:N) = -u2(2:N)+s0*u1(2:N)+s1*(u1(1:N-1)+u1(3:N+1)); % scheme calculation
    out(n) = (1-rp_frac)*u(rp_int)+rp_frac*u(rp_int+1); % readout
    u2 = u1; u1 = u; % update of grid variables
end
%%%%%%%% end main loop
% plot output waveform
plot([0:NF-1]*k, out, 'k');
xlabel('t'); ylabel('u'); title('1D Wave Equation: Difference Scheme Output');
axis tight
% play sound
soundsc(out,SR);

```

A.5 The 1D Wave Equation: Modal Synthesis

```

% matlab script waveeq1dfd.m
% modal synthesis method for the 1D wave equation
% *fixed boundary conditions
% *raised cosine initial conditions
%%%%%%%% begin global parameters
SR = 44100; % sample rate (Hz)
f0 = 80; % fundamental frequency (Hz)
TF = 1; % duration of simulation (s)
ctr = 0.7; % center location of excitation (0-1)
wid = 0.05; % width of excitation
u0 = 1; % maximum initial displacement
v0 = 2; % maximum initial velocity
rp = 0.3; % position of readout (0-1)

```

```

%%%%%%%%% end global parameters
%%%%%%%%% begin derived parameters
k = 1/SR; % time step
NF = floor(SR*TF); % duration of simulation (samples)
N = floor(0.5*SR/f0); % number of modes
temp = 2*pi*[1:N]'*f0/SR;
coeff = 2*cos(temp);
outexp = sin([1:N]*pi*rp);
%%%%%%%%% initialize grid functions and output
U = zeros(N,1); U1 = zeros(N,1); U2 = zeros(N,1);
out2 = zeros(NF,1);
%%%%%%%%% create raised cosine and find Fourier coefficients
rc = zeros(N,1);
for qq=1:N
    pos = (qq-1)/N; dist = ctr-pos;
    if(abs(dist)<=wid/2)
        rc(qq) = 0.5*(1+cos(2*pi*dist/wid));
    end
end
rcfs = -imag(fft([rc; zeros(N,1)])); rcfs = 2*rcfs(2:N+1)/N;
%%%%%%%%% set initial conditions
U2(1:N) = u0*rcfs;
U1(1:N) = (u0*cos(temp)+v0*sin(temp)./(2*pi*[1:N]'*f0)).*rcfs;
%%%%%%%%% start main loop
for n=3:NF
    U = -U2+coeff.*U1; % scheme calculation
    out(n) = outexp*U; % readout
    U2 = U1; U1 = U; % update of modal weights
end
%%%%%%%%% end main loop
% plot output waveform
plot([0:NF-1]*k, out, 'k');
xlabel('t'); ylabel('u'); title('1D Wave Equation: Modal Synthesis Output');
axis tight
% play sound
soundsc(out,SR);

```

A.6 The 1D Wave Equation: Digital Waveguide Synthesis

```

% matlab script waveeq1ddw.m
% digital waveguide method for the 1D wave equation
% *fixed boundary conditions
% *raised cosine initial conditions
% *linear interpolation
%%%%%%%%% begin global parameters
SR = 44100; % sample rate (Hz)
f0 = 447; % fundamental frequency (Hz)
TF = 1; % duration of simulation (s)
ctr = 0.7; % center location of excitation (0-1)
wid = 0.1; % width of excitation
u0 = 1; % maximum initial displacement
v0 = 3; % maximum initial velocity
rp = 0.3; % position of readout (0-1)
%%%%%%%%% end global parameters
%%%%%%%%% begin derived parameters

```

```

k = 1/SR; % time step
NF = floor(SR*TF); % duration of simulation (samples)
N = floor(0.5*SR/f0); % length of delay lines
rp_int = 1+floor(N*rp); % rounded grid index for readout
rp_frac = 1+rp*N-rp_int; % fractional part of readout location
%%%%% initialize delay lines and output
wleft = zeros(N,1); wright = zeros(N,1);
out = zeros(NF,1);
%%%%% create raised cosine and integral
rc = zeros(N,1); rcint = zeros(N,1);
for qq=1:N
    pos = (qq-1/2)/N; dist = ctr-pos;
    if(abs(dist)<=wid/2)
        rc(qq) = 0.5*(1+cos(2*pi*dist/wid));
    end
    if(qq>1)
        rcint(qq) = rcint(qq-1)+rc(qq)/N;
    end
end
%%%%% set initial conditions
wleft = 0.5*(u0*rc+v0*rcint/(2*f0));
wright = 0.5*(u0*rc-v0*rcint/(2*f0));
%%%%% start main loop
for n=3:NF
    temp1 = wright(N); temp2 = wleft(1);
    wright(2:N) = wright(1:N-1);
    wleft(1:N-1) = wleft(2:N);
    wright(1) = -temp2; wleft(N) = -temp1;
    % readout
    out(n) = (1-rp_frac)*(wleft(rp_int)+wright(rp_int))+rp_frac*(wleft(rp_int+1)+wright(rp_int+1));
end
%%%%% end main loop
% plot output waveform
plot([0:NF-1]*k, out, 'k');
xlabel('t'); ylabel('u'); title('1D Wave Equation: Digital Waveguide Synthesis Output');
axis tight
% play sound
soundsc(out,SR);

```

A.7 The Ideal Bar

```

% matlab script idealbarfd.m
% finite difference scheme for the ideal bar equation
% *clamped/pivoting boundary conditions
% *raised cosine initial conditions
%%%%% begin global parameters
SR = 44100; % sample rate (Hz)
K = 10; % stiffness parameter
TF = 1; % duration of simulation (s)
ctr = 0.5; % center location of excitation (0-1)
wid = 0.1; % width of excitation
u0 = 1; % maximum initial displacement
v0 = 0; % maximum initial velocity
mu = 0.5; % scheme free parameter
rp = 0.5; % position of readout (0-1)
bc = [1 1]; % boundary condition type, [left right] with

```

```

% 1: clamped, 2: pivoting

%%%%%%%% end global parameters
%%%%%%%% begin derived parameters
k = 1/SR; % time step
NF = floor(SR*TF); % duration of simulation (samples)
h = sqrt(K*k/mu); % grid spacing
N = floor(1/h); % number of subdivisions of spatial domain
h = 1/N; % reset h
mu = K*k/h^2; % reset Courant number
s0 = 2*(1-3*mu^2); s1 = 4*mu^2; s2 = -mu^2; % scheme parameters
rp_int = 1+floor(N*rp); % rounded grid index for readout
rp_frac = 1+rp/h-rp_int; % fractional part of readout location
%%%%%%%% initialize grid functions and output
u = zeros(N+1,1); u1 = zeros(N+1,1); u2 = zeros(N+1,1);
out = zeros(NF,1);
%%%%%%%% create raised cosine
rc = zeros(N+1,1);
for qq=1:N+1
    pos = (qq-1)*h; dist = ctr-pos;
    if(abs(dist)<=wid/2)
        rc(qq) = 0.5*(1+cos(2*pi*dist/wid));
    end
end
%%%%%%%% set initial conditions
u2 = u0*rc; u1 = (u0+k*v0)*rc;
%%%%%%%% start main loop
for n=3:NF
    % scheme calculation (interior)
    u(3:N-1) = -u2(3:N-1)+s0*u1(3:N-1)+s1*(u1(2:N-2)+u1(4:N))...
        +s2*(u1(1:N-3)+u1(5:N+1));
    % calculations at boundary points
    if(bc(1)==1)
        u(2) = -u2(2)+s0*u1(2)+s1*u1(3)+s2*u1(4);
    else
        u(2) = -u2(2)+(s0+s2)*u1(2)+s1*u1(3)+s2*u1(4);
    end
    if(bc(2)==1)
        u(N) = -u2(N)+s0*u1(N)+s1*u1(N-1)+s2*u1(N-2);
    else
        u(N) = -u2(N)+(s0+s2)*u1(N)+s1*u1(N-1)+s2*u1(N-2);
    end
    out(n) = (1-rp_frac)*u(rp_int)+rp_frac*u(rp_int+1); % readout
    u2 = u1; u1 = u; % update of grid variables
end
%%%%%%%% end main loop
% plot output waveform
plot([0:NF-1]*k, out, 'k');
xlabel('t'); ylabel('u'); title('Ideal Bar Equation: Difference Scheme Output');
axis tight
% play sound
soundsc(out,SR);

```

A.8 The Stiff String

```

% matlab script ssfd.m
% finite difference scheme for the stiff string
% *clamped boundary conditions
% *raised cosine initial conditions
% *stereo output
% *implicit scheme: matrix update form

```

