

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/392191213>

Eigenvalue Problems and Modal Schemes in Musical Acoustics: A Tutorial

Chapter · May 2025

CITATIONS

0

READS

104

1 author:



Michele Ducceschi

University of Bologna

83 PUBLICATIONS 575 CITATIONS

SEE PROFILE

Eigenvalue Problems and Modal Schemes in Musical Acoustics: A Tutorial

Michele Ducceschi

Department of Industrial Engineering, University of Bologna, Italy

1 Introduction

Acoustics is a large research topic concerned with modelling, measuring, analysing, and simulating sound waves propagating in elastic media. Examples include elastic waves in cables and strings, rods, bars, tubes, membranes, plates and rooms, see Morse and Ingard (1987). The nature of sound propagation varies case by case, and physical models change accordingly. Cables, strings, rods and cylindrical tubes display the simplest kind of wave propagation, which, in a first approximation, can be described in terms of the equation by d'Alembert (1747); see also Fletcher and Rossing (2012). This is a one-dimensional type of wave equation for which extensive analytical tools exist. Most notably, the d'Alembert equation admits an analytic solution in the form of two travelling wavefronts that never change shape or size. Some computational models for the wave equation exploit such a property to achieve an extremely efficient simulation, such as *digital waveguides* developed by Smith (1987, 2010). The mathematical validity of the simple wave equation is limited to lossless, non-dispersive systems in one dimension. When losses and wave dispersion are included, the shape of the initial wavefronts changes over time, and the d'Alembert solution is no longer valid. More involved acoustic systems not described by the d'Alembert equation also require different solution techniques. For these reasons, it is worth introducing the simulation methods that can be applied to all such problems, starting from general principles.

Numerical simulation is now commonplace among scientists in the field. Commercially available software has enabled the study of systems that cannot be approached analytically, and many computational tools and frameworks exist. *Finite elements*, a kind of numerical method in which the solution is computed as a superposition of elementary functions with compact support, occupy a central role in many branches of computational

physics, including acoustics, see Hughes (1987); Bathe (1996). In this lecture, however, we will develop the method of *finite differences*, which are conceptually very different, see LeVeque (2007); Strikwerda (2004). Instead of expanding the solution in terms of elementary functions, finite differences discretise the differential operators directly at specific locations along the domain, called the *nodes*. The continuous solutions are then represented by an approximate grid function defined at the nodes. When the problem is time-dependent, as is often the case in acoustics, the grid functions are updated in time, which is also discretised. Time updating requires storing the grid function's values from previous time steps. There are multiple reasons why one would want to learn the method of finite differences:

- They are conceptually simple. The basic workings of finite differences can be understood using high-school calculus.
- They occupy a central role in computational acoustics in classical and modern applications. Examples of recent developments of the finite difference method in acoustics include the simulation of large acoustic spaces (see, e.g., Hamilton and Bilbao (2016); Bilbao et al. (2016); Kowalczyk and van Walstijn (2011); Botts and Savioja (2014)) and nonlinear systems such as strings and plates (see, e.g., Ducceschi and Bilbao (2022, 2019); Ducceschi et al. (2014); Bilbao et al. (2015))
- No matter which computational tool one adopts to approximate the spatial part of a differential problem, time integration is performed using finite differences.
- Recent applications have overcome traditional limitations imposed by the method, particularly in discretising non-Cartesian domains and complex boundary conditions, see Bilbao (2022); Bilbao and Hamilton (2013).

Figure 1 summarises the finite element and finite difference paradigms.

1.1 The finite difference method

The working principle of finite differences is straightforward: one approximates derivatives of functions to turn a differential problem into an algebraic one. The finite difference method can be applied to discretise both the temporal and the spatial domains, though the two present difficulties of their own and must be approached using specific discretisation techniques. A typical problem arising in time discretisation concerns the growth of high-frequency oscillations, an issue known at least since the seminal work by Courant et al. (1928). When such growth is unbounded, numerical instability ensues as the numerical solution diverges from the true solution.

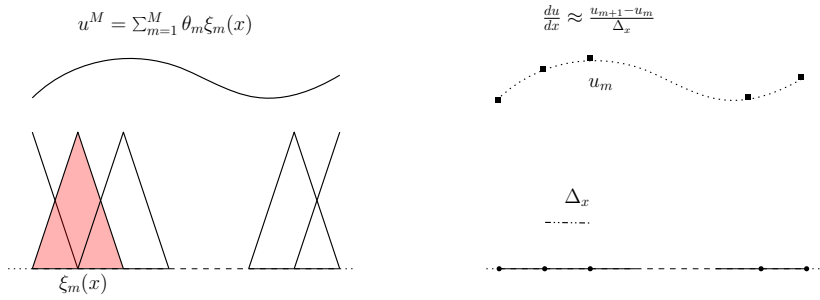


Figure 1. Finite elements (left) and finite differences (right). In the FEM paradigm, a solution is discretised using a finite sum of simple test functions with a compact support (the elements). In the FD paradigm, differential operators are discretised on a grid of points.

This unwanted numerical behaviour may be avoided by respecting appropriate *stability conditions*, as found using various analytical techniques such as Fourier-based methods or energy methods, see Ha-Duong and Joly (1994); Joly (1989). On the other hand, difficulties arising in spatial discretisation include the correct realisation of the boundary geometry and the setting of appropriate numerical boundary conditions.

The temporal and spatial problems become closely intertwined in discretising partial differential equations. Stability conditions yield the smallest grid spacing for a given time step or the largest time step for a given mesh size. In some other cases, the schemes are unconditionally stable, allowing one to select any time step and grid spacing combination. Not all choices will return a meaningful output as other numerical artefacts arise, such as frequency warping effects and other kinds of numerical distortion severely affecting any musical application, see Bilbao (2009). Designing stable schemes with the least amount of numerical artefacts is the core objective of any numerical simulation.

1.2 Boundary Value Problems

Because of the inherent difficulties in discretising the temporal and spatial domains, it is worth approaching the two problems separately. In a typical numerical workflow, the spatial part is discretised first. This turns the original differential problem into an initial-value problem for a system of coupled time-dependent equations, which may be updated in time using a time-stepping scheme. Thus, it is worth approaching the study of finite

differences beginning with the theory of boundary value problems (BVPs). In acoustics, typical boundary-value problems include the *Laplace equation* and the *biharmonic equation*, with or without a source*. Without a source, solutions to such problems may be found in one dimension by direct integration. Such solutions depend on a few constants of integration, which are fixed by imposing appropriate boundary conditions. Since the performance of the numerical approximations can be assessed against the closed-form solution obtained via direct integration, approaching these simple problems numerically allows one to understand the working principles of the finite difference method. Concepts such as the order of accuracy and the order of convergence may be understood through such examples, see LeVeque (2007). More complex problems in one spatial dimension arise when the medium properties are space-dependent. Examples include acoustic tubes with a variable cross-section or cables and rods with non-uniform material and geometric properties. In these cases, analytic solutions, if any, are much harder to obtain, and a numerical approach is necessary. Non-uniform mesh sizes can be adapted to account for such spatial variability, such as through the use of mimetic operators introduced by Shashkov (1996).

Solutions to BVPs in more than one space dimension are seldom available in closed form. Notable exceptions include problems defined on a cartesian domain with simple boundary conditions. Problems defined over non-cartesian domains or presenting more complex boundary conditions can only be approached numerically using extensions of the techniques developed for the one-dimensional case. Particular care must be taken to ensure that the boundary geometry is realised appropriately, keeping the overall accuracy of the discretisation unchanged compared to the interior points.

1.3 Eigenvalue and Frequency Domain Problems

When the problem is time-dependent, but certain assumptions on the system's time evolution hold, a reduction to some form of BVP is usually possible. The most notable example in acoustics is a time-dependent system's unforced, undamped, steady state, see Meirovitch (1970). The solutions to this problem are time-harmonic and, hence, the BVP is an *eigenvalue problem* (in the case of the Laplace operator, the defining equation is known as the *Helmholtz equation*). Time-harmonic solutions also arise when a system is forced at a specific frequency. In this case, if a single point in space can approximate the spatial extent of the source, the solution to the spatial differential problem is a *Green's function*. Such descriptions are commonplace in acoustics, where problems are often linear and time-invariant

*The Laplace equation with a source is known as the *Poisson equation*

and can be solved equivalently in the time and frequency domains. The frequency-domain approach is sometimes preferable since closed-form solutions are easier to obtain, and most acoustic wave propagation properties in media are naturally described as a function of frequency, such as decay times, absorption coefficients, and so on. Additionally, frequency-domain approaches serve as the basis for many inverse modelling techniques, such as material parameter estimation, see e.g. Viala et al. (2018); Ducceschi et al. (2024).

1.4 Modal Methods

Applications in the frequency domain are restricted to steady-state or time-harmonic problems. However, simulating transient responses requires an appropriate update of the acoustic equations over time. Once spatial semi-discretisation is carried out through finite differences or finite elements, the resulting system of ordinary differential equations may be advanced in time using a suitable time-stepping scheme. Time is often discretised using a finite-difference approach, typically employing a constant sample rate throughout the simulation. Many approaches, collectively known as *time-domain* methods, operate directly on the spatially discretised equations without further transformation. Examples range from room acoustics (Okuzono and Yoshida (2022); Bilbao and Hamilton (2013); Kowalczyk and van Walstijn (2011)), to musical instrument simulation (Tournemenne and Chabassier (2019); Thibault and Chabassier (2021); Chabassier and Joly (2015); Bilbao (2009); McIntyre et al. (1983)). Time-domain methods have numerous applications in acoustics. Finite-difference time-domain (FDTD) techniques gained prominence in the 1990s, largely due to the pioneering work of Chaigne and colleagues (see, e.g., Chaigne (1992); Chaigne and Askenfelt (1994); Doutaut et al. (1998)), and have since become a key simulation method in acoustics, see Bilbao (2009). Finite-element approaches in the time domain have experienced a similar trajectory, thanks to influential work by Hughes (Hughes (1987); Hughes and Tzou (1977)), Joly and associates (Joly (2008); Duruflé et al. (2009); Cohen et al. (1995)), and others. Besides time-domain and frequency-domain approaches, hybrid methods exist. Modal methods are one such prominent example. In this approach, spatial discretisation is first completed, after which the resulting semi-discretised system is projected onto a modal basis to derive the modal equations. These modal equations are then advanced in time, and the solution is reconstructed at specific output points using a reduced modal sum, see Morrison and Adrien (1993); Adrien (1991); Woodhouse (2004); Bank and Sujbert (2005); Bank (2010). This approach entails additional steps

before time integration compared to purely time-domain methods. These preliminary steps are often performed offline and may become a computational bottleneck for larger systems. However, once the modal equations are obtained, the system reduces to a set of parallel oscillators that can be updated highly efficiently over time. Additionally, exact time integrators exist for linear problems, effectively eliminating artefacts from time discretisation errors, see e.g. van Walstijn et al. (2016, 2024). This modal approach may thus be preferable for certain acoustic systems and will be further illustrated in this lecture. Of course, synthesis methods exist that employ a variety of numerical techniques, such as the boundary conditions and body radiativity via modal methods and wave propagation via time-domain methods, see e.g. Maestre et al. (2018, 2017).

1.5 Lecture Outline

This lecture develops the method of finite differences from first principles. Section 2 derives approximations of derivatives in one dimension, demonstrating how difference operators may be constructed using straightforward calculus. Interpolation, a somewhat more advanced topic, is subsequently introduced, enabling the construction of more general difference operators through the coefficients of interpolating polynomials. This approach proves particularly effective in constructing difference operators for functions sampled at non-uniform intervals. Grids and grid functions are introduced in Section 3, building upon the preceding discussion. Section 4 presents the first application of the finite difference method via the solution of simple boundary value problems in one dimension. These problems, which admit analytical solutions through direct integration, introduce concepts such as the order of convergence of a difference scheme. Both uniform and non-uniform grids are employed in the solutions, disproving the common misconception that difference schemes are restricted to domains with constant mesh sizes. Eigenvalue problems in acoustics are explored in Section 5, specifically in the context of the one-dimensional wave equation and the Euler-Bernoulli beam with variable thickness. Applications to musical acoustics include shape optimisation for overtone tuning in marimbas. Section 6 extends the discussion to eigenvalue problems in two dimensions, focusing on the orthotropic Kirchhoff-Love plate equation modelling musical instrument plates and soundboards. The MAGPIE toolbox is introduced as a means of solving the orthotropic plate equation in both the frequency and time domains. Finally, Section 7 presents a time-stepping scheme for the modal equations, with illustrative examples drawn from the MAGPIE toolbox.

1.6 Code repositories

Example Matlab code is uploaded to the following GitHub repositories:

- CISM.Udine2024.git. Within the `src` folder, the user will find two subfolders: `FiniteDiffCoefficients` covering most examples from Section 2; `EulerBernoulliBars` covering the marimba test cases in Section 5.
- magpie-matlab.git This repository stores the MAGPIE toolbox for treating the orthotropic plate problem described in Section 6.

Both repositories are open-source and distributed under the MIT license.

2 Approximation of derivatives

Suppose one wants to derive an approximate value for the derivative of a function of one variable. Let the function be $u = u(x) : \mathcal{I} \subseteq \mathbb{R} \rightarrow \mathbb{R}$, where \mathcal{I} is a *closed interval* in \mathbb{R} , and let the function be sufficiently smooth, such that one may compute derivatives u' , u'' , ... and these are continuous. Furthermore, consider the small parameter $\Delta_x > 0$. The derivative of $u(x)$ at $x_0 \in \mathcal{I}$ may be defined in terms of the limit of a difference:

$$u'(x_0) = \lim_{\Delta_x \rightarrow 0} \frac{u(x_0 + \Delta_x) - u(x_0)}{\Delta_x}. \quad (1)$$

Of course, this definition is not unique.