```

% *two-parameter frequency-dependent loss
%%%%% begin global parameters
SR = 44100; % sample rate(Hz)
B = 0.001; % inharmonicity parameter (>0)
f0 = 50; % fundamental(Hz)
TF = 2; % duration of simulation(s)
ctr = 0.1; % center location of excitation(0-1)
wid = 0.05; % width of excitation
u0 = 0; % maximum initial displacement
v0 = 10; % maximum initial velocity
rp = [0.3 0.7]; % positions of readout(0-1)
loss = [100, 5; 1000, 2]; % loss [freq.(Hz), T60(s), freq.(Hz), T60(s)]
theta = 1; % implicit scheme free parameter (>0.5)
%%%%% begin derived parameters
k = 1/SR; % time step
NF = floor(SR*TF); % duration of simulation (samples)
gamma = 2*f0; K = sqrt(B)*(gamma/pi); % set scheme parameters
% find grid spacing
h = sqrt((gamma^2*k^2+sqrt(gamma^4*k^4+16*K^2*k^2*(2*theta-1)))/(2*(2*theta-1)));
N = floor(1/h); % number of subdivisions of spatial domain
h = 1/N; % reset h
mu = K*k/h^2; lambda = gamma*k/h; % reset scheme parameters
rp_int = 1+floor(N*rp); % rounded grid indeces for readout
rp_frac = 1+rp/h-rp_int; % fractional part of readout locations
% set scheme loss parameters
zeta1 = (-gamma^2+sqrt(gamma^4+4*K^2*(2*pi*loss(1,1))^2))/(2*K^2);
zeta2 = (-gamma^2+sqrt(gamma^4+4*K^2*(2*pi*loss(2,1))^2))/(2*K^2);
sig0 = 6*log(10)*(-zeta2/loss(1,2)+zeta1/loss(2,2))/(zeta1-zeta2);
sig1 = 6*log(10)*(1/loss(1,2)-1/loss(2,2))/(zeta1-zeta2);
%%%%% initialize grid functions and output (stereo)
u = zeros(N-1,1); u1 = zeros(N-1,1); u2 = zeros(N-1,1);
out = zeros(NF,2);
%%%%% create update matrices
M = sparse(toeplitz([theta (1-theta)/2 zeros(1,N-3)]));
A = M+sparse(toeplitz([sig1*k/(h^2)+sig0*k/2 -sig1*k/(2*h^2) zeros(1,N-3)]));
C = M+sparse(toeplitz([-sig1*k/(h^2)-sig0*k/2 sig1*k/(2*h^2) zeros(1,N-3)]));
B = 2*M+sparse(toeplitz([-2*lambda^2-6*mu^2 lambda^2+4*mu^2 -mu^2 zeros(1,N-4)]));
%%%%% create raised cosine
rc = zeros(N-1,1);
for qq=1:N-1
    pos = qq*h; dist = ctr-pos;
    if(abs(dist)<=wid/2)
        rc(qq) = 0.5*(1+cos(2*pi*dist/wid));
    end
end
%%%%% set initial conditions
u2 = u0*rc; u1 = (u0+k*v0)*rc;
%%%%% start main loop
for n=3:NF
    u = A\b(B*u1-C*u2);
    out(n,:) = (1-rp_frac).*u(rp_int)'+rp_frac.*u(rp_int+1)'; % readout
    u2 = u1; u1 = u; % update
end
%%%%% end main loop
% plot output waveform
subplot(2,1,1); plot([0:NF-1]*k, out(:,1), 'k');
xlabel('t'); ylabel('u');

```

```

title('Stiff String Equation: Difference Scheme Output (left)');
subplot(2,1,2); plot([0:Nf-1]*k, out(:,2), 'k');
xlabel('t'); ylabel('u');
title('Stiff String Equation: Difference Scheme Output (right)');
axis tight
% play sound
soundsc(out,SR);

```

A.9 The Kirchhoff-Carrier Equation

```

% matlab script kcfd.m
% finite difference scheme for the Kirchhoff-Carrier equation
% *fixed boundary conditions
% *triangular initial conditions
clear all
close all
%%%%%%%% begin global parameters
SR = 44100; % sample rate (Hz)
f0 = 200; % fundamental frequency (Hz)
alpha = 0.01; % nonlinear string parameter
TF = 1; % duration of simulation (s)
ctr = 0.5; % center location of excitation (0-1)
u0 = 0.05; % maximum initial displacement
v0 = 0; % maximum initial velocity
lambda = 0.5; % Courant number
rp = 0.5; % position of readout (0-1)
%%%%%%%% end global parameters
%%%%%%%% begin derived parameters
gamma = 2*f0; % wave equation free parameter
k = 1/SR; % time step
NF = floor(SR*TF); % duration of simulation (samples)
h = gamma*k/lambda; % grid spacing
N = floor(1/h); % number of subdivisions of spatial domain
h = 1/N; % reset h
lambda = gamma*k/h; % reset Courant number
rp_int = 1+floor(N*rp); % rounded grid index for readout
rp_frac = 1+rp/h-rp_int; % fractional part of readout location
%%%%%%%% initialize grid functions and output
u = zeros(N+1,1); u1 = zeros(N+1,1); u2 = zeros(N+1,1);
out = zeros(NF,1);
%%%%%%%% create triangular function
tri = zeros(N+1,1);
for qq=1:N+1
    pos = (qq-1)*h;
    if(pos<=ctr)
        tri(qq) = pos/ctr;
    else tri(qq) = (pos-1)/(ctr-1);
    end
end
%%%%%%%% set initial conditions
u2 = u0*tri; u1 = (u0+k*v0)*tri;
%%%%%%%% start main loop
for n=3:Nf
    % calculate nonlinearity g
    u1x = (u1(2:N+1)-u1(1:N))/h; u1xx = (u1x(2:N)-u1x(1:N-1))/h;
    g = (1+h*sum(u1x.*u1x)/(2*alpha))/(1+k^2*gamma^2*h*sum(u1xx.*u1xx)/(4*alpha));

```

```

    u(2:N) = 2*u1(2:N)-u2(2:N)+g*gamma^2*k^2*u1xx(1:N-1);      % scheme calculation
    out(n) = (1-rp_frac)*u(rp_int)+rp_frac*u(rp_int+1);        % readout
    u2 = u1; u1 = u;                                           % update of grid variables
end
%%%%% end main loop
% plot output waveform
plot([0:Nf-1]*k, out, 'k');
xlabel('t'); ylabel('u'); title('Kirchhoff-Carrier Equation: Difference Scheme Output');
axis tight
% play sound
soundsc(out,SR);

```

A.10 Vocal Tract Synthesis

A.11 The 2D Wave Equation

```

% matlab script waveeq2dloss.m
% finite difference scheme for the 2D wave equation with loss
% *fixed boundary conditions
% *raised cosine initial conditions
% *bilinear interpolation
%%%%%% begin global parameters
SR = 44100;          % sample rate(Hz)
gamma = 1000;       % fundamental(Hz)
TF = 2;             % duration of simulation(s)
ctr = [0.3,0.5];    % center location of excitation(0-1)
wid = 0.1;          % width of excitation
u0 = 0;             % maximum initial displacement
v0 = 10;            % maximum initial velocity
rp = [0.3 0.7];     % position of readout([0-1,0-1])
T60 = 4;            % loss [freq.(Hz), T60(s), freq.(Hz), T60(s)]
epsilon = 0.6;      % domain aspect ratio
lambda = 1/sqrt(2); % Courant number
%%%%%% begin derived parameters
k = 1/SR;           % time step
NF = floor(SR*TF);  % duration of simulation (samples)
h = sqrt(2)*gamma*k; % find grid spacing
Nx = floor(1/h);    % number of x-subdivisions of spatial domain
Ny = floor(epsilon/h); % number of y-subdivisions of spatial domain
h = 1/Nx;           % reset h
lambda = gamma*k/h; % reset Courant number
sig0 = 6*log(10)/T60; % scheme parameters
s0 = (2-4*lambda^2)/(1+sig0*k);
s1 = lambda^2/(1+sig0*k);
t0 = -(1-sig0*k)/(1+sig0*k);
rp_int = 1+floor([Nx Ny].*rp); % rounded grid indices for readout
rp_frac = 1+rp/h-rp_int; % fractional part of readout locations
%%%%%% initialize grid functions and output
u = zeros(Nx+1,Ny+1); u1 = zeros(Nx+1,Ny+1); u2 = zeros(Nx+1,Ny+1);
out = zeros(NF,2);
%%%%%% create 2D raised cosine
rc = zeros(Nx+1,Ny+1);
for qq=0:Nx
    for rr=0:Ny
        pos = [qq*h rr*h]; dist = norm(ctr-pos);

```

```

        if(abs(dist)<=wid/2)
            rc(qq+1,rr+1) = 0.5*(1+cos(2*pi*dist/wid));
        end
    end
end
%%%%% set initial conditions
u2 = u0*rc; u1 = (u0+k*v0)*rc;
%%%%% start main loop
for n=3:NF
    u(2:Nx,2:Ny) = s1*(u1(3:Nx+1,2:Ny)+u1(1:Nx-1,2:Ny)+u1(2:Nx,3:Ny+1)+u1(2:Nx,1:Ny-1))...
        +s0*u1(2:Nx,2:Ny)+t0*u2(2:Nx,2:Ny);
    out(n,:) = (1-rp_frac(1))*(1-rp_frac(2))*u(rp_int(1),rp_int(2))+...
        (1-rp_frac(1))*rp_frac(2)*u(rp_int(1),rp_int(2)+1)+...
        rp_frac(1)*(1-rp_frac(2))*u(rp_int(1)+1,rp_int(2))+...
        rp_frac(1)*rp_frac(2)*u(rp_int(1)+1,rp_int(2)+1);
    u2 = u1; u1 = u; % update
end
%%%%% end main loop
% plot output waveform
plot([0:NF-1]*k, out, 'k');
xlabel('t'); ylabel('u');
title('2D Wave Equation with Loss: Difference Scheme Output');
axis tight
% play sound
soundsc(out,SR);

```

A.12 Plate Reverberation

A.13 Gong

Appendix B

List of Symbols

Symbol	definition
A	string cross-sectional area
\mathfrak{B}	boundary term (continuous)
\mathfrak{b}	boundary term (discrete)
c	wave speed (m/sec.)
c_{rc}	raised cosine distribution
c_{tri}	triangular distribution
c	wave speed (m/s)
C	capacitance (Farads)
D	plate parameter = $EH^3/12(1 - \nu^2)$
$\mathbf{D}\bullet$	spatial differentiation matrix (various)
d_+, d_-	endpoints of spatial domain (1D)
\mathcal{D}	spatial domain
$\underline{\mathcal{D}}$	discrete spatial domain without right end point
$\overline{\mathcal{D}}$	discrete spatial domain without left end point
e_{d+}	forward shift operator in coordinate direction d
e_{d-}	backward shift operator in coordinate direction d
E	Young's modulus
f	frequency (Hz)
f	force (N)
f_s	sample rate (Hz)
f_0	fundamental frequency (Hz)
F	force/mass (m/s^2)
\mathfrak{g}	averaged nonlinearity, difference scheme for Kirchhoff-Carrier equation
\mathfrak{G}	averaged nonlinearity, Kirchhoff-Carrier equation
h	grid spacing
H	plate thickness
\mathfrak{H}	total system energy (continuous)
\mathfrak{h}	total system energy (discrete)
I	interpolation operator
I	bar moment of inertia
\mathbf{I}	identity matrix
k	time step (s)
K	stiffness
\mathbf{K}	stiffness matrix

l	spatial index to grid function
L	string length (m)
L	inductance (Henrys)
M	mass (kg)
\mathbf{M}	mass matrix
n	time index (integer)
O	big oh, order of
P	symbol, partial differential equation
P_d	symbol, finite difference scheme
\mathbb{R}	set of real numbers $[-\infty, \infty]$
\mathbb{R}^+	set of non-negative real numbers $[0, \infty]$
s	complex frequency variable (Hz)
t	time variable (sec.)
T_0	nominal string tension
T_{60}	60 dB decay time (sec.)
\mathcal{T}	kinetic energy (continuous)
\mathbf{t}	kinetic energy (discrete)
u	dependent variable, time series, or grid function
\hat{u}	Laplace transform of time series, dependent variable or grid function
\tilde{u}	spatial Fourier transform of dependent variable or grid function
$u^{(+)}, u^{(-)}$	traveling wave components
u_d	discrete time series or grid function
U	modal function
\mathbb{U}	spatial unit interval $[0,1]$
\mathbb{U}_N	discrete N -point spatial unit interval $[0, \dots, N]$
v_g	group velocity
v_ϕ	phase velocity
\mathbf{v}	potential energy (discrete)
\mathcal{V}	potential energy (continuous)
\mathbf{w}	discrete state update vector
x	spatial independent variable (non-dimensionalized from Section 6.1.2 onwards)
z	complex frequency variable
\mathbb{Z}	set of all integers $[-\infty, \dots, \infty]$
\mathbb{Z}^+	set of non-negative integers $[0, \dots, \infty]$
α	dimensionless string parameter ($= T_0/EA$)
β	wavenumber
γ	wave speed (Hz, spatially non dimensionalized)
δ_{d+}	forward difference operator, in coordinate direction d , $= (e_{d+} - 1)/k$
δ_{d-}	backward difference operator, in coordinate direction d , $= (1 - e_{d-})/k$
δ_d	centered difference operator, in coordinate direction d , $= (e_{d+} - e_{d-})/2k$
δ_{dd}	second difference operator, in coordinate direction d , $= \delta_{d+}\delta_{d-}$
δ_{dddd}	fourth difference operator, in coordinate direction d , $= \delta_{dd}\delta_{dd}$
δ_Δ	Laplacian difference operator
Δ	Laplacian operator
$\eta_{(1)}, \eta_{(2)}$	transverse displacements
θ	angle (rad.)
θ	free parameter, implicit finite difference scheme
κ	stiffness parameter, bars, plates
λ	Courant number, for the wave equation ($= k\gamma/h$)
μ	difference scheme free parameter
μ_{d+}	forward averaging operator, in coordinate direction d , $= (e_{d+} + 1)/2$
μ_{d-}	backward averaging operator, in coordinate direction d , $= (1 + e_{d-})/2$
μ_d	centered averaging operator, in coordinate direction d , $= (e_{d+} + e_{d-})/2$

ν	Poisson's ratio
ρ_l	linear mass density (kg/m)
ρ	material density (kg/m ³)
σ	loss (Hz)
σ_0	frequency-independent loss parameter (Hz)
σ_1	frequency-dependent loss parameter (Hz)
τ	free parameter, implicit scheme
ϕ	phase of sinusoid (rad.)
ω	angular frequency (rad./sec.)
ω_0	fundamental or oscillator angular frequency (rad./sec.)
1	one, or identity operation

Bibliography

- [1] J. Abel, D. Berners, S. Costello, and J. O. Smith III. Digital delay networks for designing artificial reverberators, 2006. In the Proceedings of the 121st Audio Engineering Society Convention. Preprint 6954.
- [2] J.-M. Adrien. The missing link: Modal synthesis. In G. DePoli, A. Picialli, and C. Roads, editors, *Representations of Musical Signals*, pages 269–297. MIT Press, Cambridge, MA, 1991.
- [3] W. Ames. *Numerical Methods for Partial Differential Equations*. Thomas Nelson and Sons, London, 1969.
- [4] G. Anand. Large-amplitude damped free vibration of a stretched string. *Journal of the Acoustical Society of America*, 45(5):1089–1096, 1969.
- [5] S. Antman. *Nonlinear Problems of Elasticity*. Springer-Verlag, New York, 1995.
- [6] M. Aramaki. *Analyse-synthese de sons impulsifs: approches physiques et perceptives*. PhD thesis, Universite de la Mediterranée-Aix Marseille II, 2002.
- [7] M. Aramaki and R. Kronland-Martinet. Analysis-synthesis of impact sounds by real-time dynamic filtering. *IEEE Transactions on Audio Speech and Language Processing*, 14(2):695–705, 2006.
- [8] D. Arfib. Digital synthesis of complex spectra by beams of multiplication of nonlinear distorted sine waves. *Journal of the Audio Engineering Society*, 27(10):757–768, 1979.
- [9] F. Avanzini and D. Rocchesso. Efficiency, accuracy, and stability issues in discrete time simulations of single reed instruments. *Journal of the Acoustical Society of America*, 111(5):2293–2301, 2002.
- [10] R. Bacon and J. Bowsher. A discrete model of a struck string. *Acustica*, 41:21–7, 1978.
- [11] R. Bader. *Computational Mechanics of the Classical Guitar*. Springer, 2005.
- [12] B. Bank and L. Sujbert. Modeling the longitudinal vibration of piano strings. In *Proceedings of the Stockholm Musical Acoustics Conference*, pages 143–146, Stockholm, Sweden, August 2003.
- [13] B. Bank and L. Sujbert. A piano model including longitudinal string vibration. In *Proceedings of the International Digital Audio Effects Conference*, pages 89–94, Naples, Italy, October 2004.

- [14] B. Bank and L. Sujbert. Generation of longitudinal vibrations in piano strings: From physics to sound synthesis. *Journal of the Acoustical Society of America*, 117(4):539–557, 2005.
- [15] B. Bank and L. Sujbert. Generation of longitudinal vibrations in piano strings: From physics to sound synthesis. *Journal of the Acoustical Society of America*, 117(4):2268–2278, 2005.
- [16] M. Beeson and D. Murphy. Roomweaver: A digital waveguide mesh based room acoustics research tool. In *Proceedings of the Digital Audio Effects Conference, ADDRESS =*.
- [17] V. Belevitch. Summary of the history of circuit theory. *Proceedings of the IRE*, 50:848–855, May 1962.
- [18] R. Benamar and M. Bennouna. The effects of large vibration amplitudes on the mode shapes and natural frequencies of thin elastic structures, part II: Fully clamped rectangular isotropic plates. *Journal of Sound and Vibration*, 164(2):295–316, 1993.
- [19] J. Bensa, S. Bilbao, R. Kronland-Martinet, and J. O. Smith III. The simulation of piano string vibration: From physical models to finite difference schemes and digital waveguides. *Journal of the Acoustical Society of America*, 114(2):1095–1107, 2003.
- [20] J. Bensa, S. Bilbao, R. Kronland-Martinet, J.O. Smith III, and T. Voinier. Computational modeling of stiff piano strings using digital waveguides and finite differences. *Acustica*, 91:289–298, 2005.
- [21] H. Berger. A new approach to the analysis of large deflections of plates. *Journal of Applied Mathematics*, 22:465–472, 1955.
- [22] D. Berners. *Acoustics and Signal Processing Techniques for Physical Modelling of Brass Instruments*. PhD thesis, Department of Electrical Engineering, Stanford University, 1999.
- [23] S. Bilbao. Parameterized families of finite difference schemes for the wave equation. *Numerical Methods for Partial Differential Equations*, 20(3):463–480, 2004.
- [24] S. Bilbao. *Wave and Scattering Methods for Numerical Simulation*. John Wiley and Sons, Chichester, UK, 2004.
- [25] S. Bilbao. Conservative numerical methods for nonlinear strings. *Journal of the Acoustical Society of America*, 118(5):3316–3327, 2005.
- [26] S. Bilbao. Energy-conserving numerical methods for nonlinear plates of Berger type, 2006. Under review, *Acustica united with Acta Acustica*.