Definition 2.1. The *identity*, *forward shift* and *backward shift* operators are defined as, respectively:

$$1u(x) = u(x), \quad e_{x+}u(x) = u(x + \Delta_x), \quad e_{x-}u(x) = u(x - \Delta_x) \quad (2)$$

□

Definition 2.2. The *forward*, *backward*, *centred* difference operators are defined as, respectively:

$$\delta_{x+} = \frac{e_{x+} - 1}{\Delta_x}, \quad \delta_{x-} = \frac{1 - e_{x-}}{\Delta_x}, \quad \delta_{x\cdot} = \frac{e_{x+} - e_{x-}}{2\Delta_x}. \quad (3)$$

□

These definitions can be used interchangeably to define the continuous derivative, such that one has:

$$u'(x_0) = \lim_{\Delta_x \rightarrow 0} \delta_{x+}u(x_0) = \lim_{\Delta_x \rightarrow 0} \delta_{x-}u(x_0) = \lim_{\Delta_x \rightarrow 0} \delta_{x\cdot}u(x_0). \quad (4)$$

Applying the definition of the difference operators to the function $u(x)$ yields the expressions:

$$\delta_{x+}u(x) = \frac{u(x + \Delta_x) - u(x)}{\Delta_x}, \quad (5a)$$

$$\delta_{x-}u(x) = \frac{u(x) - u(x - \Delta_x)}{\Delta_x}, \quad (5b)$$

$$\delta_x.u(x) = \frac{u(x + \Delta_x) - u(x - \Delta_x)}{2\Delta_x}. \quad (5c)$$

When Δ_x is finite, the difference operators yield an approximate value of the derivative, and an error is introduced.

2.1 Truncation errors

The error introduced by the difference operators is obtained via Taylor series arguments. Since the function $u(x)$ is smooth, one has:

$$u(x_0 + \Delta_x) \approx u(x_0) + \Delta_x u'(x_0) + \frac{\Delta_x^2}{2} u''(x_0) + \frac{\Delta_x^3}{6} u'''(x_0), \quad (6a)$$

$$u(x_0 - \Delta_x) \approx u(x_0) - \Delta_x u'(x_0) + \frac{\Delta_x^2}{2} u''(x_0) - \frac{\Delta_x^3}{6} u'''(x_0). \quad (6b)$$

These can be used to infer the order of the approximation of the difference operators:

$$\delta_{x+}u(x_0) \approx u'(x_0) + \frac{\Delta_x}{2} u''(x_0) = u'(x_0) + \mathcal{O}(\Delta_x), \quad (7a)$$

$$\delta_{x-}u(x_0) \approx u'(x_0) - \frac{\Delta_x}{2} u''(x_0) = u'(x_0) + \mathcal{O}(\Delta_x), \quad (7b)$$

$$\delta_x.u(x_0) \approx u'(x_0) + \frac{\Delta_x^2}{6} u'''(x_0) = u'(x_0) + \mathcal{O}(\Delta_x^2). \quad (7c)$$

Note that the odd powers drop out of the Taylor series for δ_x , a property typical of centred operators.

Definition 2.3. The *truncation error* is defined as (LeVeque (2007)):

$$E_{\delta_o}(u(x_0)) := \delta_o u(x_0) - u'(x_0), \quad (8)$$

where δ_o is any of the difference operators defined in (3).

Definition 2.4. The *order of accuracy* p of a difference operator is defined as the exponent of the leading term in the Taylor series of the error E :

$$E_{\delta_o} \approx C \Delta_x^p, \rightarrow p = \log(\Delta_x)^{-1} (\log(|E_{\delta_o}|) - \log(|C|)), \quad (9)$$

Δ_x	$E_{\delta_{x+}}$	$E_{\delta_{x-}}$	E_{δ_x}
0.1000	-0.135	0.133	-0.00127
0.0178	-0.024	0.0239	-4.03e-5
0.0032	-0.00426	0.00425	-1.27e-6
0.0006	-7.57e-4	7.57e-4	-4.03e-8
0.0001	-1.35e-4	1.35e-4	-1.27e-9

Table 1. Truncation errors.

where δ_\circ is any of the difference operators defined in (3) \square

Thus, from (7), the forward and backward differences are *first-order* accurate, whereas the centred difference is *second-order* accurate.

Example 2.5. To understand the behaviour of the difference operators, the derivatives of $e^{\sin(x)}$ at $x_0 = 1.5$ are computed using the definitions (3). Figure 2 reports the slopes and the error trends of the finite difference oper-

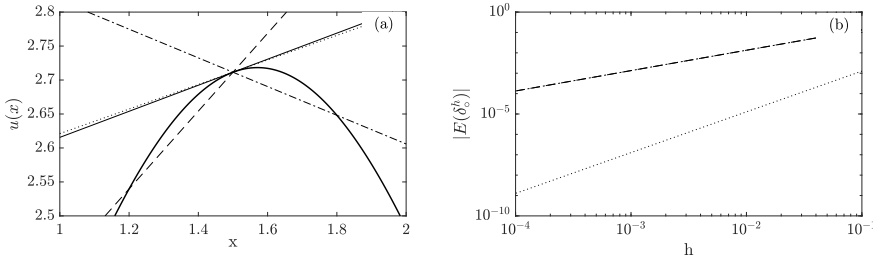


Figure 2. Approximation of the derivative of $u(x) = e^{\sin(x)}$ using finite differences. (a): slopes obtained using δ_{x+} (dashed), δ_{x-} (dash-dotted), δ_x (dotted), and exact (solid). In all cases, $x_0 = 1.5$, $\Delta_x = 0.3$. The continuous, thick line is $u(x)$. (b): log-log plot of the absolute value of the truncation error for the difference operators, with the same line style as panel (a).

ators, under various values of the small parameter Δ_x . The error values are reported in Table 1. One important aspect concerns the power expansions (7): the derivatives of $u(x)$ appearing in them do not depend on Δ_x , and are, therefore, constants when the expansion point x_0 is kept fixed. Thus, one should expect the errors in Table 1 to have the form (7), with the expansion coefficients given by the appropriate derivatives of $u(x)$. This is,

indeed, the case. Running a polynomial fit on the columns of Table 1 yields:

$$E_{\delta_{x+}} \approx -1.3540 \Delta_x, \quad E_{\delta_{x-}} \approx 1.3280 \Delta_x, \quad E_{\delta_x} \approx -0.1269 \Delta_x^2,$$

and note that:

$$\frac{u''(x_0)}{2} \approx -1.3456, \quad \frac{u'''(x_0)}{6} \approx -0.1275, \quad (10)$$

Hence, the numerical error trends are in good agreement with the corresponding analytical expressions. Since the errors are defined as powers, according to definition (9), log plots such as panel (b) of Figure 2 can be employed, so that the order of accuracy p appears as the slope of the error lines \square

Before proceeding, it is worth illustrating the Taylor series of the backward and forward difference operators when applied at $x_0 \pm \Delta_x/2$. Formally:

$$\begin{aligned} \delta_{x+} u \left(x_0 - \frac{\Delta_x}{2} \right) &= \delta_{x-} u \left(x_0 + \frac{\Delta_x}{2} \right) \\ &= \frac{u \left(x_0 + \frac{\Delta_x}{2} \right) - u \left(x_0 - \frac{\Delta_x}{2} \right)}{\Delta_x}. \end{aligned} \quad (11)$$

Note that these operators are centred around x_0 and, hence, one can expect the odd powers to drop out of the Taylor series, yielding higher-accurate operators. This is, indeed, the case. One has:

$$\begin{aligned} \delta_{x+} u \left(x_0 - \frac{\Delta_x}{2} \right) &= \delta_{x-} u \left(x_0 + \frac{\Delta_x}{2} \right) \\ &\approx u'(x_0) + \frac{\Delta_x^2}{8} u'''(x_0) = u'(x_0) + \mathcal{O}(\Delta_x^2). \end{aligned}$$

2.2 Vandermonde and Lagrange interpolation

The discussion in the previous section suggests building the difference operators using the values of a function at arbitrary locations, provided the correct coefficients are applied at the sampled points of the function. In essence, the problem of finding appropriate difference coefficients amounts to finding an appropriate interpolating function (LeVeque (2007)). See, e.g. Davis (1975); Hamming (1973) for an introduction to interpolation.

Vandermonde interpolation One may find such weights using the Taylor series approach. Suppose to have sampled the function $u(x)$ at x_0 , $x_0 + \Delta_x$, $x_0 + 2\Delta_x$. The sample points are equally spaced in this case, but

the technique described below would work no matter where the points are located. In this example, the difference operator is defined as the following:

$$\delta_{x\star}u(x_0) := c_0 u(x_0) + c_1 u(x_0 + \Delta_x) + c_2 u(x_0 + 2\Delta_x), \quad (12)$$

for unknown weight coefficients c_0, c_1, c_2 . One may expand $u(x_0 + \Delta_x)$ as suggested in (6), and use an analogous expansion for $u(x_0 + 2\Delta_x)$. Inserting the expansions in (12), one obtains:

$$\delta_{x\star}u(x_0) \approx (c_0 + c_1 + c_2) u(x_0) + (c_1 + 2c_2)\Delta_x u'(x_0) + \frac{(c_1 + 4c_2)\Delta_x^2}{2} u''(x_0).$$

For this to approximate the first derivative, one requires:

$$c_0 + c_1 + c_2 = 0, \quad (c_1 + 2c_2)\Delta_x = 1, \quad c_1 + 4c_2 = 0, \quad (13)$$

yielding

$$c_0 = -\frac{3}{2\Delta_x}, \quad c_1 = \frac{2}{\Delta_x}, \quad c_2 = -\frac{1}{2\Delta_x}. \quad (14)$$

Notice that the coefficients are proportional to Δ_x^{-1} , and hence one can expect $\delta_{x\star}$ to be second-order accurate, that is:

$$\delta_{x\star}u(x_0) = u'(x_0) + \mathcal{O}(\Delta_x^2). \quad (15)$$

This approach can be extended further to derive a general formula for constructing difference operators. Suppose to have sampled the function $u(x)$ at $M + 1$ points x_0, \dots, x_M (not necessarily equally spaced) and to be wanting to compute the coefficients of an $M + 1$ -point difference operator acting at \bar{x} (this may or may not be equal to any of the sampling points x_m). One constructs a polynomial $p(x)$ of degree M as:

$$p(x) = \alpha_0 + \alpha_1(x - \bar{x}) + \alpha_2(x - \bar{x})^2 + \dots + \alpha_M(x - \bar{x})^M, \quad (16)$$

and imposes $p(x_0) = u(x_0)$, $p(x_1) = u(x_1)$, ..., $p(x_M) = u(x_M)$. This can be expressed conveniently via the *Vandermonde matrix* as:

$$\begin{bmatrix} 1 & (x_0 - \bar{x}) & (x_0 - \bar{x})^2 & \cdots & (x_0 - \bar{x})^M \\ 1 & (x_1 - \bar{x}) & (x_1 - \bar{x})^2 & \cdots & (x_1 - \bar{x})^M \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & (x_M - \bar{x}) & (x_M - \bar{x})^2 & \cdots & (x_M - \bar{x})^M \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_M \end{bmatrix} = \begin{bmatrix} u(x_0) \\ u(x_1) \\ \vdots \\ u(x_M) \end{bmatrix}. \quad (17)$$

Solving this system returns the coefficients α_m as a function of the sampled points $u(x_m)$. Note that the system is well-defined when the sample points are distinct. Then, the analytic expression of the $M + 1$ -point difference operator acting at \bar{x} is recovered by computing $p'(\bar{x})$, that is, α_1 .

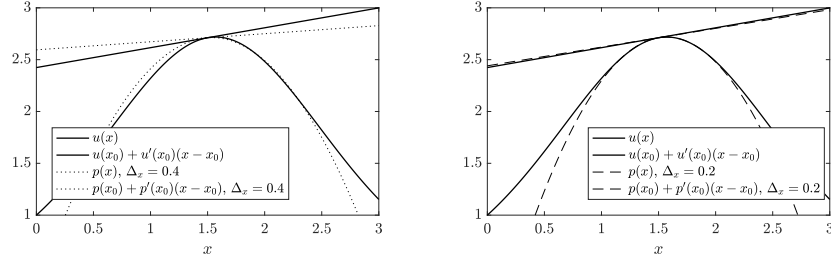


Figure 3. The function $u(x) = e^{\sin(x)}$, the interpolating polynomials $p(x)$ obtained via the Vandermonde method, using the sampled points $u(x_0)$, $u(x_0 + \Delta_x)$, $u(x_0 + 2\Delta_x)$, with $\Delta_x = 0.4$ (left) and $\Delta_x = 0.2$ (right) and $x_0 = 1.5$. The slopes at x_0 are also plotted.

Example 2.6. The coefficients c_0 , c_1 , c_2 in (12) are computed using the Vandermonde matrix method. To that end, note that $\bar{x} = x_0$, and that $x_1 - x_0 = \Delta_x$, $x_2 - x_0 = 2\Delta_x$. Thus:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & \Delta_x & \Delta_x^2 \\ 1 & 2\Delta_x & 4\Delta_x^2 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} u(x_0) \\ u(x_0 + \Delta_x) \\ u(x_0 + 2\Delta_x) \end{bmatrix}. \quad (18)$$

Solving the system, one obtains:

$$\alpha_1 = -\frac{3u(x_0)}{2\Delta_x} + \frac{2u(x_0 + \Delta_x)}{\Delta_x} - \frac{u(x_0 + 2\Delta_x)}{2\Delta_x}, \quad (19)$$

that is, the same as (12) with coefficients (14) \square

Lagrange interpolation In some cases, particularly for large M , the Vandermonde matrix becomes poorly conditioned, and small perturbations in the matrix entries produce large errors in the interpolating coefficients α_m . Whilst large values of $M + 1$ are, in practice, never needed to build difference operators, it is worth illustrating an alternative way of building the interpolating polynomial $p(x)$ through Lagrange polynomials. Applications of Lagrange interpolants are much wider than merely deriving finite difference coefficients and are thus worth discussing here. As before, assume to have sampled the function $u(x)$ at $M + 1$ points x_0, x_1, \dots, x_M .