- [27] S. Bilbao. A family of conservative finite difference schemes for the dynamical von Karman plate equations, 2006. Under review, *Computer Methods in Applied Mechanics and Engineering*.
- [28] S. Bilbao. Fast modal synthesis by digital waveguide extraction. *IEEE Signal Processing Letters*, Jan 2006.
- [29] S. Bilbao. A digital plate reverberation algorithm. *Journal of the Audio Engineering Society*, March 2007.

- [30] S. Bilbao, K. Arcas, and A. Chaigne. A physical model of plate reverberation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, 2006.
- [31] S. Bilbao, L. Savioja, and J. O. Smith III. 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.
- [32] S. Bilbao and J. O. Smith III. Finite difference schemes for the wave equation: Stability, passivity and numerical dispersion. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pages 255–266, May 2003.
- [33] S. Bilbao and J. O. Smith III. Energy-conserving finite difference schemes for nonlinear strings. *Acustica*, 91:299–311, 2005.
- [34] S. Bilbao and J. O. Smith III. Energy conserving finite difference schemes for nonlinear strings. *Acustica*, 91:299–311, 2005.
- [35] I. Bisnovatyi. Flexible software framework for modal synthesis. In *Proceedings of the Digital Audio Effects Conference*, Verona, Italy, December 2000.
- [36] G. Borin, G. DePoli, and A. Sarti. Algorithms and structures for synthesis using physical models. *Computer Music Journal*, 16(4):30–42, 1992.
- [37] G. Borin, G. DePoli, and A. Sarti. Musical signal synthesis. In C. Roads, S. Pope, A. Piccialli, and G. DePoli, editors, *Musical Signal Processing*, pages 5–30. Swets and Zeitlinger, Lisse, The Netherlands, 1997.
- [38] G. Borin, G. De Poli, and D. Rochesso. Elimination of delay-free loops in discrete-time models of nonlinear acoustic systems. *IEEE Transactions on Speech and Audio Processing*, 8:597–606, 2000.
- [39] R. Boulanger, editor. *The csound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing, and Programming*. MIT Press, Cambridge, Massachusetts, 2001.
- [40] X. Boutillon. Model for piano hammers: Experimental determination and digital simulation. *Journal of the Acoustical Society of America*, 83(2):746–754, 1988.
- [41] T. Bridges and S. Reich. Numerical methods for Hamiltonian PDEs. *Journal of Physics A: Mathematical and General*, 39:5287–5320, 2006.
- [42] A. Bruckstein and T. Kailath. An inverse scattering framework for several problems in signal processing. *IEEE ASSP Magazine*, 4(1):6–20, January 1987.
- [43] C. Bruyns. Modal stnthesis for arbitrarily shaped objects. *Computer Music Journal*, pages 22–37, 2006.
- [44] C. Cadoz, A. Luciani, and J.-L. Florens. Responsive input devices and sound synthesis by simulation of instrumental mechanisms. *Computer Music Journal*, 8(3):60–73, 1983.
- [45] C. Cadoz, A. Luciani, and J.-L. Florens. Cordis-anima: A modeling and simulation system for sound and image synthesis. *Computer Music Journal*, 17(1):19–29, 1993.

- [46] J. Cage and D. Charles. *For The Birds: John Cage in Conversation with Daniel Charles*. Marion Boyers, 1981.
- [47] M. Campbell and C. Greated. *The Musician's Guide to Acoustics*. Oxford University Press, 1994.
- [48] C. Canuto, M. Hussaini, A. Quarteroni, and T. Zang. *Spectral Methods in Fluid Dynamics*. Springer, 1988.
- [49] G. F. Carrier. On the nonlinear vibration problem of the elastic string. *Quarterly of Applied Mathematics*, 3:157–165, 1945.
- [50] S. Cavaliere and A. Piccialli. Granular synthesis of musical signals. In C. Roads, S. Pope, A. Piccialli, and G. DePoli, editors, *Musical Signal Processing*, pages 155–186. Swets and Zeitlinger, Lisse, The Netherlands, 1997.
- [51] A. Chaigne. On the use of finite differences for musical synthesis. Application to plucked stringed instruments. *Journal d'Acoustique*, 5(2):181–211, 1992.
- [52] A. Chaigne and A. Askenfelt. Numerical simulations of struck strings. I. A physical model for a struck string using finite difference methods. *Journal of the Acoustical Society of America*, 95(2):1112–1118, February 1994.
- [53] A. Chaigne and A. Askenfelt. Numerical simulations of struck strings. II. Comparisons with measurements and systematic exploration of some hammer-string parameters. *Journal of the Acoustical Society of America*, 95(3):1631–40, Mar. 1994.
- [54] A. Chaigne and C. Lambourg. Time-domain simulation of damped impacted plates. I Theory and experiments. *Journal of the Acoustical Society of America*, 109(4):1422–1432, 2001.
- [55] D. Cheng. *Field and Wave Electromagnetics*. Addison-Wesley, USA, second edition, 1990.
- [56] J. Chowning. The synthesis of complex audio spectra by means of frequency modulation. *Journal of the Audio Engineering Society*, 21(7), 1973.
- [57] C. Christopoulos. *The Transmission-Line Modelling Method*. IEEE Press, New York, New York, USA, 1995.
- [58] G. Cohen and P. Joly. Construction and analysis of fourth-order finite difference schemes for the acoustic wave equation in non-homogeneous media. *SIAM Journal of Numerical Analysis*, 33(4):1266–1302, 1996.
- [59] H. Conklin. Design and tone in the mechanoacoustic piano. part III. piano strings and scale design. *Journal of the Acoustical Society of America*, 100(3):1286–1298, 1996.
- [60] 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, 1990.
- [61] P. Cook. Tbone: An interactive waveguide brass instrument synthesis workbench for the next machine. In *Proceedings of the International Computer Music Conference*, pages 297–299, Montreal, Canada, 1991.

- [62] P. Cook. Spasm: A real-time vocal tract physical model editor/controller and singer: the companion software synthesis system. *Computer Music Journal*, 17(1):30–44, 1992.
- [63] P. Cook. Physically informed sonic modelling (PhISM): Synthesis of percussive sounds. *Computer Music Journal*, 21(3):38–49, 1997.
- [64] P. Cook. *Real Sound Synthesis for Interactive Applications*. A. K. Peters, 2002.
- [65] P. Cook and G. Scavone. The synthesis tool kit. In *Proceedings of the International Computer Music Conference*, Beijing, China, 1999.
- [66] R. Cook, editor. *Concepts and applications of finite element analysis*. Wiley, New York, fourth edition, 2002.
- [67] J. Cooley and J. Tukey. An algorithm for the machine computation of complex Fourier series. *Mathematical Computation*, 19:297–301, 1965.
- [68] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, 100:32–74, 1928.
- [69] M. Dablain. The application of high-order differencing to the scalar wave equation. *Geophysics*, 51(1):54–66, 1986.
- [70] G. Dahlquist. A special stability problem for linear multistep methods. *BIT*, 3:27–43, 1963.
- [71] P. Depalle and S. Tassart. State space sound synthesis and a state space synthesiser builder. In *Proceedings of the International Computer Music Conference*, pages 88–95, Banff, 1995.
- [72] G. DePoli, A. Piccilli, and C. Roads, editors. *Representations of Musical Signals*. MIT Press, Cambridge, MA, 1991.
- [73] R. Dickey. Infinite systems of nonlinear oscillation equations related to the string. *Proceedings of the American Mathematics Society*, 23(3):459–468, 1969.
- [74] R. Dickey. Stability of periodic solutions of the nonlinear string. *Quarterly of Applied Mathematics*, 38:253–259, 1980.
- [75] C. Dodge and T. Jerse. *Computer Music: Synthesis, Composition and Performance*. Schirmer Books, New York, 1985.
- [76] M. Dolson. The phase vocoder: A tutorial. *Computer Music Journal*, 10(4):14–27, 1986.
- [77] V. Doutaut, D. Matignon, and A. Chaigne. Numerical simulations of xylophones. II. Time domain modeling of the resonator and of the radiated sound pressure. *Journal of the Acoustical Society of America*, 104(3):1633–1647, 1998.
- [78] E. Ducasse. A physical model of a single reed wind instrument, including actions of the player. *Computer Music Journal*, 27(1):59–70, 2003.
- [79] G. Eckel, F. Iovino, and R. Caussé. Sound synthesis by physical modelling with modalys. In *Proceedings of the International Symposium on Musical Acoustics*, pages 479–482, Dourdan, France, 1995.

- [80] G. Efstathiades. A new approach to the large-deflection vibrations of imperfect circular disks using Galerkin's procedure. *Journal of Sound and Vibration*, 16(2):231–253, 1971.
- [81] C. Erkut. *Aspects in Analysis and Model-Based Sound Synthesis of Plucked String Instruments*. PhD thesis, Laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology, 2002.
- [82] C. Erkut, M. Karjalainen, P. Huang, and V. Vlimki. Acoustical analysis and model-based sound synthesis of the kantele. *Journal of the Acoustical Society of America*, 112(4):1681–1691, 2002.
- [83] G. Essl, S. Serafin, P. Cook, and J. O. Smith III. Musical applications of banded waveguides. *Computer Music Journal*, 28(1):51–63, 2004.
- [84] G. Essl, S. Serafin, P. Cook, and J. O. Smith III. Theory of banded waveguides. *Computer Music Journal*, 28(1):37–50, 2004.
- [85] G. Evangelista. Wavelet representations of musical signals. In C. Roads, S. Pope, A. Piccialli, and G. DePoli, editors, *Musical Signal Processing*, pages 127–153. Swets and Zeitlinger, Lisse, The Netherlands, 1997.
- [86] G. Evans, J. Blackledge, and P. Yardley. *Numerical Methods for Partial Differential Equations*. Springer, London, 1999.
- [87] A. Fettweis. Digital filters related to classical structures. *AEU: Archive für Elektronik und Übertragungstechnik*, 25:79–89, February 1971. (See also U.S. Patent 3,967,099, 1976, now expired.)