Definition 2.7. The Lagrange basis functions $l_m(x)$, $m = 0, \dots, M$ are defined as:

$$\begin{aligned} l_m(x) &:= \frac{(x - x_0) \dots (x - x_{m-1})(x - x_{m+1}) \dots (x - x_M)}{(x_m - x_0) \dots (x_m - x_{m-1})(x_m - x_{m+1}) \dots (x_m - x_M)} \\ &= \frac{\prod_{n \neq m} (x - x_n)}{\prod_{n \neq m} (x_m - x_n)}, \end{aligned}$$

and the associated Lagrange polynomial is:

$$p(x) = \sum_{m=0}^M u(x_m) l_m(x). \quad (20)$$

Note that $p(x_m) = u(x_m) \forall m$, which is the required condition for interpolation \square

Compared to the Vandemorde matrix method, the Lagrange interpolant is built explicitly and not via the solution of a linear system. This allows the construction of polynomials of large degrees without much difficulty. Furthermore, if the function $u(x)$ is sampled further via $u(x_{M+1})$, $u(x_{M+2})$, ..., the first $M+1$ Lagrange basis functions can be computed by adjusting the old ones: one only needs to multiply the numerator and the denominator of each $l_m(x)$ by the factors associated with the new sample points. As before, the analytic expression of the $M+1$ -point difference operator acting at \bar{x} is recovered by computing $p'(\bar{x})$.

Example 2.8. The coefficients c_0 , c_1 , c_2 in (12) are computed using Lagrange interpolation. In this case, one has:

$$l_0 = \frac{(x - x_1)(x - x_2)}{2\Delta_x^2}, \quad l_1 = -\frac{(x - x_0)(x - x_2)}{\Delta_x^2}, \quad l_2 = \frac{(x - x_0)(x - x_1)}{2\Delta_x^2}. \quad (21)$$

Using these to construct $p(x)$ as in (20), and computing $p'(x_0)$, one obtains again the coefficients (14) \square

2.3 Approximation of higher derivatives

All the techniques illustrated in the case of the first derivative extend directly to the cases of higher derivatives.

Second derivative The second derivative in one dimension is the simplest example of the *Laplace operator*, appearing in many physical laws

modelling a variety of phenomena: elastic bending, wave propagation, and heat conduction, just to name a few. This operator, thus, deserves special treatment. In the continuous case, a definition of the second derivative is often given as:

$$\begin{aligned} u''(x_0) &= \lim_{\Delta_x \rightarrow 0} \frac{(\delta_{x+} - \delta_{x-})u(x_0)}{\Delta_x} \\ &= \lim_{\Delta_x \rightarrow 0} \frac{u(x_0 + \Delta_x) - 2u(x_0) + u(x_0 - \Delta_x)}{\Delta_x^2}. \end{aligned} \quad (22)$$

Definition 2.9. The *second difference* operator is defined as:

$$\delta_{xx} := \frac{\delta_{x+} - \delta_{x-}}{\Delta_x} = \frac{e_+ - 2 + e_-}{\Delta_x^2} \quad (23)$$

□

It is easy to obtain the Taylor series of this operator using (6). Thus, one has:

$$\begin{aligned} \delta_{xx}u(x_0) &= \frac{u(x_0 + \Delta_x) - 2u(x_0) + u(x_0 - \Delta_x)}{\Delta_x^2} \\ &\approx u''(x_0) + \frac{\Delta_x^2}{12}u''''(x_0) = u'' + \mathcal{O}(\Delta_x^2). \end{aligned}$$

Since δ_{xx} is a centred operator, the odd terms drop out of the series. Note as well that the error is second-order in the expansion parameter Δ_x . Various other definitions are possible, and obtainable using much of the same techniques as detailed in Section (2.2).

Example 2.10. The second derivative is approximated at x_0 via Lagrange interpolation, by sampling $u(x)$ at the points $x_{-2} = x_0 - 2\Delta_x$, $x_{-1} = x_0 - \Delta_x$, x_0 , $x_1 = x_0 + \Delta_x$, $x_2 = x_0 + 2\Delta_x$. In this case, one has:

$$\begin{aligned} l_{-2} &= \frac{(x - x_{-1})(x - x_0)(x - x_1)(x - x_2)}{24\Delta_x^4}, \\ l_{-1} &= \frac{(x - x_{-2})(x - x_0)(x - x_1)(x - x_2)}{6\Delta_x^4}, \dots \end{aligned}$$

These are used to construct $p(x)$ as in (20). To obtain the second difference operator, the second derivative is required, giving:

$$p''(x_0) = -\frac{u(x_0 - 2\Delta_x)}{12\Delta_x^2} + \frac{4u(x_0 - \Delta_x)}{3\Delta_x^2} - \frac{5u(x_0)}{2\Delta_x^2} + \frac{4u(x_0 + \Delta_x)}{3\Delta_x^2} - \frac{u(x_0 + 2\Delta_x)}{12\Delta_x^2}.$$

Computing the Taylor series, one has:

$$p''(x_0) = u''(x_0) + \mathcal{O}(\Delta_x^4), \quad (24)$$

and, hence, fourth-order accuracy is achieved. Whilst the definition of δ_{xx} was given in terms of the difference of δ_{x+} and δ_{x-} , note that the following identity holds, as one may show immediately:

$$\delta_{xx} = \delta_{x+}\delta_{x-} = \delta_{x-}\delta_{x+}. \quad (25)$$

Thus, higher difference operators may be constructed by *composition*. This idea will be exploited later when interpreting finite difference operators as matrices acting on grid functions: in that framework, the composition of the operators is realised by multiplying the corresponding matrices \square

Third and fourth derivatives One may go on and build difference operators for higher derivatives. The fourth derivative, also known as the *biharmonic operator*, appears in the models of the thin bar and is worth introducing here. One has:

$$u''''(x_0) = \lim_{\Delta_x \rightarrow 0} \frac{u''(x_0 - \Delta_x) - 2u''(x_0) + u''(x_0 + \Delta_x)}{\Delta_x^2} = \lim_{\Delta_x \rightarrow 0} \frac{u(x_0 - 2\Delta_x) - 4u(x_0 - \Delta_x) + 6u(x_0) - 4u(x_0 + \Delta_x) + u(x_0 + 2\Delta_x)}{\Delta_x^4}.$$

Definition 2.11. The *fourth difference operator* is defined as:

$$\delta_{xxxx} := \delta_{xx}\delta_{xx} = \frac{e_{x-}^2 - 4e_{x-} + 6 - 4e_{x+} + e_{x+}^2}{\Delta_x^4} \quad (26)$$

\square

Expanding the operator using a Taylor series, one obtains:

$$\delta_{xxxx}u(x_0) = u''''(x_0) + \mathcal{O}(\Delta_x^2), \quad (27)$$

that is, the operator is second-order accurate.

Besides the fourth derivative, third derivatives appear when analysing the boundary conditions of the biharmonic operator. The composition of operators can be used such that $\delta_{xx}\delta_{x+}$, $\delta_{xx}\delta_{x-}$, and $\delta_{xx}\delta_{x\cdot}$ can all be employed to approximate the third derivative. The third difference operators will be denoted as follows:

$$\delta_{xxx+} := \delta_{xx}\delta_{x+}, \quad \delta_{xxx-} := \delta_{xx}\delta_{x-}, \quad \delta_{xxx\cdot} := \delta_{xx}\delta_{x\cdot}. \quad (28)$$

The order of the approximations may be computed via the Taylor series and left as an exercise for the reader. It is easy to show that both δ_{xxx+} and δ_{xxx-} are first-order accurate operators: they inherit this property directly from the first-order accuracy of the difference operators δ_{x+} , δ_{x-} . The centred operator δ_{xxx} is second-order accurate, as all the odd terms drop out of its Taylor series.

3 Grids and grid functions

The concepts introduced in the previous sections allow estimating the derivatives of a sampled, continuous function $u(x)$ defined over a closed interval $\mathcal{I} \subseteq \mathbb{R}$. The finite difference coefficients appearing in the definitions of the difference operators can be found by interpolating the continuous function at the available sample points x_m , yielding several difference operators with varying accuracy.

It is yet unclear how one may use such techniques to solve a differential problem comprising $u(x)$ and its derivatives. The method of finite differences is used to compute an *approximate* solution at specific locations in \mathcal{I} . Suppose that the interval is bounded so that $\mathcal{I} = \{x \mid 0 \leq x \leq L\}$. Just like several sample points were used above to interpolate polynomials, a set of discrete points belonging to \mathcal{I} is used to compute an approximate solution to a differential problem.

Definition 3.1. One refers to a *grid* (sometimes also called a *mesh*) as the collection of *grid points* (also called the *nodes*) x_m and related *cells* $[x_m, x_{m+1}]$ such that:

$$0 = x_0 < x_1 < \dots < x_m < x_{m+1} \dots < x_M = L \quad (29)$$

□

Various kinds of grids can be defined, see e.g. Shashkov (1996):

- The simplest grid is certainly the *uniform grid*: here, all the grid points are separated by the *grid spacing* Δ_x , and one has $x_m = m\Delta_x$, $m \in [0, M] \subset \mathbb{N}$, and the number of grid intervals is $M = \Delta_x/L$. This type of grid is often used in the simulation of isotropic, homogeneous problems in which the properties of the medium are independent of space.
- The *smooth, non-uniform grid* is defined by the smooth function $y(x)$, and one has $y_m = y(x_m)$. The smooth function y maps \mathcal{I} onto itself and is one-to-one. An example is given by $y = (L^2 - x^2)/L$.
- The *non-smooth, non-uniform grid* is a non-uniform grid where $y(x)$ is not smooth. An example is given by a grid whose distance between

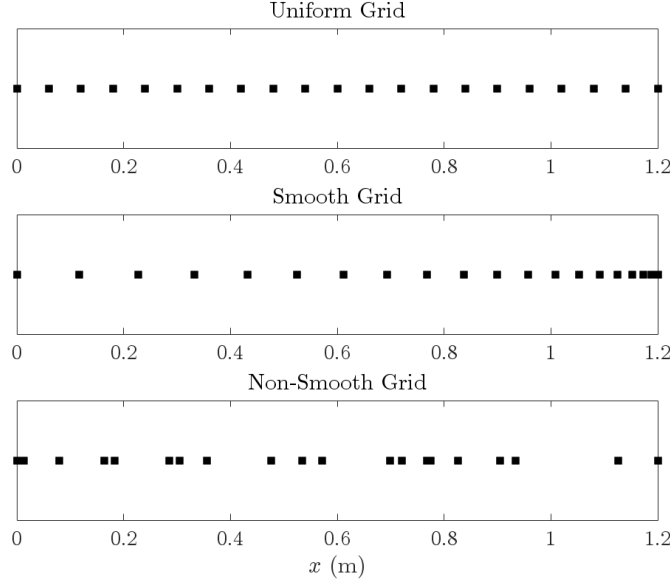


Figure 4. Examples of uniform, smooth and non-smooth grids, using $M = 20$ and $L = 1.2$. The smooth grid is obtained as $y_{M-m} = (L^2 - x_m^2)/L^2$.

nodes alternates $\Delta_x, 2\Delta_x, \Delta_x, \dots$, or where the grid intervals are sampled from a random distribution.

In all cases, the length between two nodes is bounded by a smallest and by a largest value, such that

$$C_{\min} \Delta_x \leq x_{m+1} - x_m \leq C_{\max} \Delta_x, \quad (30)$$

where Δ_x is a typical cell size (for instance, the mean value of all the cells).

Example 3.2. For the uniform grid, $C_{\min} = C_{\max} = 1 \forall M$. For the grid defined by $y_{M-m} = (L^2 - x_m^2)/L$ with $L = 1.2$, $M = 120$, one has $\Delta_x = 0.01$, $C_{\min} = 0.0083$, $C_{\max} = 1.9917$ \square

Another important grid is the *interleaved grid*, parametrised by a half-integer. So, all the grid points are still separated by a Δ_x , but they are found at $x_{m+\frac{1}{2}} = (m + \frac{1}{2})\Delta_x$, $m \in [1, M-1] \subset \mathbb{N}$

Definition 3.3. A *grid function* $\mathbf{u} : \mathcal{I} \rightarrow \mathbb{R}^{M+1}$ is any $M+1 \times 1$ vector defined at the grid points x_m . In practice, grid functions approximate the

Group	Definition
Identity	$1u_m = u_m$
Shift	$e_{x+}u_m = u_{m+1}$ (Forward) $e_{x-}u_m = u_{m-1}$ (Backward)
First Difference	$\delta_{x+}u_m = \Delta_x^{-1}(u_{m+1} - u_m)$ (Forward) $\delta_{x-}u_m = \Delta_x^{-1}(u_m - u_{m-1})$ (Backward) $\delta_{x\cdot}u_m = \frac{1}{2}\Delta_x^{-1}(u_{m+1} - u_{m-1})$ (Centred)
Second Difference (Laplacian)	$\delta_{xx}u_m = \Delta_x^{-2}(u_{m+1} - 2u_m + u_{m-1})$ (Centred)
Third Difference	$\delta_{xxx+}u_m = \Delta_x^{-3}(u_{m+2} - 3u_{m+1} + 3u_m - u_{m-1})$ (Forward) $\delta_{xxx-}u_m = \Delta_x^{-3}(u_{m+1} - 3u_m + 3u_{m-1} - u_{m-2})$ (Backward) $\delta_{xxx\cdot}u_m = \frac{1}{2}\Delta_x^{-3}(u_{m+2} - 2u_{m+1} + 2u_{m-1} - u_{m-2})$ (Centred)
Fourth Difference (Biharmonic)	$\delta_{xxxx}u_m = \Delta_x^{-4}(u_{m+2} - 4u_{m+1} + 6u_m - 4u_{m-1} + u_{m-2})$ (Centred)

Table 2. Summary of the spatial finite difference operators acting on a grid function u_m defined on a uniform grid with grid spacing Δ_x .

“true” solution of a differential problem at the grid locations. It will be convient to denote grid functions using indices, such that $u_m := (\mathbf{u})_m$ (the m -th component of the vector \mathbf{u}) \square

3.1 Finite difference operators acting on grid functions

The difference operators defined above can be applied conveniently to grid functions. Though the formal definition of the operators changes slightly when applied to a grid function u_m as opposed to a continuous function $u(x)$, the interpretation is straightforward: the differences are now computed as differences of the *elements* of the vector \mathbf{u} . Table 2 reports the definitions of the spatial difference operators acting on the grid function u_m .

Definition 3.4. The finite difference method is used to solve a differential

problem in $u(x)$ by computing a grid function \mathbf{u} such that:

$$u_m = u(x_m) + \mathcal{O}(\Delta_x^q), \quad (31)$$

where q is the *order of convergence* of the method \square

This definition implies that, as the grid spacing decreases towards zero, the grid function approximates the true solution exactly at the grid points. The definition (31) above is not entirely satisfying, as it implies a local convergence of one specific point in the domain. Refining a grid by making Δ_x smaller defines a sequence of grids \mathfrak{I}^n . The problem here is that the same grid point may not be present in all grids in the sequence, making the definition (31) impractical or ill-defined. It is more convenient, then, to define a form of global error over the whole grid, such as the *root mean square deviation*:

$$\text{RMSD}(\Delta_x) := \sqrt{\frac{\Delta_x}{L} \sum_{m=1}^{M-1} (u_m - u(x_m))^2}, \quad (32)$$

for which one also has $\text{RMSD}(\Delta_x) = \mathcal{O}(\Delta_x^q)$ when (31) holds.

The order of convergence q is closely related to the order of accuracy of the difference operators p , as defined in (9), though the two have different definitions and meanings. Solving a differential problem using p -th accurate difference operators often yields a p -th convergent finite difference scheme. In some cases, particularly when solving the model problem over non-uniform grids, the local truncation error may be as large as first-order for a given Δ_x , yet the RMSD has a higher convergence rate. This happens because the formal low order of accuracy is compensated by a rather small error at certain grid locations where the interval between grid points becomes much smaller than Δ_x . Determining the conditions for which the approximation (31) holds is the central problem of the finite difference method.

4 Boundary Value Problems

The techniques illustrated in the preceding sections allow for investigating several problems of interest in acoustics. We shall begin from simple *boundary value problems*. These problems do not depend on time and form a subclass of the model problems encountered in acoustics, which usually involve spatial as well as temporal differential operators. The model equations are, thus, *partial differential equations*. Boundary value problems emerge as special cases when, for instance, the steady state of a system is considered, and the time dependence of the model problem may be simplified accordingly.

Definition 4.1. A boundary value problem (BVP) in one dimension is defined as:

$$f(u, u', u'', \dots, u^{(n)}) = w(x), \quad (33)$$

where $w(x)$ is a source term, and n is the order of the BVP. To be complete, the BVP is supplied with n *boundary conditions* specifying the value of u and/or its derivatives up to the order $n - 1$ at the domain's boundary \square

4.1 The one-dimensional Poisson equation

It may be useful to start from the simple case of the one-dimensional Poisson equation, which, whilst not modelling an acoustic phenomenon as such, is useful as a test case for the finite difference techniques described in the previous section. This equation models the steady state of a diffusion-type system with a time-independent source, a problem encountered in fluid dynamics such as in the heat equation.

Definition 4.2. The one-dimensional Poisson equation with Dirichlet boundary conditions is defined as:

$$u''(x) = w(x), \quad u(0) = u_0, \quad u(L) = u_L, \quad (34)$$

where $w(x)$ is a time-independent source, and u_0, u_L are constants \square

Clearly, an analytic solution can be obtained by directly integrating (34) twice and setting the two integration constants using the boundary conditions. The accuracy of the finite difference method can be assessed against such an analytic solution.

Example 4.3. The Poisson equation for $w(x) = \tanh^2(x) - 1$, and with $u_0 = u_L = 0$ is solved via direct integration. In this case, the solution is

$$u(x) = L^{-1}x \log(\cosh(L)) - \log(\cosh(x)), \quad (35)$$

as one can verify after differentiating the solution twice \square

Solving the Poisson equation involves approximating the Laplace operator. Here, approximations will be constructed on the three types of grids presented in Section 3 that is, the uniform grid, the non-uniform smooth grid and the non-uniform non-smooth grid. In all three cases, the Laplacian will be approximated using:

$$u''(x_m) \approx \frac{u(x_{m-1})}{\Delta_{m-1/2}\tilde{\Delta}_m} - \frac{2u(x_m)}{\Delta_{m-1/2}\Delta_{m+1/2}} + \frac{u(x_{m+1})}{\Delta_{m+1/2}\tilde{\Delta}_m} \quad (36)$$

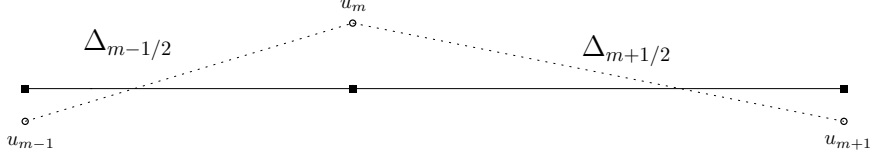


Figure 5. Three-point non-uniform mesh for the approximation of the Laplace operator.

where $\tilde{\Delta}_m := (\Delta_{m-1/2} + \Delta_{m+1/2})/2$. The approximation (36) may be obtained using the Vandermonde or Lagrange polynomial method, as detailed in Section 2.2.

Assume to generate a mesh using $M + 1$ (generally unequally spaced) sample points, as per (29). The boundary grid points u_0 and u_M need not be stored or updated since the boundary conditions fix their value. The Poisson equation defined by (34) is thus discretised as:

$$\frac{-2u_1}{\Delta_{1/2}\Delta_{3/2}} + \frac{u_2}{\Delta_{3/2}\tilde{\Delta}_1} = w_1, \quad (37a)$$

$$\frac{u_1}{\Delta_{3/2}\tilde{\Delta}_2} - \frac{2u_2}{\Delta_{3/2}\Delta_{5/2}} + \frac{u_3}{\Delta_{5/2}\tilde{\Delta}_2} = w_2, \quad (37b)$$

$$\dots + \dots = \dots$$

$$\frac{-2u_{M-1}}{\Delta_{M-3/2}\Delta_{M-1/2}} + \frac{u_{M-2}}{\Delta_{M-3/2}\tilde{\Delta}_{M-1}} = w_{M-1}, \quad (37c)$$

which may be cast more compactly as the $M - 1 \times M - 1$ system:

$$\mathbf{D}_{xx} \mathbf{u} = \mathbf{w} \quad (38)$$

with:

$$\mathbf{D}_{xx} := \begin{bmatrix} -\frac{2}{\Delta_{1/2}\Delta_{3/2}} & \frac{1}{\Delta_{3/2}\tilde{\Delta}_1} & 0 & \dots & 0 \\ \frac{1}{\Delta_{3/2}\tilde{\Delta}_2} & -\frac{2}{\Delta_{3/2}\Delta_{5/2}} & \frac{1}{\Delta_{5/2}\tilde{\Delta}_2} & 0 & \vdots \\ 0 & \frac{1}{\Delta_{5/2}\tilde{\Delta}_3} & -\frac{2}{\Delta_{5/2}\Delta_{7/2}} & \frac{1}{\Delta_{7/2}\tilde{\Delta}_3} & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \frac{1}{\Delta_{M-3/2}\tilde{\Delta}_{M-1}} & -\frac{2}{\Delta_{M-1/2}\Delta_{M-3/2}} \end{bmatrix}$$

System (38) is tridiagonal: note that the rows of the matrix are linearly independent, and hence, the matrix is invertible, and a unique solution exists. One slightly worrying aspect is the absence of symmetry of the matrix for all but the special case of the uniform grid ($\Delta_{m-1/2} = \Delta_x \forall m$). This, however, may be remedied via a similarity transformation.

Definition 4.4. Two square matrices \mathbf{A}, \mathbf{B} are called *similar* if there exist an invertible matrix \mathbf{S} such that:

$$\mathbf{B} = \mathbf{S}\mathbf{A}\mathbf{S}^{-1} \quad (39)$$

□

Theorem 4.5. *Two similar matrices \mathbf{A} and \mathbf{B} share the same eigenvalues. Furthermore, if (39) holds and \mathbf{a} is an eigenvector of \mathbf{A} with eigenvalue λ , then $\mathbf{b} := \mathbf{S}\mathbf{a}$ is an eigenvector of \mathbf{B} with eigenvalue λ .*

The proof is immediate. By definition $\mathbf{A}\mathbf{a} = \lambda\mathbf{a}$. From here $\mathbf{S}\mathbf{A}\mathbf{S}^{-1}\mathbf{S}\mathbf{a} = \lambda\mathbf{S}\mathbf{a}$, from which the proof follows □

Example 4.6. The matrix \mathbf{D}_{xx} is symmetrised using the similarity transformation $\mathbf{D}_{xx}^{sym} = \mathbf{S}\mathbf{D}_{xx}\mathbf{S}^{-1}$. To that end, consider the diagonal matrix:

$$[\mathbf{S}]_{m,m} := \prod_{j=2}^m \sqrt{\frac{\tilde{\Delta}_j}{\tilde{\Delta}_{j-1}}}, \quad m = 2, \dots, M-1, \quad (40)$$

and $[\mathbf{S}]_{1,1} = 1$. Then:

$$\mathbf{D}_{xx}^{sym} := \begin{bmatrix} -\frac{2}{\Delta_{\frac{1}{2}}\Delta_{\frac{3}{2}}} & \frac{1}{\Delta_{\frac{3}{2}}\sqrt{\tilde{\Delta}_1\tilde{\Delta}_2}} & 0 & \dots \\ \frac{1}{\Delta_{\frac{3}{2}}\sqrt{\tilde{\Delta}_1\tilde{\Delta}_2}} & -\frac{2}{\Delta_{\frac{3}{2}}\Delta_{\frac{5}{2}}} & \frac{1}{\Delta_{\frac{5}{2}}\sqrt{\tilde{\Delta}_2\tilde{\Delta}_3}} & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & \frac{1}{\Delta_{M-\frac{3}{2}}\sqrt{\tilde{\Delta}_{M-1}\tilde{\Delta}_{M-2}}} & -\frac{2}{\Delta_{M-\frac{1}{2}}\Delta_{M-\frac{3}{2}}} \end{bmatrix}$$

Symmetry guarantees that the eigenvalues are all real. In this special case, they are also positive, as one may show via the Gershgorin circle theorem, see e.g. Horn and Johnson (2013) □

The numerical solutions on the three grids are plotted in Figure 6. A cursory inspection allows appreciating the qualitatively similar solutions computed on the three grids. The solution computed on the random grid

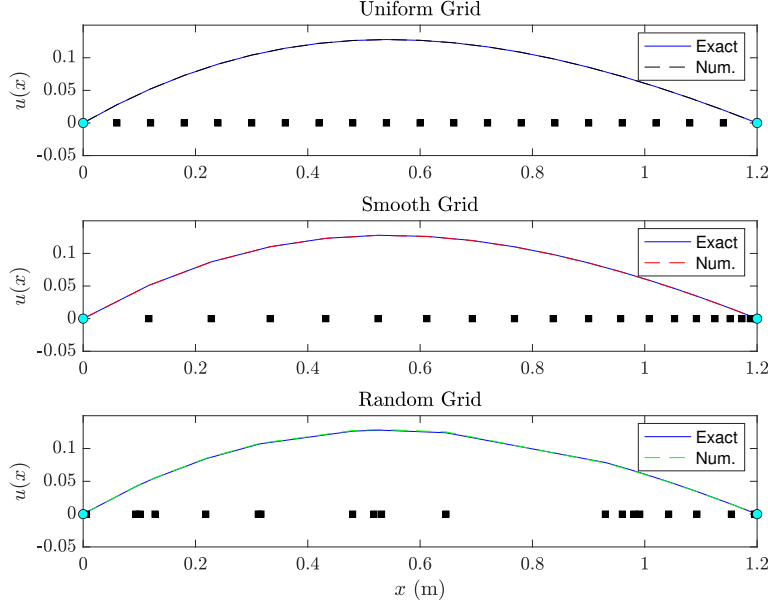


Figure 6. Numerical solutions of the Poisson equation (34) with $w(x) = \tanh^2(x) - 1$, using $L = 1.2$ and $M = 20$. The analytic solution (35) is also plotted at the grid points for reference.

presents a somewhat less smooth behaviour when the interpolating spline is plotted through the solution points, but this is a consequence of the uneven distribution of the sample points rather than a degradation in the quality of the numerical approximation. This claim can be made more precise by plotting the RMSD defined in (32) as a function of $\Delta_x := L/M$, i.e. the grid spacing of the uniform grid. Such grid spacing can be used as a reference to compute the bounding constants C_{\min} , C_{\max} in (30), so that a cell's size never exceeds those bounds. Figure 7 plots the RMSD as a function of Δ_x . The uniform grid presents the smallest error of all three, followed by the smooth grid. However, the error behaviour is consistent for all three cases. In particular:

$$\text{RMSD}(\Delta_x) \approx \mathcal{O}(\Delta_x^2), \quad (41)$$

and, hence, the solutions computed on the three grids are second-order *convergent*. This is an important result since the local truncation error of

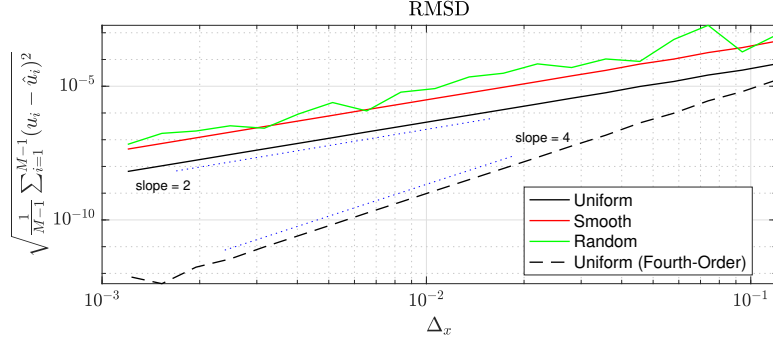


Figure 7. Root mean squared deviations (RMSD) as a function of $\Delta_x := L/M$. Blue dotted lines with slopes of two and four are also plotted for reference.

the difference operator (36) is, in theory, only first-order in Δ_x for the non-uniform grids! But when the errors are computed globally, they compensate. This example also shows that one is better off using a uniform grid in this case, since not only \mathbf{D}_{xx} is symmetric, but also Toeplitz. More importantly, the isotropy of the model problem (34) justifies the use of the uniform mesh on a physical basis. This makes the second-order convergence of the solutions computed on the non-uniform grids all the more remarkable. Using non-uniform grids is beneficial in simulating systems with varying spatial properties, as will be shown in later sections. Always in Figure 7, one sees the error trend of a solution computed on a uniform grid with a fourth-order accurate approximation of the Laplacian for interior points away from the boundaries, as per Example 2.10. Again, the expected error trend is recovered.