- [88] A. Fettweis. Wave digital filters: Theory and practice. *Proceedings of the IEEE*, 74(2):270–327, February 1986.
- [89] A. Fettweis. Discrete passive modelling of physical systems described by PDEs. In *Proceedings of EUSIPCO-92, Sixth European Signal Processing Conference*, volume 1, pages 55–62, Brussels, Belgium, August 1992.
- [90] A. Fettweis. Discrete passive modelling of viscous fluids. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages 1640–1643, San Diego, California, USA, May 1992.
- [91] A. Fettweis and K. Meerkötter. On adaptors for wave digital filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-23(6):516–525, December 1975.
- [92] A. Fettweis and G. Nitsche. Transformation approach to numerically integrating PDEs by means of WDF principles. *Multidimensional Systems and Signal Processing*, 2(2):127–159, May 1991.
- [93] H. D. Fischer. Wave digital filters for numerical integration. *ntz-Archiv*, 6:37–40, February 1984.
- [94] J. Flanagan and R. Golden. The phase vocoder. *Bell System Technical Journal*, 45:1493–1509, 1966.

- [95] H. Fletcher. Normal vibration frequencies of a stiff string. *Journal of the Acoustical Society of America*, 36(1):203–209, 1964.
- [96] N. Fletcher and T. Rossing. *The Physics of Musical Instruments*. Springer-Verlag, New York, New York, USA, 1991.
- [97] J.-L. Florens and C. Cadoz. The physical model: Modeling and simulating the instrument universe. In G. DePoli, A. Picialli, and C. Roads, editors, *Representations of Musical Signals*, pages 227–268. MIT Press, Cambridge, MA, 1991.
- [98] F. Fontana and D. Rocchesso. Physical modelling of membranes for percussive instruments. *Acustica United with Acta Acustica*, 84:529–542, May/June 1998.
- [99] F. Fontana and D. Rocchesso. Signal-theoretic characterization of waveguide mesh geometries for models of two-dimensional wave propagation in elastic media. *IEEE Transactions on Speech and Audio Processing*, 9(2):152–61, Feb. 2001.
- [100] B. Fornberg. *A Practical Guide to Pseudospectral Methods*. Cambridge Monographs on Applied and Computational Mathematics, Cambridge, England, 1995.
- [101] F. Friedlander. On the oscillations of the bowed string. *Proceedings of the Cambridge Philosophical Society*, 49:516, 1953.
- [102] D. Furihata. Finite difference schemes for nonlinear wave equation that inherit energy-conservation property. *Journal of Computational and Applied Mathematics*, 134(1–2):37–57, 2001.
- [103] P. Garabedian. *Partial Differential Equations*. Chelsea Publishing Company, New York, New York, USA, second edition, 1986.
- [104] M. Garber. Master’s thesis, Massachusetts Institute of Technology, 1982.
- [105] Y. Genin. An algebraic approach to A-stable linear multistep-multiderivative integration formulas. *BIT*, 14(4):382–406, 1974.
- [106] N. Giordano. Simple model of a piano soundboard. *Journal of the Acoustical Society of America*, 102(2):1159–1168, 1997.
- [107] N. Giordano. Sound production by a vibrating piano soundboard: Experiment. *Journal of the Acoustical Society of America*, 104(3):1648–1653, 1998.
- [108] N. Giordano and M. Jiang. Physical modeling of the piano. *Eurasip Journal of Applied Signal Processing*, pages 926–33, 2004.
- [109] D. Gottlieb and S. Orszag. *Numerical Analysis of Spectral Methods*. SIAM–CBMS, Philadelphia, 1977.
- [110] C. Gough. The nonlinear free vibration of a damped elastic string. *Journal of the Acoustical Society of America*, 75(6):1770–1776, 1984.
- [111] K. Graff. *Wave Motion in Elastic Solids*. Dover, New York, New York, USA, 1975.

- [112] D. Greenspan. Conservative numerical methods for $\ddot{x} = f(x)$. *J. Comp. Phys.*, 56:28–41, 1984.
- [113] B. Gustaffson, H.-O. Kreiss, and J. Olinger. *Time Dependent Problems and Difference Methods*. John Wiley and Sons, New York, New York, USA, 1995.
- [114] B. Gustaffson, H.-O. Kreiss, and A. Sundstrom. Stability theory of difference approximations for mixed initial boundary value problems. II. *Mathematics of Computation*, 26(119):649–686, 1972.
- [115] D. Hall. Numerical simulations of xylophones. I. Time domain modeling of vibrating bars. *Journal of the Acoustical Society of America*, 81(2):535–546, 1987.
- [116] D. Hall. Piano string excitation II: General solution for a hard narrow hammer. *Journal of the Acoustical Society of America*, 101(1):539–557, 1997.
- [117] T. Hélie. On the use of Volterra series for real-time simulations of weakly nonlinear analog audio devices: Application to the Moog ladder filter. In *Proceedings of the International Digital Audio Effects Conference*, pages 7–12, Montreal, Canada, September 2006.
- [118] H. Hilber, T. Hughes, and R. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering and Structural Dynamics*, 5:283–292, 1977.
- [119] L. Hiller and P. Ruiz. Synthesizing musical sounds by solving the wave equation for vibrating objects: Part I. *Journal of the Audio Engineering Society*, 19(6):462–470, 1971.
- [120] L. Hiller and P. Ruiz. Synthesizing musical sounds by solving the wave equation for vibrating objects: Part II. *Journal of the Audio Engineering Society*, 19(7):542–550, 1971.
- [121] M. Hirsch and S. Smale. *Differential Equations, Dynamical Systems and Linear Algebra*. Academic Press, New York, 1974.
- [122] A. Hirschberg, J. Kergomard, and G. Weinreich, editors. *Mechanics of Musical Instruments*. Springer, New York, 1995.
- [123] W. J. R. Hoefer. *The Electromagnetic Wave Simulator*. John Wiley and Sons, Chichester, England, 1991.
- [124] D. Jaffe and J. O. Smith III. Extensions of the Karplus-Strong plucked string algorithm. *Computer Music Journal*, 7(2):56–68, Summer 1983.
- [125] H. Järveläinen, T. Verma, and V. Välimäki. Perception and adjustment of pitch in inharmonic string instrument tones. 31(4):311–319, 2002.
- [126] W. Jin and L. Ruxun. A new approach to design high-order schemes. *Journal of Computational and Applied Mathematics*, 134:59–67, 2001.
- [127] P. Johns and R. Beurle. Numerical solution of 2-dimensional scattering problems using a transmission-line matrix. *Proceedings of the IEE*, 118:1203–1208, September 1971.
- [128] J. Johnson and A. Bajaj. Amplitude modulated and chaotic dynamics in resonant motion of strings. *Journal of Sound and Vibration*, 128(1):87–107, 1989.

- [129] J.-M. Jot and A. Chaigne. Spring reverb emulation using dispersive allpass filters in a waveguide structure, 1991. In the Proceedings of the 90th Audio Engineering Society Convention. Preprint 3030.
- [130] W. Kaegi and S. Tempelaars. VOSIM—a new sound synthesis system. *Journal of the Audio Engineering Society*, 26(6):418–426, 1978.
- [131] T. Kailath. *Linear Systems*. Prentice Hall, 1980.
- [132] T. Kailath and A. Sayed. Displacement structure: Theory and applications. *SIAM Review*, 37(3):297–386, 1995.
- [133] M. Karjalainen. Block-compiler: Efficient simulation of acoustic and audio systems. In *Preprints of AES 114th Convention*, Amsterdam, the Netherlands, May 2003.
- [134] M. Karjalainen. Time-domain physical modelling and real-time synthesis using mixed modelling paradigms. In *Proceedings of the Stockholm Musical Acoustics Conference*, volume 1, pages 393–396, Stockholm, Sweden, August 2003.
- [135] M. Karjalainen and C. Erkut. Digital waveguides vs. finite difference schemes: Equivalence and mixed modeling. *EURASIP Journal on Applied Signal Processing*, pages 978–989, June 2004.
- [136] M. Karjalainen, V. Välimäki, and Z. Janosy. Towards high-quality sound synthesis of the guitar and string instruments. In *Proceedings of the International Computer Music Conference*, pages 56–63, Tokyo, Japan, 1993.
- [137] M. Karjalainen, V. Välimäki, and T. Tolonen. Plucked-string synthesis: From the karplus-strong algorithm to digital waveguides and beyond. *Computer Music Journal*, 22(3):17–32, 1998.
- [138] K. Karplus and A. Strong. Digital synthesis of plucked-string and drum timbres. *Computer Music Journal*, 7(2):43–55, Summer 1983.
- [139] W. Kausel. *A Musical Acoustician's Guide to Computational Physics*. Institut für Wiener Klangstil, Vienna, 2003.
- [140] D. Keefe. Physical modeling of wind instruments. *Computer Music Journal*, 16(4):57–73, 1992.
- [141] J. L. Kelly and C. C. Lochbaum. Speech synthesis. In *Proceedings of the Fourth International Congress on Acoustics*, pages 1–4, Copenhagen, Denmark, 1962. Paper G42.
- [142] J. Kergomard. *Elementary Considerations on Reed-instrument Oscillations*. Springer, New York, New York, USA, 1995.
- [143] J.W. Kim and D. J. Lee. Optimized compact finite difference schemes with maximum resolution. *AIAA Journal*, 34(5):887–92, 1996.
- [144] R. Kirby and Z. Yosibash. Dynamic response of various von Karman non-linear plate models and their 3-D counterparts. *International Journal of Solids and Structures*, 193(6–8):575–599, 2004.

- [145] R. Kirby and Z. Yosibash. Solution of von Karman dynamic non-linear plate equations using a pseudo-spectral method. *Computer Methods in Applied Mechanics and Engineering*, 193(6–8):575–599, 2004.
- [146] G. Kirchhoff. *Vorlesungen über Mechanik*. Tauber, Leipzig, 1883.
- [147] H.-O. Kreiss. Initial boundary value problems for hyperbolic systems. *Communications on Pure and Applied Mathematics*, 23:277–298, 1970.
- [148] E. Kreyszig. *Introductory Functional Analysis with Applications*. John Wiley and Sons, New York, 1978.
- [149] G. Kron. Equivalent circuit of the field equations of Maxwell. *Proceedings of the IRE*, 32(5):284–299, May 1944.
- [150] W. Kuhl. The acoustical end technological properties of the reverberation plate. *E. B. U. Review*, A(49), May 1958.
- [151] E. Kurmyshev. Transverse and longitudinal mode coupling in a free vibrating soft string. *Physics Letters A*, 310(2–3):148–160, 2003.
- [152] M. Kurz and B. Feiten. Physical modeling of a stiff string by numerical integration. In *Proceedings of the International Computer Music Conference*, pages 361–364, Hong Kong, 1996.
- [153] T. Laakso, V. Välimäki, M. Karjalainen, and U. Laine. Splitting the unit delay—tools for fractional delay filter design. *IEEE Signal Processing Magazine*, 13(1):30–60, January 1996.
- [154] M. Laurson, C. Erkut, V. Välimäki, and M. Kuuskankare. Methods for modeling realistic playing in acoustic guitar synthesis. *Computer Music Journal*, 25(3):38–49, 2001.
- [155] M. LeBrun. Digital waveshaping synthesis. *Journal of the Audio Engineering Society*, 27(4):250–266, 1979.
- [156] K. Legge and N. Fletcher. Nonlinear generation of missing modes on a vibrating string. *Journal of the Acoustical Society of America*, 76(1):5–12, 1984.
- [157] S. K. Lele. Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics*, 103:16–42, 1992.
- [158] A. Leung and S. Mao. A symplectic Galerkin method for nonlinear vibration of beams and plates. *Journal of Sound and Vibration*, 183(3):475–491, 1995.
- [159] S. Li and L. Vu-Quoc. Finite difference calculus invariant structure of a class of algorithms for the nonlinear Klein Gordon equation. *SIAM Journal of Numerical Analysis*, 32:1839–1875, 1995.
- [160] I. Liu and M. Rincon. Effect of moving boundaries on the vibrating elastic string. *Applied Numerical Mathematics*, 47:159–172, 2003.
- [161] D. P. Lockard, K. S. Brentner, and H. L. Atkins. High-accuracy algorithms for computational aeroacoustics. *AIAA Journal*, 33(2):247–51, 1995.

- [162] A. Lyapunov. *Stability of Motion*. Academic Press, New York, 1966.
- [163] P. Manning. *Electronic and Computer Music*. Clarendon Press, Oxford, 1985.
- [164] J. D. Markel and A. H. Gray, Jr. *Linear Prediction of Speech Signals*. Springer-Verlag, New York, New York, USA, 1976.
- [165] R. McAulay and T. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34(4):744–754, 1986.
- [166] B. McCartin. An alternative analysis of Duffing’s equation. *SIAM Review*, 34(3):482–491, 1992.
- [167] J. McCartney. Supercollider: A new real-time sound synthesis language. In *Proceedings of the International Computer Music Conference*, pages 257–258, Hong Kong, 1996.
- [168] M. McIntyre, R. Schumacher, and J. Woodhouse. On the oscillations of musical instruments. *Journal of the Acoustical Society of America*, 74(5):1325–1345, 1983.
- [169] M. McIntyre and J. Woodhouse. On the fundamentals of bowed string dynamics. *Acustica*, 43(2):93–108, 1979.
- [170] R. Moore. *Elements of Computer Music*. Prentice Hall, 1990.
- [171] D. Morrill. Trumpet algorithms for computer composition. *Computer Music Journal*, 1(1):46–52, 1977.
- [172] D. Morrison and J.-M. Adrien. Mosaic: A framework for modal synthesis. *Computer Music Journal*, 17(1):45–56, 1993.
- [173] P. Morse. *Vibration and Sound*. Acoustical Society of America, 1936.
- [174] P. Morse and U. Ingard. *Theoretical Acoustics*. Princeton University Press, Princeton, New Jersey, 1968.
- [175] R. Msallam, S. Dequidt, S. Tassart, and R. Caussé. Physical model of the trombone including nonlinear propagation effects. application to the sound synthesis of loud tones. *Acta Acustica United with Acustica*, 86:725–736, 2000.
- [176] D. Murphy, A. Kelloniemi, J. Mullen, and S. Shelley. Acoustic modelling using the digital waveguide mesh. *IEEE Signal Processing Magazine*, 24(2):55–66, 2007.
- [177] R. Narasimha. Nonlinear vibration of an elastic string. *J. Sound Vib.*, 8:134–146, 1968.
- [178] A. Nayfeh and D. Mook. *Nonlinear Oscillations*. John Wiley and Sons, New York, 1979.
- [179] N. Newmark. A method of computation for structural dynamics. *ACSE Journal of the Engineering Mechanics Division*, 85:67–94, 1959.
- [180] A. Noor and W. Pilkey. *State-of-the-Art Surveys on Finite Element Technology*. ASME, New York, 1983.

- [181] B. J. Noye and J. Rankovic. An accurate explicit finite difference technique for solving the one-dimensional wave equation. *Communications in Applied Numerical Methods*, 2:557–61, 1986.
- [182] B. J. Noye and J. Rankovic. An accurate five-point implicit finite difference method for solving the one-dimensional wave equation. *Communications in Applied Numerical Methods*, 5:247–52, 1989.
- [183] A. V. Oppenheim and R. Schaffer. *Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1975.
- [184] O. O'Reilly and P. Holmes. Non-linear, non-planar and non-periodic vibrations of a string. *Journal of Sound and Vibration*, 153(3):413–435, 1992.
- [185] S. Osher. Stability of difference approximations of dissipative type for mixed initial boundary value problems. I. *Mathematics of Computation*, 23:335–340, 1969.
- [186] J. Pakarinen, V. Välimäki, and M. Karjalainen. Physics-based methods for modeling nonlinear vibrating strings. *Acustica United with Acta Acustica*, 91(2):312–325, 2005.
- [187] F. Pedersini, A. Sarti, S. Tubaro, and R. Zettoni. Towards the automatic synthesis of nonlinear wave digital models for musical acoustics. In *Proceedings of EUSIPCO-98, Ninth European Signal Processing Conference*, volume 4, pages 2361–2364, Rhodes, Greece, 1998.
- [188] H. Penttinen, C. Erkut, J. Plkki, V. Välimäki, and M. Karjalainen. Design and analysis of a modified kantele with increased loudness. *Acustica United with Acta Acustica*, 91(2):261–268, 2005.
- [189] S. Petrausch, J. Escolano, and R. Rabenstein. A general approach to block-based physical modeling. with mixed modeling strategies for digital sound synthesis. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005.
- [190] R. Pitteroff and J. Woodhouse. Mechanics of the contact area between a violin bow and a string. Part II: Simulating the bowed string. *Acustica United with Acta Acustica*, 84:744–757, 1998.
- [191] M. Portnoff. Implementation of the digital phase vocoder using the fast Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(3):243–248, 1976.
- [192] J. Proakis. *Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, New Jersey, USA, third edition, 1996.
- [193] M. Puckette. Pure data. In *Proceedings of the International Computer Music Conference*, pages 269–272, 1996.
- [194] M. Puckette. Theory and techniques of electronic music, 2006. Available online at <http://www-crca.ucsd.edu/msp/techniques.htm>.
- [195] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer, Berlin, 1997.

- [196] R. Rabenstein, S. Petrausch, A. Sarti, G. De Sanctis, C. Erkut, and M. Karjalainen. Block-based physical modeling for digital sound synthesis. *IEEE Signal Processing Magazine*, 24(2):42–54, 2007.
- [197] L. Rabiner and R. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1978.
- [198] M. Rath, F. Avanzini, and D. Rocchesso. Physically based real-time modeling of contact sounds. In *Proceedings of the International Computer Music Conference*, Goteborg, Sweden, 2002.
- [199] L. Rhaouti, A. Chaigne, and P. Joly. Time-domain modeling and numerical simulation of a kettledrum. *Journal of the Acoustical Society of America*, 105(6):3545–3562, 1999.
- [200] P. Ribeiro and M. Petyt. Geometrical non-linear, steady state forced periodic vibration of plates, part I: Model and convergence studies. *Journal of Sound and Vibration*, 226(5):955–983, 1999.
- [201] R. Richtmyer and K. Morton. *Difference Methods for Initial-Value Problems*. Krieger Publishing Company, Malabar, Florida, USA, second edition, 1994.
- [202] J.-C. Risset. Computer study of trumpet tones. Technical report, Bell Technical Laboratories, Murray Hill, New Jersey, 1966.
- [203] C. Roads. Granular synthesis of sound. In C. Roads and J. Strawn, editors, *Foundations of Computer Music*, pages 145–159. 1985.
- [204] C. Roads. A tutorial on nonlinear distortion or waveshaping synthesis. In C. Roads and J. Strawn, editors, *Foundations of Computer Music*, pages 83–94. 1985.
- [205] C. Roads. *The Computer Music Tutorial*. MIT Press, Cambridge, Massachusetts, 1996.
- [206] C. Roads, S. Pope, A. Piccialli, and G. DePoli, editors. *Musical Signal Processing*. Swets and Zeitlinger, Lisse, The Netherlands, 1997.
- [207] C. Roads and J. Strawn, editors. *Foundations of Computer Music*. MIT Press, 1985.
- [208] D. Rocchesso. The ball within the box: A sound processing metaphor. *Computer Music Journal*, 19(4):47–57, 1995.
- [209] D. Rocchesso. Maximally diffusive yet efficient feedback delay networks for artificial reverberation. *IEEE Signal Processing Letters*, 4(9):252–255, September 1997.
- [210] D. Rocchesso and F. Fontana, editors. *The Sounding Object*. Mondo Estremo, 2003. Available Online at <http://www.mondo-estremo.com/info/publications/public.html>.
- [211] D. Rocchesso and J. O. Smith III. Circulant and elliptic feedback delay networks for artificial reverberation. *IEEE Transactions on Speech and Audio Processing*, 5(1):51–63, January 1997.
- [212] X. Rodet. Time-domain formant-wave-function synthesis. *Computer Music Journal*, 8(3):9–14, 1980.