5 Eigenvalue Problems in One Dimension

Let us now turn the attention to *eigenvalue problems*. These crop up in various forms in the field of acoustics. Often, they originate after the temporal part of a model problem is transformed in the frequency domain via, e.g. a Fourier transform. Eigenvalue problems are a subclass of boundary value problems, as defined in (33), where f is linear, the source w is absent, and one is interested in finding the functions for which an action of the differential operator amounts to multiplying the function by a scalar.

Definition 5.1. For a linear spatial differential operator \mathcal{L} , the *eigenvalue*

problem (EVP) is a subclass of the BVP defined in (33) such that:

$$\mathcal{L}\hat{u} = \lambda\hat{u}, \quad (42)$$

where \hat{u} is called an *eigenfunction*, and λ is the corresponding *eigenvalue*. In most cases of interest in acoustics, the differential operator \mathcal{L} is such that its eigenvalues are amenable to the squared resonant frequencies of the system, $\lambda = \omega^2$. Generally, the eigenvalue problem is unsolvable analytically, and one resorts to spatial discretisation of the problem to obtain a numerical eigenvalue problem of the form:

$$\mathbf{L}\hat{\mathbf{u}} = \mathfrak{L}\hat{\mathbf{u}}, \quad (43)$$

where \mathbf{L} is a sparse matrix discretising the differential operator \mathcal{L} , $\hat{\mathbf{u}}$ is a corresponding eigenvector, and $\mathfrak{L} = \lambda + \mathcal{O}(\Delta_x^q)$. The simplest examples are drawn from the simple wave equation, now described \square

5.1 The wave equation

This simple equation describes a range of phenomena from tension waves in cables to longitudinal waves in elastic rods to acoustic pressure waves in tubes, taking the following forms:

$$\rho_c A \frac{\partial^2 u}{\partial t^2} = T_0 \frac{\partial^2 u}{\partial x^2}, \quad \rho_b \frac{\partial^2 v}{\partial t^2} = E \frac{\partial^2 v}{\partial x^2}, \quad \rho_a \frac{\partial^2 w}{\partial t^2} = K_a \frac{\partial^2 w}{\partial x^2}. \quad (44a)$$

In the above, u, v, w represent a cable's flexural displacement, a bar's longitudinal elongation, and air's longitudinal compression in a tube. ρ_c, ρ_b, ρ_a are the volume densities of the cable, the bar and the air, respectively; A is the cable's cross-section's area, T_0 is the applied tension; E is Young's modulus and K_a is the air's bulk modulus. From Figure 8, it results that the three equations can be written analogously as the one-dimensional wave equation with speed c .

Definition 5.2. The wave equation with speed c is given by:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}. \quad (45)$$

For tension waves in cables, u is a displacement, and we shall treat u as such in the forthcoming sections, knowing that the mathematics of the wave equation remains unchanged regardless of whichever physical system it describes \square

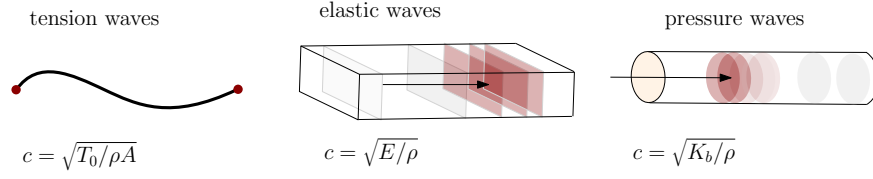


Figure 8. Examples of systems described by the one-dimensional wave equation: flexural waves in cables, longitudinal waves in bars, pressure waves in acoustics tubes.

Here, clearly, the state variable u is a function of both time and space, such that $u = u(t, x)$. It is convenient, for now, to consider temporal and spatial unbounded domains, such that $u = u(t, x) : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$. This is, of course, an abstraction since no physical system has existed from the beginning of time and is infinitely large, but this idealisation turns out to be useful for deriving a special relationship between the *frequency* and the *wavenumber* of a plane wave travelling through the system.

Definition 5.3. The *dispersion relation* for the wave equation is obtained as the relation $\omega = \omega(\gamma)$ after substituting the particular solution $u(t, x) = e^{j(\omega t + \gamma x)}$ (a plane wave) in (45), that is:

$$\omega = c\gamma, \quad (46)$$

where ω and γ are the radian temporal frequency and the wavenumber, respectively. These are real, positive numbers, i.e., $\omega, \gamma \in \mathbb{R}_0^+$. These are related to the period of the plane wave t and its wavelength l by:

$$t = 2\pi\omega^{-1}, \quad l = 2\pi\gamma^{-1} \quad (47)$$

□

Definition 5.4. The *phase velocity* c_ϕ and the *group velocity* c_g are defined via the dispersion relation, as:

$$c_\phi := \frac{\omega}{\gamma}, \quad c_g := \frac{d\omega}{d\gamma}. \quad (48)$$

For the simple wave equation, both velocities are equal to c and, hence, independent of frequency. The wave equation is a *dispersionless* system □

To derive an eigenvalue problem, the wave equation must be supplied with appropriate boundary conditions. Let now the spatial domain be the

closed, bounded interval $\mathcal{I} = \{x \mid x \in [0, L]\}$. Boundary conditions emerge naturally after an inspection of the energy of the continuous system (45).

Definition 5.5. For the wave equation, the total energy H is the sum of the kinetic and potential energies, defined as:

$$H_k := \frac{1}{2} \int_{\mathcal{I}} \left(\frac{\partial u}{\partial t} \right)^2 dx, \quad H_p := \frac{c^2}{2} \int_{\mathcal{I}} \left(\frac{\partial u}{\partial x} \right)^2 dx, \quad (49)$$

where the energies are here scaled by linear mass density in the case of tension wave in cables, and by the bar's and air's densities in the case of compressional elastic waves in bars and acoustic tubes \square

Taking the time derivative of the total energy $H = H_k + H_p$ and rearranging terms, one gets:

$$\int_{\mathcal{I}} \left(\frac{\partial u}{\partial t} \right) \left(\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} \right) dx = -c^2 \frac{\partial u}{\partial t} \frac{\partial u}{\partial x} \Big|_0^L. \quad (50)$$

The left-hand side now expresses the weak form of the equation of motion. The boundary conditions are recovered by nullifying the right-hand side, i.e. when $\partial u / \partial t$ (Dirichlet condition) or $\partial u / \partial x$ (Neumann condition) vanishes at the boundary. Often, a Dirichlet condition is referred to as *fixed* (since u is constant at the boundary), and a Neumann condition as *free*. The latter expression refers to the fact that the boundary is *free of loads* and is, thus, able to move.

Definition 5.6. The *eigenvalue problem* for the wave equation is obtained after substituting a time-transformed solution $u(t, x) = \hat{u}(x)e^{j\omega t}$ in the wave equation (45) defined over the closed, bounded interval \mathcal{I} , and imposing either a Dirichlet or a Neumann condition at $x = 0, L$. That is:

$$-c^2 \hat{u}''(x) = \omega^2 \hat{u}(x), \quad (51)$$

with either \hat{u} or \hat{u}' specified at the boundaries $x = \{0, L\}$. Considering the general definition of the eigenvalue problem (42), here $\mathcal{L} = -c^2 \frac{\partial^2}{\partial x^2}$, and hence the eigenvalues of the wave equation are real and positive \square

A larger class of boundary conditions is available besides the Dirichlet and Neumann types. The latter, as shown, have the interpretation of realising a form of energy conservation for the wave equation. In some cases, particularly in acoustics, the boundary may itself store or dissipate energy. The boundary condition is often expressed as an *impedance* in such cases.

For instance, the bridge of a guitar is not entirely fixed, and its mobility changes with frequency. Such more complicated boundary expressions will be considered here. See e.g. Maestre et al. (2017) for an example of impedance conditions in violin string vibration.

Solutions to the second-order differential problem (51) are obtained again through an appeal to complex exponentials. Using $\omega = c\gamma$ as per (46), consider the following particular solution:

$$\hat{u}(x) = A_+ e^{j\gamma x} + A_- e^{-j\gamma x}, \quad (52)$$

for constant A_{\pm} . When substituted in (51), one obtains an identity and therefore \hat{u} solves the eigenvalue problem. In general, $A_{\pm} \in \mathbb{C}$, but imposing the boundary conditions of Dirichlet or Neumann kind yields a set of real eigenfunctions. To show this, use $\hat{u}(0) = \hat{u}(L) = 0$ for Dirichlet, and $\hat{u}'(0) = \hat{u}'(L) = 0$ for Neumann. Two systems result:

$$\text{Dir: } \begin{bmatrix} 1 & 1 \\ e^{j\gamma L} & e^{-j\gamma L} \end{bmatrix} \begin{bmatrix} A_+ \\ A_- \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \text{Neu: } \begin{bmatrix} 1 & -1 \\ e^{j\gamma L} & -e^{-j\gamma L} \end{bmatrix} \begin{bmatrix} A_+ \\ A_- \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Non-trivial solutions are obtained by imposing the determinant of the matrices to be null, implying in both cases:

$$\sin \gamma L = 0 \implies \gamma = \gamma^{(p)} := p\pi/L, \quad p \in \mathbb{N}. \quad (53)$$

The wavenumbers are now quantised by the positive integer p . Turning to the eigenfunctions, one may use the first row for both systems above, giving $A_+ = -A_-$ for the Dirichlet case and $A_+ = A_-$ for Neumann. Using these in (52), one gets:

$$\text{Dir: } \hat{u}^{(p)}(x) = A^{(p)} \sin(\gamma^{(p)} x), \quad \text{Neu: } \hat{u}^{(p)}(x) = A^{(p)} \cos(\gamma^{(p)} x), \quad (54)$$

where the constant of proportionality may be chosen to normalise the eigenfunctions in some manner. It is customary to choose:

$$\text{Dir: } A^{(p)} = \left(\int_{\mathcal{I}} \sin^2(\gamma^{(p)} x) dx \right)^{-1/2}, \quad \text{Neu: } A^{(p)} = \left(\int_{\mathcal{I}} \cos^2(\gamma^{(p)} x) dx \right)^{-1/2}$$

and note that, for the Dirichlet case, $p = 1, 2, 3, \dots$ whereas for Neumann p may be zero, i.e. $p = 0, 1, 2, \dots$. Under such choice for the normalisation constants, one has $\int_{\mathcal{I}} (\hat{u}^{(p)})^2 dx = 1$, that is, the eigenfunctions are *normalised*. Note as well that, regardless of the constant of normalisation, $\int_{\mathcal{I}} \hat{u}^{(p)} \hat{u}^{(q)} dx = 0 \quad \forall p \neq q$, that is, the eigenfunctions are also *orthogonal*.

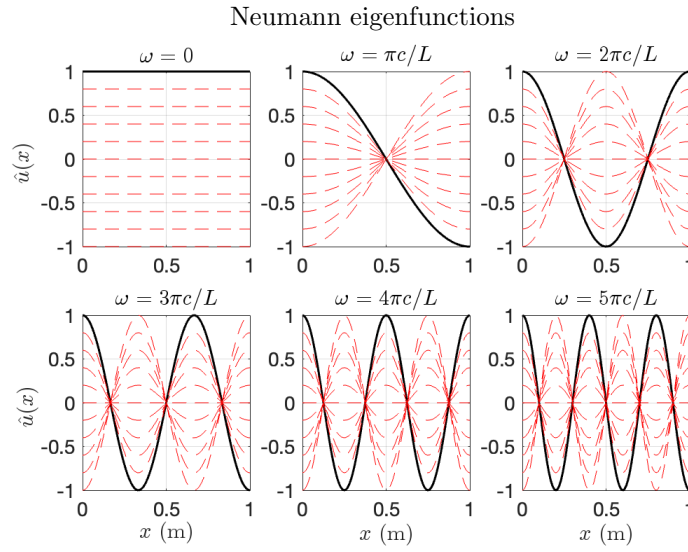
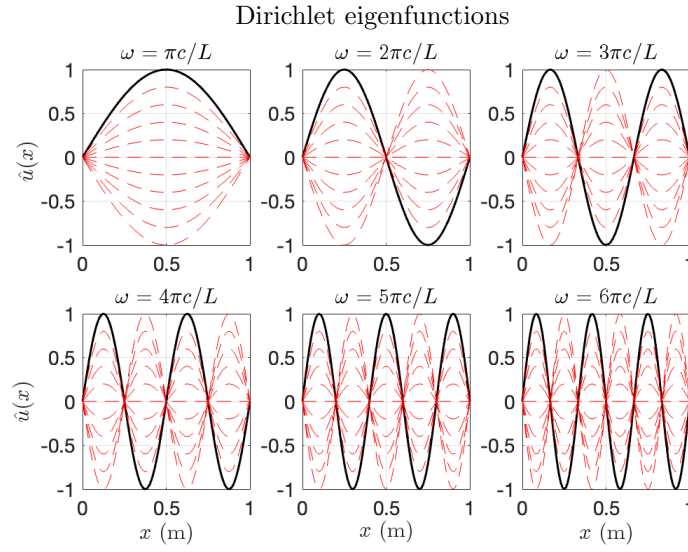


Figure 9. First six eigenfunctions for the eigenvalue problem (51) under a choice of Dirichlet or Neumann boundary conditions, as given.

The discrete eigenvalue problem A discrete eigenvalue problem is achieved simply by discretising the second derivative over a grid. It is convenient to begin the discussion using the uniform grid. Thus, consider a number $M := L/\Delta_x$ of grid subintervals, all of length Δ_x , yielding $M + 1$ grid points, including the endpoints.