- [213] X. Rodet and C. Vergez. Nonlinear dynamics in physical models: From basic models to true musical-instrument models. *Computer Music Journal*, 23(3):35–49, 1999.
- [214] X. Rodet and C. Vergez. Nonlinear dynamics in physical models: Simple feedback loop systems and properties. *Computer Music Journal*, 23(3):18–34, 1999.
- [215] T. Rossing and N. Fletcher. Nonlinear vibrations in plates and gongs. *Journal of the Acoustical Society of America*, 73(1):345–351, 1983.
- [216] T. Rossing, F. Moore, and P. Wheeler. *The Science of Sound*. Addison Wesley, third edition, 2002.
- [217] D. Rowland. Parametric resonance and nonlinear string vibrations. *American Journal of Physics*, 72(6):758–765, 2004.
- [218] D. Roze and T. Hélie. Sound synthesis of nonlinear strings using Volterra series. In *International Congress on Acoustics*, Madrid, Spain, 2007. Available on CD-ROM.
- [219] M. Rubin and O. Gottlieb. Numerical solutions of forced vibration and whirling of a nonlinear string using the theory of a Cosserat point. *Journal of Sound and Vibration*, 197(1):85–101, 1996.
- [220] P. Ruiz. A technique for simulating the vibrations of strings with a digital computer. Master’s thesis, University of Illinois, 1969.
- [221] J. Sanz-Serna. An explicit finite-difference scheme with exact conservation properties. *Journal of Computational Physics*, 47:199–210, 1982.
- [222] J. Sanz-Serna. Symplectic integrators for hamiltonian problems: An overview. *Acta Numerica*, 1:243–286, 1991.
- [223] A. Sarti and G. DePoli. Toward nonlinear wave digital filters. *IEEE Transactions on Signal Processing*, pages 1654–1668, 1999.
- [224] L. Savioja and V. Välimäki. Reduction of the dispersion error in the interpolated digital waveguide mesh using frequency warping. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 973–976, New York, March 1999.
- [225] L. Savioja and V. Välimäki. Reducing the dispersion error in the digital waveguide mesh using interpolation and frequency-warping techniques. *IEEE Transactions on Speech and Audio Processing*, 8(2):184–194, March 2000.
- [226] G. Scavone. *An Acoustic Analysis of Single-Reed Woodwind Instruments with an Emphasis on Design and Performance Issues and Digital Waveguide Techniques*. PhD thesis, Department of Music, Stanford University, 1997.
- [227] S. Schedin, C. Lambourg, and A. Chaigne. Transient sound fields from impacted plates: Comparison between numerical simulations and experiments. *Journal of Sound and Vibration*, 221(32):471–490, 1999.
- [228] W. Schottstaedt. The simulation of natural instrument tones using frequency modulation with a complex modulating wave. *Computer Music Journal*, 1(4):46–50, 1977.

- [229] S. Serafin, F. Avanzini, and D. Rocchesso. Bowed string simulation using an elasto-plastic friction model. In *Proceedings of the Stockholm Musical Acoustics Conference*, volume 1, pages 95–98, Stockholm, Sweden, August 2003.
- [230] X. Serra and J. O. Smith III. Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Computer Music Journal*, 14(4):12–24, 1990.
- [231] G. R. Shubin and J. B. Bell. A modified equation approach to constructing fourth order methods for acoustic wave propagation. *SIAM Journal of Scientific and Statistical Computing*, 8:135–51, 1987.
- [232] J. Simo, N. Tarnow, and K. Wong. Exact energy-momentum conserving algorithms for symplectic schemes for nonlinear dynamics. *Computer Methods in Applied Mechanics and Engineering*, 100:63–116, 1992.
- [233] G. Smith. *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. Clarendon Press, Oxford, third edition, 1985.
- [234] J. O. Smith III. A new approach to digital reverberation using closed waveguide networks. In *Proceedings of the International Computer Music Conference*, Vancouver, Canada, September 1985. Appears in Technical Report STAN-M-39, pp. 1–7, Center for Computer Research in Music and Acoustics (CCRMA), Department of Music, Stanford University.
- [235] J. O. Smith III. Efficient simulation of the reed-bore and bow-string mechanisms. In *Proceedings of the International Computer Music Conference*, pages 275–280, The Hague, The Netherlands, 1986.
- [236] J. O. Smith III. Viewpoints on the history of digital synthesis. In *Proceedings of the International Computer Music Conference*, pages 1–10, Montreal, Canada, October 1991.
- [237] J. O. Smith III. Waveguide simulation of non-cylindrical acoustic tubes. In *Proceedings of the International Computer Music Conference*, pages 304–307, Montreal, Canada, 1991.
- [238] J. O. Smith III. Physical modelling using digital waveguides. *Computer Music Journal*, 16(4):74–91, 1992.
- [239] J. O. Smith III. Efficient synthesis of stringed musical instruments. In *Proceedings of the International Computer Music Conference*, pages 64–71, Tokyo, Japan, 1993.
- [240] J. O. Smith III. Acoustic modeling using digital waveguides. In C. Roads, S. Pope, A. Piccialli, and G. DePoli, editors, *Musical Signal Processing*, pages 221–263. Swets and Zeitlinger, Lisse, The Netherlands, 1997.
- [241] J. O. Smith III. Digital waveguides vs. finite difference schemes: Equivalence and mixed modeling, 2004. <http://arxiv.org/abs/physics/0407032/>.
- [242] J. O. Smith III. *Physical Audio Signal Processing*. draft version, Stanford, CA, 2004. Available online at <http://ccrma.stanford.edu/~jos/pasp04/>.
- [243] J. O. Smith III and P. Cook. The second-order digital waveguide oscillator. In *Proceedings of the International Computer Music Conference*, San Jose, California, September 1992.

- [244] J. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Wadsworth and Brooks/Cole Advanced Books and Software, Pacific Grove, California, USA, 1989.
- [245] H. Strube. Time-varying wave digital filters and vocal-tract models. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 923–926, Paris, France, May 1982.
- [246] A. Stulov. Hysteretic model of the grand piano hammer felt. *Journal of the Acoustical Society of America*, 97:2577–85, 1995.
- [247] R. Szilard. *Theory and Analysis of Plates*. Prentice Hall, Englewood Cliffs, New Jersey, 1974.
- [248] N. Szilas and C. Cadoz. Analysis techniques for physical modeling networks. *Computer Music Journal*, 22(3):33–48, 1998.
- [249] A. Taflove. *Computational Electrodynamics*. Artech House, Boston, Massachusetts, USA, 1995.
- [250] A. Taflove. *Advances in Computational Electrodynamics*. Artech House, Boston, Massachusetts, USA, 1998.
- [251] L. H. Thomas. Elliptic problems in linear difference equations over a network. Technical report, Watson Scientific Computing Laboratory Report, Columbia University, New York, 1949.
- [252] O. Thomas, C. Touze, and A. Chaigne. Asymmetric nonlinear forced vibrations of free-edge circular plates. part i. experiments. *Journal of Sound and Vibration*, 265:1075–1101, 2003.
- [253] O. Thomas, C. Touze, 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*, 42:3339–3373, 2005.
- [254] T. Tolonen, V. Välimäki, and M. Karjalainen. Evaluation of modern sound synthesis methods. Technical Report 48, Laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology, March 1998.
- [255] T. Tolonen, V. Välimäki, and M. Karjalainen. Modeling of tension modulation nonlinearity in plucked strings. *IEEE Transactions on Speech and Audio Processing*, 8:300–310, May 2000.
- [256] C. Touze, O. Thomas, and A. Chaigne. Asymmetric nonlinear forced vibrations of free-edge circular plates. Part I. Theory. *Journal of Sound and Vibration*, 258(4):649–676, 2002.
- [257] L. Trautmann and R. Rabenstein. *Digital Sound Synthesis by Physical Modeling Using the Functional Transformation Method*. Kluwer Academic Publishers, 2003.
- [258] L. Trefethen. Group velocity in finite difference schemes. *SIAM Review*, 24:113–136, 1982.
- [259] L. Trefethen. Instability of finite difference models for hyperbolic initial boundary value problems. *Communications on Pure and Applied Mathematics*, 37:329–367, 1984.
- [260] L. Trefethen. *Spectral Methods in Matlab*. SIAM, Philadelphia, Pennsylvania, USA, 2000.
- [261] J. Tuomela. A note on high-order schemes for the one-dimensional wave equation. *BIT*, 36(1):158–65, 1995.

- [262] J. Tuomela. On the construction of arbitrary order schemes for the many-dimensional wave equation. *BIT*, 36(1):158–165, 1996.
- [263] A. Tveito and R. Winther. *Introduction to Partial Differential Equations*. Springer, New York, 1998.
- [264] V. Välimäki and M. Karjalainen. Digital waveguide modeling of wind instrument bores constructed of truncated cones. In *Proceedings of the International Computer Music Conference*, pages 423–430, Aarhus, Denmark, 1994.
- [265] V. Välimäki and M. Karjalainen. Implementation of fractional delay waveguide models using allpass filters. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 8–12, Detroit, Michigan, USA, May 1995.
- [266] V. Välimäki, M. Laurson, and C. Erkut. Commuted waveguide synthesis of the clavichord. *Computer Music Journal*, 27(1):71–82, 2003.
- [267] V. Välimäki, J. Pakarinen, C. Erkut, and M. Karjalainen. Discrete time modeling of musical instruments. *Reports on Progress in Physics*, 69:1–78, 2005.
- [268] V. Välimäki, H. Penttinen, J. Knif, M. Laurson, and C. Erkut. Sound synthesis of the harpsichord using a computationally efficient physical model. *EURASIP Journal on Applied Signal Processing*, 69(7):934–948, 2004.