Then, apply the definition of the Laplacian from Table 2 to the grid function $e^{j\gamma m\Delta_x}$:

$$\delta_{xx}e^{j\gamma m\Delta_x} = \frac{e^{j\gamma\Delta_x} - 2 + e^{-j\gamma\Delta_x}}{\Delta_x^2}e^{j\gamma m\Delta_x}$$

which, using the half-angle trigonometric formulae, can be written as:

$$\delta_{xx}e^{j\gamma m\Delta_x} = -\frac{4}{\Delta_x^2} \sin^2 \frac{\gamma\Delta_x}{2} e^{j\gamma m\Delta_x} = -\gamma^2 \operatorname{sinc}^2 \frac{\gamma\Delta_x}{2} e^{j\gamma m\Delta_x},$$

with $\operatorname{sinc}(x) := x^{-1} \sin(x)$. This shows that the complex exponential $e^{j\gamma m\Delta_x}$ is an eigenfunction of the discrete operator δ_{xx} , with eigenvalue $-\gamma^2 \operatorname{sinc}^2 \frac{\gamma\Delta_x}{2} = -\gamma^2 + \mathcal{O}(\Delta_x^2)$. The same eigenvalue is returned by applying δ_{xx} to $e^{-j\gamma m\Delta_x}$. Note that the function $\sin^2(x)$ is periodic with period π . Considering the positive half of the range of $[-\pi/2, \pi/2]$ (of length π), the useful range for γ becomes the following:

$$0 \leq \gamma \leq \pi/\Delta_x. \quad (55)$$

Notice that the sinc function “warps” the eigenvalues of the discrete wave equation compared to the eigenvalues of the continuous wave equation, and the effect is more and more pronounced γ approaches the upper limit in (55).

Definition 5.7. The *discrete eigenvalue problem* for the wave equation defined on a uniform grid with grid size Δ_x is given by:

$$-c^2\delta_{xx}\hat{u}_m = \omega^2 \operatorname{sinc}^2 \frac{\omega\Delta_x}{2c} \hat{u}_m, \quad (56)$$

where $\omega = c\gamma$ has the interpretation of an eigenfrequency of the continuous eigenvalue problem \square

The eigenfunctions are given in this case by:

$$\hat{u}_m = A_+ e^{j\gamma m\Delta_x} + A_- e^{-j\gamma m\Delta_x}. \quad (57)$$

As per the continuous case, the discrete eigenvalue problem (56) must be supplied with appropriate boundary conditions. Such conditions are given here as $\hat{u}_0 = \hat{u}_M = 0$ in the Dirichlet case, and $\delta_x.\hat{u}_0 = \delta_x.\hat{u}_M = 0$ in the

Neumann case. Substituting such conditions in the expression (57), two systems result:

$$\text{Dir: } \begin{bmatrix} 1 & 1 \\ e^{j\gamma L} & e^{-j\gamma L} \end{bmatrix} \begin{bmatrix} A_+ \\ A_- \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{Neu: } \begin{bmatrix} 1 & -1 \\ e^{j\gamma L} & -e^{-j\gamma L} \end{bmatrix} \begin{bmatrix} A_+ \\ A_- \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

where now $L = M\Delta_x$. Remarkably, nullifying the determinants results in the same equation as for the continuous case, (53). Thus:

$$\gamma^{(p)} := p\pi/L = p\pi/M\Delta_x \quad (58)$$

are the quantised solutions of the discrete eigenvalue problem. Solving for the eigenfunctions, similarly to what was done to obtain (54), one gets:

$$\text{Dir: } \hat{u}_m^{(p)}(x) = A^{(p)} \sin(\gamma^{(p)} m \Delta_x), \quad \text{Neu: } \hat{u}_m^{(p)}(x) = A^{(p)} \cos(\gamma^{(p)} m \Delta_x),$$

The constant of normalisation may, in this case, be chosen such that:

$$\text{Dir: } A^{(p)} = \left(\sum_{m=1}^{M-1} \sin^2(\gamma^{(p)} m \Delta_x) \right)^{-\frac{1}{2}}, \quad \text{Neu: } A^{(p)} = \left(\sum_{m=0}^M \cos^2(\gamma^{(p)} m \Delta_x) \right)^{-\frac{1}{2}}.$$

It may be useful in some cases to write the discrete eigenvalue problem (56) in matrix form. This is:

$$-c^2 \mathbf{D}_{xx} \hat{\mathbf{u}} = \omega^2 \text{sinc}^2 \frac{\omega \Delta_x}{2c} \hat{\mathbf{u}}, \quad (59)$$

where the explicit form of the Laplacian under Dirichlet or Neumann conditions are:

$$\mathbf{D}_{xx}^{\text{Dir}} = \frac{1}{\Delta_x^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix} \in \mathbb{R}^{M-1 \times M-1},$$

$$\mathbf{D}_{xx}^{\text{Neu}} = \frac{1}{\Delta_x^2} \begin{bmatrix} -2 & 2 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 2 & -2 \end{bmatrix} \in \mathbb{R}^{M+1 \times M+1}.$$

Note that the Dirichlet matrix has smaller dimensions: in this case, the end-points u_0, u_M need not be stored or updated since the boundary conditions

fix their value to zero at all times. Note also that the Neumann matrix is not symmetric, but its eigenvalues are real, as shown. The matrix may be diagonalised via a similarity transformation (see Definition 4.3).

Regardless, the wave equation operator admits the following *eigenvalue decomposition*:

$$\mathbf{L} := -c^2 \mathbf{D}_{xx} = \hat{\mathbf{U}} \mathbf{\Omega}^2 \hat{\mathbf{U}}^{-1}, \quad (60)$$

where $\mathbf{\Omega}$ is diagonal and it contains the eigenvalues $(\omega^{(p)})$ since $\frac{\omega^{(p)} \Delta_x}{2c}$, where $\omega^{(p)} = c\gamma^{(p)}$ is obtained from (58), and where $\hat{\mathbf{U}}$ is a matrix whose columns are the eigenvectors, i.e.

$$\hat{\mathbf{U}} := [\hat{\mathbf{u}}^{(1)}, \hat{\mathbf{u}}^{(2)}, \dots]. \quad (61)$$

According to the boundary conditions, the eigenvectors $\hat{\mathbf{u}}^{(p)}$ are either the sampled sines or cosines. The form of $\hat{\mathbf{U}}$ is special for the Dirichlet case: the Laplacian is symmetric, meaning that $\hat{\mathbf{U}}$ is *orthonormal*. In practice, the inverse of $\hat{\mathbf{U}}$ in (60) is the transpose, and via the orthonormality property one has $\hat{\mathbf{U}} \hat{\mathbf{U}}^\top = \hat{\mathbf{U}}^\top \hat{\mathbf{U}} = \mathbf{I}$. The orthonormality condition holds only if the normalisation constants $A^{(p)}$ are set as suggested above. If the eigenvectors are normalised differently, $\hat{\mathbf{U}}$ remains *orthogonal*, meaning that multiplication by its transpose gives rise to a diagonal matrix whose diagonal elements correspond to the squared norms of the sine functions. Figure 10 reports the first few eigenvectors of the Laplacian matrix for the two cases.

5.2 The Euler-Bernoulli equation for rods

A qualitatively different kind of wave propagation is displayed by flexural motion in bars. Bars in bending are subjected to elastic forces due to the shearing of the cross-section. When a bar is bent, a bending moment is generated due to the deformation of the cross-section, resulting in a flexural elastic force.

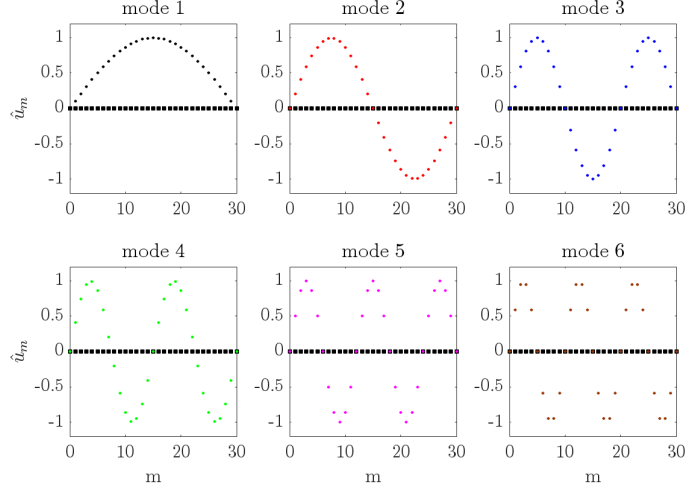
Definition 5.8. The *Euler-Bernoulli* equation, describing flexural waves in bars, is given by:

$$\rho A \frac{\partial^2 u}{\partial t^2} = - \frac{\partial^2 \mu}{\partial x^2}, \quad \mu = EI \frac{\partial^2 u}{\partial x^2} \quad (62)$$

□

Here, $u = u(t, x)$ is the vertical displacement of an element of the bar, and $\mu = \mu(t, x)$ is the bending moment originating from the internal stresses when the bar bends. Constants appear as: ρ , the volume density of the bar,

Dirichlet Finite Difference eigenfunctions



Neumann Finite Difference eigenfunctions

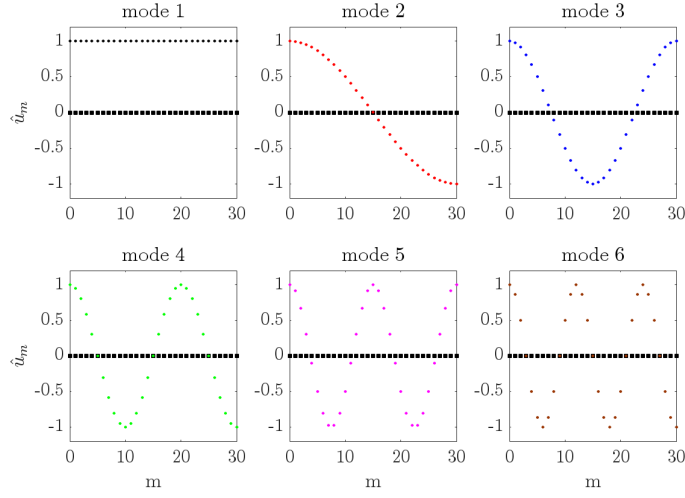


Figure 10. First six eigenfunctions for the eigenvalue problem (56) under a choice of Dirichlet or Neumann boundary conditions, as given, for $M = 30$.

and E , Young's modulus. In the following, $A = A(x)$ is the area of the cross-section, allowed here to vary; $I = I(x)$ is the corresponding area moment of inertia, defined as:

$$I(x) = \int_{A(x)} \zeta^2 dA, \quad (63)$$

In the above, ζ is the distance of the area element dA from the axis of rotation. For a circular cross-section of radius r , $I = \pi r^4/4$, whereas for a rectangular cross-section of width b and height a , $I = a^3b/12$. Notice that both A and I are solely geometric parameters with no dependence on the bar's material parameters. Bars with a spatially varying cross-section are ubiquitous in musical acoustics and found across a variety of percussion instruments such as xylophones and marimbas. Before proceeding with discretising (62), it is worth studying the behaviour of the elastic waves in the continuous case, as this allows selecting the mesh size according to wave speed considerations.

The dispersion relation for system (62) is obtained after substituting $u(x, t) = G(x)e^{j\omega t}e^{j\gamma x}$. As per (47), γ and ω represent a plane wave's spatial and temporal frequency. Here, however, a spatially dependent wave amplitude $G(x)$ is assumed, accounting for the geometric variation of the cross-section. In a neighbourhood of x , the amplitude factor can be assumed constant and one has:

$$\gamma \approx \left(\frac{\rho A(x)}{EI(x)} \right)^{\frac{1}{4}} \omega^{\frac{1}{2}}, \quad (64)$$

The phase and group velocities for the bar with a varying cross-section are obtained from (64), as:

$$c_\phi(x, \omega) \approx \left(\frac{EI(x)}{\rho A(x)} \right)^{\frac{1}{4}} \omega^{\frac{1}{2}}, \quad c_g(x, \omega) \approx 2c_\phi(x, \omega), \quad (65)$$

and, hence, the bar is a *dispersive* system since the phase velocity depends on frequency. As a consequence, an initial wavefront travelling across the bar will gradually distort and flatten out as shorter wavelengths travel faster. The bar with varying cross-sections is further characterised by the phase and group velocities' spatial dependence through $A(x)$ and $I(x)$.

Example 5.9. A marimba's typical cross-section is considered here, as per Fig. 11. A rectangular cross-section is assumed throughout, for which:

$$I = \frac{a^3b}{12}, \quad \frac{dI}{dx} = \frac{a^2b}{4} \frac{da}{dx}, \quad \frac{d^2I}{dx^2} = \frac{ab}{2} \left(\frac{da}{dx} \right)^2 + \frac{a^2b}{4} \frac{d^2a}{dx^2}, \quad (66)$$



Figure 11. Picture of the Majestic 5.0 Octave Rosewood Bar Concert Marimba. Copyright: Majestic Percussion. [Link to video.](#)

where $a = a(x)$ is the thickness of the cross-section, and b is the width. Assume that the thickness varies as a power law:

$$a = a^{(min)} + \frac{2^p \Delta a}{l^p} |x|^p, \quad (-l/2 \leq x \leq l/2), \quad (67)$$

with $\Delta a := a^{(max)} - a^{(min)}$. In the following, take $\Delta a = 3$ cm, $l = 40$ cm, $a^{(min)} = 1$ cm, $b = 6$ cm, and $p \in \{1, 2, 3, 4\}$. Results are summarised in Fig. 12 where it is seen that the moment of inertia and its derivatives are small. Note, however, terms such as $\frac{1}{l} \frac{dI}{dx}$ are not as small \square

Definition 5.10. The energy H of a bent bar is obtained as the sum of its kinetic and potential energies H_k , H_p , defined as follows:

$$H_k := \int_{\mathcal{I}} \frac{\rho A(x)}{2} \left(\frac{\partial u}{\partial t} \right)^2 dx, \quad H_p := \frac{1}{2} \int_{\mathcal{I}} \frac{\partial^2 u}{\partial x^2} \mu(x, t) dx, \quad (68)$$

where the definition of the moment $\mu(x, t)$ is given in (62), and where $\mathcal{I} = \{x | 0 \leq x \leq L\}$ \square

Taking the time derivative of $H = H_k + H_p$ one gets:

$$\int_{\mathcal{I}} \left(\frac{\rho A}{2} \frac{\partial^2 u}{\partial t^2} + \frac{\partial^2 \mu}{\partial x^2} \right) dx = \frac{\partial u}{\partial t} q - \frac{\partial^2 u}{\partial t \partial x} \mu \Big|_0^L, \quad (69)$$

where the beam's shear force was defined as $q := \frac{\partial \mu}{\partial x}$. Nullifying the integrand above results in the strong form of the equation of motion (62); the right-hand side returns the boundary conditions. Here, two such conditions

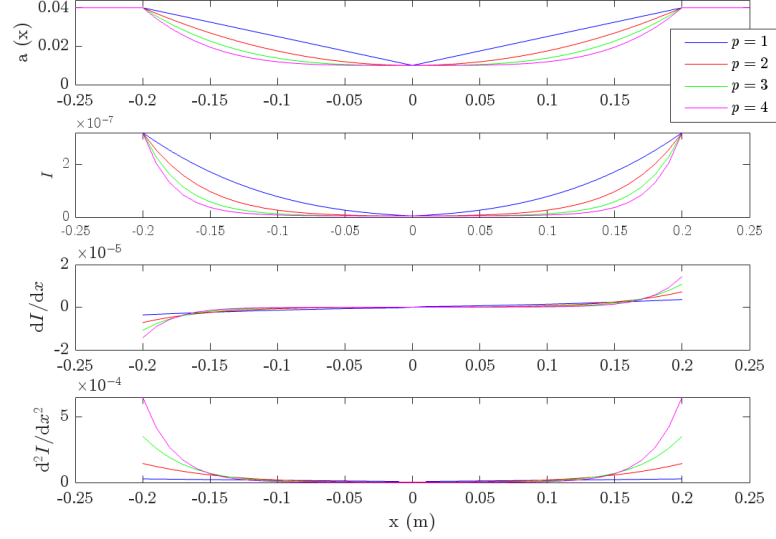


Figure 12. Thickness profiles and corresponding moment of inertia and derivatives. The thickness profile follows the power law (67), under four different values for the exponent p , as indicated. Here, $a^{(max)} = 4$ cm, $a^{(min)} = 1$ cm, $b = 6$ cm, $l = 40$ cm. Note that the moment of inertia and its derivatives are small, such that higher-order powers can safely be neglected when deriving results.

emerge at each end, a feature distinguishing the bar model compared to the simple wave equation. Conditions of classic type emerge as:

$$\text{free: } \mu = q = 0, \text{ clamped: } u = \frac{\partial u}{\partial x} = 0, \text{ simply-supported: } \mu = u = 0.$$

Free-end conditions correspond to the vanishing of the applied moment and its derivative, the net shear force; clamped conditions are purely geometrical with vanishing displacement and slope; simply-supported conditions are somewhat in between, having a null moment and displacement. Note that a fourth condition arises mathematically, namely $\partial u / \partial x = \partial \mu / \partial x = 0$, but this is rarely given in textbooks as it is seldom realised in practice.

The eigenvalue problem with constant thickness In the simplest case of bars have a constant cross-section, the Euler-Bernoulli equation sim-

plifies to:

$$\frac{\partial^2 u}{\partial t^2} = -\kappa^4 \frac{\partial^4 u}{\partial x^4}, \quad (70)$$

with $\kappa := (EI/\rho A)^{1/4}$. Let $\mathcal{I} := \{x|x \in [0, L]\}$ denote the domain occupied by the unstretched bar. The *eigenvalue problem* for the bar in bending is obtained after substituting the trial solution $u(x, t) = \hat{u}(x)e^{j\omega t}$, resulting in:

$$\kappa^4 \hat{u}''''(x) = \omega^2 \hat{u}(x), \quad (71)$$

This equation admits four distinct solutions in the form of exponentials, such that:

$$\hat{u}(x) = A_+ e^{j\gamma x} + A_- e^{-j\gamma x} + B_+ e^{\gamma x} + B_- e^{-\gamma x}, \quad (72)$$

with $\gamma^4 := \omega^2/\kappa^4$. The four constants A_+, A_-, B_+, B_- are determined by applying the boundary conditions at the left and right endpoints of the interval \mathcal{I} . Nullifying the determinant of the resulting system gives a transcendental equation whose solution returns the quantised wave numbers $\gamma^{(p)}$.

Example 5.11. The quantised wavenumbers for a bar simply-supported at both ends are obtained by imposing the boundary conditions $\hat{u} = d^2\hat{u}/dx^2 = 0$ at both ends. Imposing the condition at $x = 0$ results in $\hat{u} = 2jA_+ \sin(\gamma x) + 2B_+ \sinh(\gamma x)$. Further imposing the conditions at $x = L$ results in $0 = 2jA_+ \sin(\gamma L) + 2B_+ \sinh(\gamma L)$ and $0 = 2jA_+ \sin(\gamma L) - 2B_+ \sinh(\gamma L)$, which is only possible when $B_+ = 0$ and

$$\gamma^{(p)} = p\pi/L, \quad p \in \mathbb{N}. \quad (73)$$

Note that these are the same wavenumbers obtained for the simple wave equation, but now the corresponding frequencies are given by $\omega^{(p)} = (\gamma^{(p)})^2 \kappa^2$. The corresponding eigenfunctions are, thus:

$$\hat{u}^{(p)}(x) = A^{(p)} \sin(\gamma^{(p)} x), \quad (74)$$

again, the same as the eigenfunctions of the simple wave equations under Dirichlet conditions \square

Finding analytic expressions for the eigenvalues and frequencies becomes unwieldy for bars with a variable cross-section. When system (62) is transformed in time using a Fourier transform, as per Example 5.11, the following system results:

$$\omega^2 \rho A(x) \hat{u}(x) = \hat{\mu}''(x), \quad \hat{\mu}(x) = EI(x) \hat{u}''(x) \quad (75)$$

The problem can be approached using a discrete eigenvalue problem defined over a grid.

The discrete eigenvalue problem A formula analogous to (36) is used to approximate the second derivatives over a non-uniform grid. The *discrete eigenvalue problem* for the Euler-Bernoulli bar with non-uniform cross-section is derived from:

$$\omega^2 \rho A_m \hat{u}_m = \frac{\hat{\mu}_{m-1}}{\Delta_{m-1/2} \tilde{\Delta}_m} - \frac{2\hat{\mu}_m}{\Delta_{m-1/2} \Delta_{m+1/2}} + \frac{\hat{\mu}_{m+1}}{\Delta_{m+1/2} \tilde{\Delta}_m} \quad (76a)$$

$$(EI_m)^{-1} \hat{\mu}_m = \frac{\hat{u}_{m-1}}{\Delta_{m-1/2} \tilde{\Delta}_m} - \frac{2\hat{u}_m}{\Delta_{m-1/2} \Delta_{m+1/2}} \frac{\hat{u}_{m+1}}{\Delta_{m+1/2} \tilde{\Delta}_m}, \quad (76b)$$

where, as per Section 4.1, $\Delta_{m-1/2}$ is the grid spacing between grid points x_{m-1} and x_m , and where $\tilde{\Delta}_m := (\Delta_{m-1/2} + \Delta_{m+1/2})/2$. ω^2 is the eigenvalue associated with the eigenvector $\hat{\mathbf{u}}$. Here, $m \in [0, M] \subset \mathbb{N}$, defining the grid $\mathcal{J} = \{x_j | x_j = \sum_{m=1}^j \Delta_{m-1/2}\}$.

Numerical boundary conditions must be imposed at the bar's ends to account for the action of the difference operators on points near the boundary. Considering the marimba bar in Fig. 12, the cross-section is constant near the boundary, and the boundary conditions can be assumed to be of free type. Hence, the following are imposed:

$$\hat{\mu}_0 = 0, \quad \hat{\mu}_M = 0, \quad \hat{\mu}_{-1} = \hat{\mu}_1, \quad \hat{\mu}_{M+1} = \hat{\mu}_{M-1}, \quad (77)$$

discretising free-end conditions with constant bar cross-section near the boundary. Thus, $\hat{\mu}_0$ and $\hat{\mu}_M$ need not be stored or updated since their value is fixed at all times. Starting with (76a), and using the numerical conditions above, one has:

$$\begin{aligned} \omega^2 \rho A_0 \hat{u}_0 &= \frac{2\hat{\mu}_1}{\Delta_{1/2} \tilde{\Delta}_0} \\ \omega^2 \rho A_1 \hat{u}_1 &= -\frac{2\hat{\mu}_1}{\Delta_{1/2} \Delta_{3/2}} + \frac{\hat{\mu}_2}{\Delta_{3/2} \tilde{\Delta}_1} \\ \omega^2 \rho A_2 \hat{u}_2 &= \frac{\hat{\mu}_1}{\Delta_{3/2} \tilde{\Delta}_2} - \frac{2\hat{\mu}_2}{\Delta_{3/2} \Delta_{5/2}} + \frac{\hat{\mu}_3}{\Delta_{5/2} \tilde{\Delta}_2} \\ &\dots \end{aligned}$$

It is convenient to collect the coefficients multiplying the elements of the

vector $\hat{\boldsymbol{\mu}}$ in a matrix:

$$\mathbf{D}_{xx} := \begin{bmatrix} \frac{1}{\Delta_{\frac{1}{2}} \bar{\Delta}_{\frac{1}{2}}} & & & & 0 \\ -\frac{1}{\Delta_{\frac{1}{2}} \bar{\Delta}_{\frac{3}{2}}} & \frac{1}{\Delta_{\frac{3}{2}} \bar{\Delta}_1} & 0 & \dots & 0 \\ \frac{1}{\Delta_{\frac{3}{2}} \bar{\Delta}_2} & -\frac{2}{\Delta_{\frac{3}{2}} \bar{\Delta}_{\frac{5}{2}}} & \frac{1}{\Delta_{\frac{5}{2}} \bar{\Delta}_2} & 0 & \vdots \\ 0 & \frac{1}{\Delta_{\frac{5}{2}} \bar{\Delta}_3} & -\frac{2}{\Delta_{\frac{5}{2}} \bar{\Delta}_{\frac{7}{2}}} & \frac{1}{\Delta_{\frac{7}{2}} \bar{\Delta}_3} & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \frac{1}{\Delta_{M-\frac{3}{2}} \bar{\Delta}_{M-1}} & -\frac{2}{\Delta_{M-\frac{1}{2}} \bar{\Delta}_{M-\frac{3}{2}}} \\ 0 & & & & \frac{1}{\Delta_{M-\frac{1}{2}} \bar{\Delta}_{M-\frac{1}{2}}} \end{bmatrix},$$

defining a rectangular matrix of size $(M+1) \times (M-1)$. Collecting coefficients on the right hand side of (76b) produces the matrix:

$$\mathbf{D}_{xx}^* := \begin{bmatrix} \frac{1}{\Delta_{\frac{1}{2}} \bar{\Delta}_1} & -\frac{2}{\Delta_{\frac{1}{2}} \bar{\Delta}_{\frac{3}{2}}} & \frac{1}{\Delta_{\frac{3}{2}} \bar{\Delta}_1} & \dots & 0 & 0 \\ 0 & \frac{1}{\Delta_{\frac{3}{2}} \bar{\Delta}_2} & -\frac{2}{\Delta_{\frac{3}{2}} \bar{\Delta}_{\frac{5}{2}}} & \frac{1}{\Delta_{\frac{5}{2}} \bar{\Delta}_2} & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & \frac{1}{\Delta_{M-\frac{3}{2}} \bar{\Delta}_{M-1}} & -\frac{2}{\Delta_{M-\frac{1}{2}} \bar{\Delta}_{M-\frac{3}{2}}} & \frac{1}{\Delta_{M-\frac{1}{2}} \bar{\Delta}_{M-1}} \end{bmatrix},$$

of size $(M-1) \times (M+1)$. Before proceeding, note that in the case of a uniform grid $\Delta_{m-1/2} = \Delta_x \forall m$ one has $\mathbf{D}_{xx}^* = \mathbf{D}_{xx}^{\mathbf{I}}$.

Definition 5.12. The discrete eigenvalue problem for bars with varying cross-section is written compactly as:

$$\omega^2 \rho \mathbf{A} \hat{\mathbf{u}} = \mathbf{D}_{xx} \hat{\boldsymbol{\mu}}, \quad (E\mathbf{I})^{-1} \hat{\boldsymbol{\mu}} = \mathbf{D}_{xx}^* \hat{\mathbf{u}}, \quad (78)$$

where \mathbf{A} and \mathbf{I} are diagonal matrices storing the values of the area and moment of inertia of the cross-section. The system may be reduced further and be expressed in terms of $\hat{\mathbf{u}}$ alone, as:

$$\tilde{\kappa}^4 (\tilde{A} \mathbf{A}^{-1}) \mathbf{D}_{xx} (\tilde{I}^{-1} \mathbf{I}) \mathbf{D}_{xx}^* \hat{\mathbf{u}}(x) = \omega^2 \hat{\mathbf{u}}(x), \quad \text{with } \tilde{\kappa} := \left(\frac{E \tilde{I}}{\rho \tilde{A}} \right)^{\frac{1}{4}}, \quad (79)$$

and where \tilde{A} , \tilde{I} are scaling factors (such as the average value of the area and moment of inertia along the bar or their largest or smallest values) \square

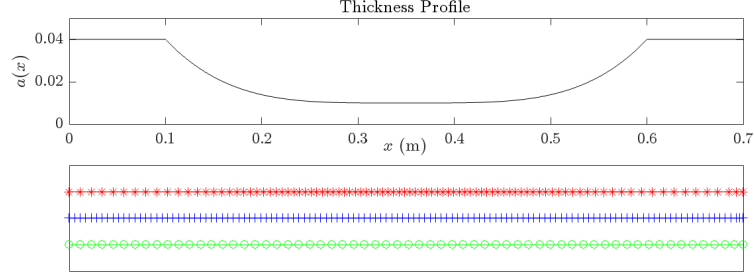


Figure 13. Top: thickness profile. Bottom: grid spacings using a coarse constant spacing (green); a fine constant spacing (blue); and adaptive spacing (red). The grid spacings are selected according to (80), using $\text{ppw} = 50$, and $\omega = 20000\pi$. For the constant grid spacings, the reference phase velocity is selected as the phase velocity near the bar’s ends (yielding the largest grid spacing) and at the bar’s centre (yielding the finest grid spacing). The adaptive mesh naturally maps the wave velocity to the corresponding grid spacing yielding a constant number of points per wavelength across the bar.

One obvious question is how to select an appropriate grid to solve the numerical eigenvalue problem.