- [269] V. Välimäki, T. Tolonen, and M. Karjalainen. Plucked-string synthesis algorithms with tension modulation nonlinearity. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 977–980, Phoenix, Arizona, USA, March 1999.
- [270] C. Vallette. The mechanics of vibrating strings. In A. Hirschberg, J. Kergomard, and G. Weinreich, editors, *Mechanics of Musical Instruments*, pages 116–183. Springer, New York, 1995.
- [271] S. A. van Duyne, J. R. Pierce, and J. O. Smith III. Travelling wave implementation of a lossless mode-coupling filter and the wave digital hammer. In *Proceedings of the International Computer Music Conference*, pages 411–418, Århus, Denmark, September 1994.
- [272] S. A. van Duyne and J. O. Smith III. Physical modelling with the 2D digital waveguide mesh. In *Proceedings of the International Computer Music Conference*, pages 40–47, Tokyo, Japan, September 1993.
- [273] S. A. van Duyne and J. O. Smith III. A simplified approach to modelling dispersion caused by stiffness in strings and plates. In *Proceedings of the International Computer Music Conference*, pages 407–410, Århus, Denmark, September 1994.
- [274] S. A. van Duyne and J. O. Smith III. The 3D tetrahedral digital waveguide mesh with musical applications. In *Proceedings of the International Computer Music Conference*, pages 9–16, Hong Kong, August 1996.
- [275] M. van Valkenburg. *Network Analysis*. Dover, New York, 1975.
- [276] M. van Walstijn. Wave-based simulation of wind instrument resonators. *IEEE Signal Processing Magazine*, 24(2):21–31, 2007.

- [277] M. van Walstijn and G. Scavone. The wave digital tonehole model. In *Proceedings of the International Computer Music Conference*, pages 465–468, Berlin, Germany, August 2000.
- [278] M.-P. Verge, A. Hirschberg, and R. Caussé. Sound production in recorderlike instruments: II A simulation model. *Journal of the Acoustical Society of America*, 101:2925–2939, 1997.
- [279] C. Vergez and X. Rodet. Dynamic systems and physical models of trumpet-like instruments: A study and asymptotical properties. *Acta Acustica united with Acustica*, 86:147–162, 2000.
- [280] C. Vergez and X. Rodet. A new algorithm for nonlinear propagation of sound wave: Application to a physical model of a trumpet. *Journal of Signal Processing*, 4:79–88, 2000.
- [281] R. Vichnevetsky and J. Bowles. *Fourier Analysis of Numerical Approximations of Hyperbolic Equations*. SIAM, Philadelphia, 1982.
- [282] L. Vu-Quoc and S. Li. Invariant-conserving finite difference algorithms for the nonlinear Klein-Gordon equation. *Computer Methods in Applied Mechanics and Engineering*, 107:341–391, 1993.
- [283] G. Weinreich. Mechanical oscillations. In A. Hirschberg, J. Kergomard, and G. Weinreich, editors, *Mechanics of Musical Instruments*, pages 79–114. Springer, New York, 1995.
- [284] J. Woodhouse. Physical modeling of bowed strings. *Computer Music Journal*, 16(4):43–56, 1992.
- [285] J. Woodhouse. Self-sustained musical oscillators. In A. Hirschberg, J. Kergomard, and G. Weinreich, editors, *Mechanics of Musical Instruments*. Springer, New York, 1995.
- [286] K. S. Yee. Numerical solution of initial boundary value problems involving Maxwell’s equations in isotropic media. *IEEE Transactions on Antennas and Propagation*, 14:302–307, 1966.
- [287] Y. Zhiming. A nonlinear dynamical theory of non-classical plates. *Journal of Shanghai University*, 1(2):1–17, 1997.
- [288] D. Zicarelli. How I learned to love a program that does nothing. *Computer Music Journal*, 26(4):44–51, 2002.
- [289] D. W. Zingg. Comparison of high-accuracy finite-difference methods for linear wave propagation. *SIAM Journal of Scientific Computing*, 22(2):476–502, 2000.
- [290] D. W. Zingg and H. Lomax. Finite-difference schemes on regular triangular grids. *Journal of Computational Physics*, 108:306–313, 1993.
- [291] E. Zwicker. *Psychoacoustics*. Springer.

Index

- accuracy
 - spectral methods and, 247
- acoustic tube
 - 1D wave equation and, 118
- adaptor, 63
- additive synthesis, 3–5
- Adrien, J.-M., 2
- air column, 118
- aliasing, 23
- AM synthesis, 7
- Ames, W., 252
- amplification factor, 35, 106
- amplification polynomial, 107
- analysis-resynthesis, 4
- analysis-synthesis methods, 2
- ansatz, 98, 106
- artificial reverberation, 15, 24
- artificial viscosity, 29
- aspect ratio, 220

- Bank, B., 212
- bilinear transformation, 41

- Cadoz, C., 2
- Cage, J., 181
- Cauchy-Schwartz inequality, 110
- cent, 73
- CFL, *see* Courant-Friedrichs-Lewy bound
- Chaigne, A., 2, 19
- Chebyshev grid, 248
- Chebyshev norm, 128
- Chebyshev polynomials
 - waveshaping and, 8
- Chowning, J., 2, 7
- circulant delay networks, 15
- collisions, 82–84
- collocation methods, 247–248
- Courant Friedrichs Lewy stability condition, 117
- Courant number, 131
- critical band, 24
- cutoff, 152

- d’Alembert, 123
- degrees of freedom
 - for 1D wave equation, 129
- digital filters
 - difference operators and, 38–41
- digital waveguide synthesis
 - efficiency of, 24
- digital waveguides, 13–16, 118
- Dirichlet condition, 109, 124
- dispersion relation, 98, 134
 - for 1D wave equation, 122

- energy
 - for 1D wave equation, 124
- energy method, 52, 58

- fast Fourier transform (FFT), 2
 - spectral methods and, 247
 - spectral methods and, 248
- feedback delay networks, 15
- Fender Rhodes electric piano, 186
- Fettweis, A., 2
- finite difference time-domain (FDTD) methods, 18
- finite difference time-domain FDTD methods, 53
- finite element methods, 19–20, 53
- five-point Laplacian, 223, 230
- Florens, J.-L., 2
- FM synthesis, 49
 - waveshaping and, 8
- FM synthesis, 7–8
- FOF, 9
- Fourier series
 - spectral methods and, 247
- Fourier transform, 97
- frequency warping, 10
 - in difference schemes for the SHO, 56
- functional transformation method (FTM), 11

- Galerkin methods, 247–248
- Garber, M., 247
- Giordano, N., 19
- granular synthesis, 8–9
- group velocity
 - for 1D wave equation, 122
 - for ideal bar equation, 160

- Hall, D., 85
- Helie, T., 11
- Helmholtz motion, 177
- Helmholtz, H., 177

- ideal bar
 - longitudinal motion of, 118
- inharmonicities
 - psychoacoustic significance of, 24
- inharmonicities factor, 172
- initial conditions
 - for 1D wave equation, 120
 - for traveling waves, 123
- inner product
 - in 1D, definition of, 99
- instantaneous frequency, 4
- interpolation, 6

- Jaffe, D., 13

- Karplus Strong algorithm, 13
 Karplus, K., 2
 Kelly, J., 10
 Kelly-Lochbaum speech synthesis model, 15, 17
- Laplace transform, 34, 97
 LC circuit, 50
 Leslie speaker, 239
 Lochbaum, C., 10
 Luciani, A., 2
 Lyapunov stability, 46
- marimba, 194
 masking, 24
 mass-spring system, 50
 Mathews, M., 2, 7
 modal synthesis, 97
 spectral methods and, 247
- Neumann condition, 124
 nine-point Laplacian, 223
 non-dimensionalization, 96, 120
 norm
 in 1D, definition of, 99
 numerical dispersion, 28, 41, 107, 247
 numerical phase velocity, 41
- parameter reduction, 120
 partial differential equation (PDE)
 1D wave equation, 118, 120
 1D wave equation with loss, 151
 2D wave equation, 227
 classification of(, 96
 classification of), 97
 for bar of variable cross-section, 194
 for ideal bar, 159
 for ideal bar, dimensionless form of, 160
 for stiff string, 170
 for stiff string with frequency dependent loss,
 173
 for string of variable density, 191
 Kirchhoff-Carrier equation, 201
 series-approximated system for nonplanar string
 vibration, 213
 system for planar string vibration
 general form of, 207
 series-approximated form of, 209
 Webster's equation, 215
 PDE, *see* partial differential equation
 phantom partials, 201
 phase plane analysis, 46
 phase velocity
 for 1D wave equation, 122
 for ideal bar equation, 160
 phase vocoder, 5
 pitch
 for vibrating bars, 163
 pitch flattening, 91
 pitch glides, 15, 80, 201, 241
 plucks, 121
 port resistance, 62
 prepared piano, 181
- quadratic forms , 44–46
- Rabenstein, R., 2, 11
 raised cosine distribution
 in 1D, 121
 Raman, C., 177
 reciprocity, 53
 Riemann sum, 110
 ring modulation, 7
 Risset, J.-C., 3
 Routh-Hurwitz test, 37
 Ruiz, P., 2, 10, 18
 Runge-Kutta methods, 38, 53
- sampling frequency, 28
 sampling synthesis, 7
 sampling theorem, 28
 scattering junction, 233
 Schur-Cohn recursion, 37
 short-time Fourier transformation, 5
 Smith III, J.O., 3, 13
 Smith, J.O., 2, 233
 source/filter models, 5
 spectral methods, 20
 modal synthesis and, 22, 247
 spectral modeling synthesis (SMS), 5
 spring reverberation, 239
 stick-slip dynamics, 90
 strikes, 121
 string
 1D wave equation and, 118
 Strong, A., 2
 Sturm-Liouville problem, 11
 subharmonic generation, 241
 subtractive synthesis, 5, 49
 symbol, 98
 symplectic methods, 58
- tension modulation, 15, 24, 201
 plate vibration and, 241
 test function, 98
 Thomas algorithm, 108
 time series, 28–29
 time shift operator, 29
 Timoshenko beam theory, 99
 Timoshenko theory, 160
 Toeplitz matrices, 109
 transmission line
 1D wave equation and, 118
 transmission line matrix method (TLM), 234
 trapezoid rule, 41, 62
 traveling waves, 123
 triangle inequality, 110
 tuning fork, 186
- van Duyne, S., 233
 Volterra series, 11
 von Neumann analysis, 52
 VOSIM, 9
- wave digital filters, 41, 50, 118
 wave speed, 118
 wave variables, 62
 waveguide mesh, 15
 wavenumber, 97
 waveshaping, 8
 wavetable synthesis, 5–7, 49

- weighted residual methods, 247
- width
 - temporal, of difference operator, 30–31
- z transform, 34–35