Definition 5.13. The grid’s *points per wavelength* (ppw) at frequency ω are defined as:

$$\text{ppw} := \frac{2\pi c_\phi(x_m, \omega)}{\omega \tilde{\Delta}_m}, \quad (80)$$

where $c_\phi(x_m, \omega)$ is the phase velocity at frequency ω and grid point x_m , as defined in (65) \square

The idea of using “points-per-wavelength” as a control parameter for solving numerical eigenvalue problems is widespread in acoustics, particularly in room acoustics, as illustrated e.g. in Langer et al. (2017); Marburg (2002). In order to control the finesse of the grid for the purpose of computing the eigenfrequencies and shapes of the bar, one may select a largest frequency $\omega^{(max)}$ and an input ppw at such frequency, therefore yielding the mesh size $\tilde{\Delta}_m$. Fig. 13 reports the non-uniform mesh obtained by sampling the interval \mathcal{I} using definition (80). The same figure reports the grids obtained by sampling the interval using the smallest and largest grid spacing, corresponding to the largest and smallest phase velocity in the bar. Fig. 14 reports the convergence of the first three eigenfrequencies as a function of

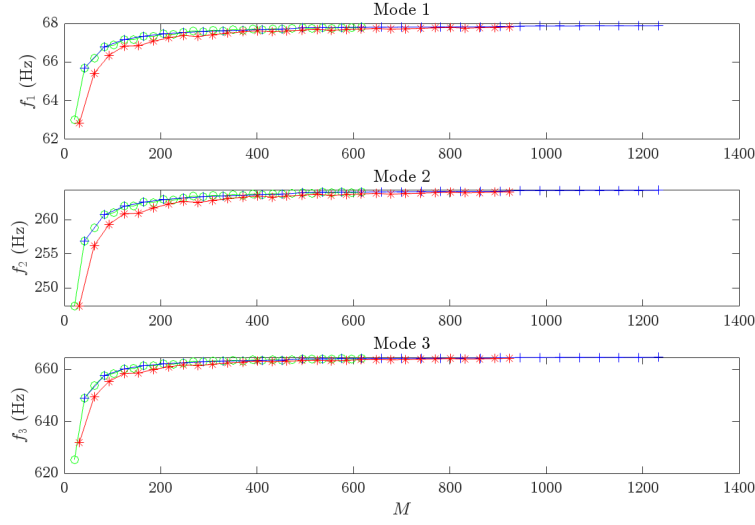


Figure 14. Convergence of the first three eigenfrequencies as a function of the number of grid subintervals M using a coarse constant spacing (green); a fine constant spacing (blue); and adaptive spacing (red). The three grid types yield a similar convergence profile.

M . As expected, when plotted as a function of M , the uniform grids yield the same eigenfrequency values $\forall M$. The eigenfrequencies computed on the non-uniform grid follow a similar convergence trend. Whilst these notes do not cover the time domain in detail, the use of non-uniform meshes is crucial in the design of efficient time stepping schemes, maximising the output signal's bandwidth at any given input sample rate, as shown elegantly by (Bilbao, 2009, Ch. 7).

Example 5.14. In the context of marimba making, the process of removing material from the centre of the bars is called *tuning* or, more specifically, *undercutting*, see Soares et al. (2021). This process involves carefully removing material from the underside of the wooden bars to achieve the desired pitch and harmonic overtones. The amount and location of the material removed affect the fundamental frequency and the relationship between the harmonics, allowing the instrument maker to fine-tune the bars for a resonant and balanced sound. Typically, the bar's centre influences the fundamental pitch, while adjustments near the nodal points can modify

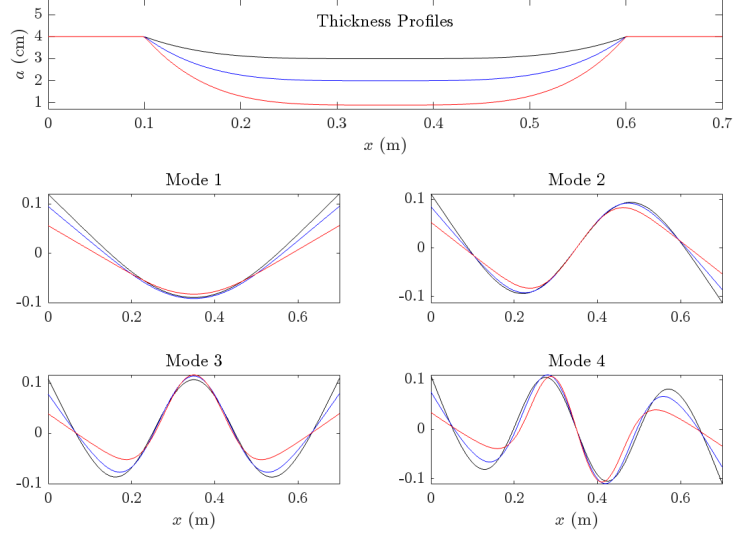


Figure 15. Bar profiles and first four eigenshapes. The thickness profiles follow the power law (67), under three different values for the smallest thickness $a^{(min)}$. Here, $a^{(max)} = 4$ cm, $p = 4$, $b = 6$ cm, $l = 40$ cm and $a^{(min)} = \{3$ cm (black), 2 cm (blue) and 0.9 cm (red) $\}$.

	f_2/f_1	f_3/f_1
$a^{(min)} = 3$ cm	3.0	6.2
$a^{(min)} = 2$ cm	3.4	7.5
$a^{(min)} = 0.9$ cm	4.0	10.2

Table 3. Frequency relationships for the three bars of Fig. 15.

the harmonic relationships. In Fig. 15, three bar profiles are considered, following (67) and presenting three different $a^{(min)}$ values. The deformation of the eigenshapes is evident, with nodes moving toward the centre as the smallest thickness is decreased. Such wavelength shortening is unsurprising in light of the increased wave speed near the bar's centre. The relationships of the first two overtones to the fundamentals are given in Table 3, highlighting the effects of undercutting \square

6 Eigenvalue Problems in Two Dimensions

Extensions to the two-dimensional case are straightforward. In the following, let $\hat{u} = \hat{u}(x, y) : \mathcal{U} \rightarrow \mathbb{R}$ be a smooth function defined over the connected domain $\mathcal{U} \subset \mathbb{R}^2$. In musical acoustics, two-dimensional instrument parts are ubiquitous, such as in percussion instruments (drum skins, cymbals), as well as soundboards and top and back plates. The eigenvalue problem definition (42) still holds for a linear operator $\mathcal{L} = \mathcal{L}(x, y)$ acting on functions defined over \mathcal{U} . As per the one-dimensional case, $\hat{u}(x, y)$ is usually recovered from a time-dependent differential problem after assuming a time-harmonic behaviour of the solution (i.e., by taking a Fourier transform).

6.1 The orthotropic Kirchhoff-Love plate equation

The Kirchhoff-Love plate theory provides a working model for thin plates and soundboards. Numerous sources provide detailed derivations of the equations of motion for plates; here, essential aspects are summarized for consistent notation. A comprehensive derivation is available in (Szilard, 2004, Ch. 1,4,10). The following foundational assumptions are adopted:

- The plate material has a uniform density ρ .
- The thickness ζ is constant and significantly smaller than the lateral dimensions.
- The medium exhibits orthotropic behaviour with two orthogonal principal axes.
- Sections perpendicular to the mid-surface remain plane during deformation, consistent with Kirchhoff-Love kinematics.
- Flexural displacements are small relative to the plate thickness, allowing a linearized model.

Let the mid-surface of the plate lie in a region $\mathcal{U} \subset \mathbb{R}^2$, bounded by a curve $\partial\mathcal{U} = \{(x, y) \in \mathbb{R}^2 \mid f(x, y) = 0\}$, where f is a continuous function. The plate thickness bounds z such that $z \in l^{(\zeta)} := [-\zeta/2, \zeta/2]$. Hence, in the undeformed configuration, the plate occupies the domain $\mathcal{U} \times l^{(\zeta)} \subset \mathbb{R}^3$. The orthotropic axes align with the x and y directions, while the flexural displacement of the mid-surface is described by $u = u(x, y, t) : \mathcal{U} \times \mathbb{R}_0^+ \rightarrow \mathbb{R}$. Refer to Fig. 16.

Definition 6.1. The strain vector is defined as $\boldsymbol{\varepsilon} := [\varepsilon_x, \varepsilon_y, \eta]^\top$, representing strain components along x and y , as well as the shear strain. Under the Kirchhoff-Love assumptions, these strains relate to the flexural displacement w and the *curvature vector* $\boldsymbol{\kappa} := [\kappa_x, \kappa_y, \chi]^\top$ via:

$$\boldsymbol{\varepsilon} = z\boldsymbol{\kappa}, \quad \text{with} \quad \kappa_x := -\frac{\partial^2 u}{\partial x^2}, \quad \kappa_y := -\frac{\partial^2 u}{\partial y^2}, \quad \chi := -2\frac{\partial^2 u}{\partial x \partial y} \quad (81)$$

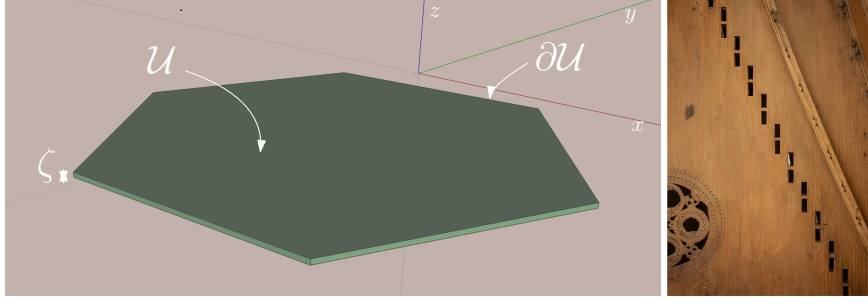


Figure 16. Left: a plate with polygonal shape and thickness ζ . Right: detail of the 1554 Gian Francesco Antegnati *arpicordo*, highlighting the wood fibres running parallel to the x axis.

□

Definition 6.2. Using Hooke's law, the stress vector $\boldsymbol{\sigma} := [\sigma_x, \sigma_y, \tau]^\top$, representing stresses along x , y , and shear stress, is given by:

$$\boldsymbol{\sigma} := \mathbf{E}\boldsymbol{\varepsilon} = z\mathbf{E}\boldsymbol{\kappa}, \quad \text{where } \mathbf{E} := \begin{bmatrix} \frac{E_x}{1-\nu_x\nu_y} & \frac{E_x\nu_y}{1-\nu_x\nu_y} & 0 \\ \frac{E_y\nu_x}{1-\nu_x\nu_y} & \frac{E_y}{1-\nu_x\nu_y} & 0 \\ 0 & 0 & G_{xy} \end{bmatrix}. \quad (82)$$

Here, \mathbf{E} is the *elasticity matrix* □

Owing to symmetry arguments of the matrix, orthotropy is realised when:

$$E_x\nu_y = E_y\nu_x. \quad (83)$$

In this formulation, E_x and E_y represent the Young's moduli along the orthotropic axes, ν_x and ν_y are the corresponding Poisson ratios, and G_{xy} is the shear modulus. In the isotropic case, the parameters simplify to $E_x = E_y := E$, $\nu_x = \nu_y := \nu$, and $G_{xy} := E/2(1 + \nu)$. For thin plates, the normal stress σ_z and shear stresses involving z are negligible.

Remark 6.3 (Non-negativity of the elasticity matrix). The elasticity matrix has non-negative eigenvalues. This is proven by direct computation and owing to (83). The eigenvalues are:

$$\lambda_x = \frac{E_x}{1 - \nu_x\nu_y}(1 - \nu_y), \quad \lambda_y = \frac{E_y}{1 - \nu_x\nu_y}(1 - \nu_x), \quad \lambda_{xy} = G_{xy},$$

which are non-negative since Poisson ratios cannot exceed unity \square

Definition 6.4. The moment vector $\boldsymbol{\mu} := [\mu_x, \mu_y, \mu_{xy}]^\top$ is derived by integrating stresses along the thickness:

$$\boldsymbol{\mu} := \int_{-\frac{\zeta}{2}}^{\frac{\zeta}{2}} z \boldsymbol{\sigma} \, dz = \int_{-\frac{\zeta}{2}}^{\frac{\zeta}{2}} z^2 \mathbf{E} \boldsymbol{\kappa} \, dz = \frac{\zeta^3}{12} \mathbf{E} \boldsymbol{\kappa}, \quad (84)$$

where the final expression follows from substituting (81) and (82) \square

Definition 6.5. The bending energy H_p is defined as:

$$H_p = \frac{1}{2} \int_{\mathcal{U}} \boldsymbol{\mu}^\top \boldsymbol{\kappa} \, dx dy, \quad (85)$$

and the kinetic energy is:

$$H_k = \frac{\rho \zeta}{2} \int_{\mathcal{U}} \left(\frac{\partial u}{\partial t} \right)^2 \, dx dy \quad (86)$$

\square

Remark 6.6 (Non-Negativity of Bending Energy). The bending energy H_p is non-negative due to the positive-definiteness of \mathbf{E} :

$$H_p = \frac{\zeta^3}{24} \int_{\mathcal{U}} \boldsymbol{\kappa}^\top \mathbf{E} \boldsymbol{\kappa} \, dx dy \geq 0 \quad (87)$$

\square

Using the energy definitions, the equation of motion is derived by requiring conservation of total energy:

$$\frac{d}{dt} H := \frac{d}{dt} (H_k + H_p) = 0. \quad (88)$$

By integrating by parts, the energy balance splits into a domain integral and a boundary integral:

$$0 = \frac{d}{dt} H = \int_{\mathcal{U}} \frac{\partial u}{\partial t} \mathbf{L}(u) \, dx dy - \int_{\partial \mathcal{U}} \mathbf{B}(u) \, d\partial \mathcal{U}. \quad (89)$$

Requiring $\mathbf{L}(u) = 0$ inside \mathcal{U} gives the equation of motion, while $\mathbf{B}(u) = 0$ provides the boundary conditions.

Definition 6.7. The *Kirchhoff-Love* model for orthotropic plates is given by:

$$\rho\zeta \frac{\partial^2 u}{\partial t^2} = -D_x \frac{\partial^4 u}{\partial x^4} - 2Q \frac{\partial^4 u}{\partial x^2 \partial y^2} - D_y \frac{\partial^4 u}{\partial y^4}, \quad (90)$$

with rigidity constants:

$$D_x := \frac{E_x \zeta^3}{12(1 - \nu_x \nu_y)}, \quad D_y := \frac{E_y \zeta^3}{12(1 - \nu_x \nu_y)}, \quad D_{xy} := \frac{G_{xy} \zeta^3}{12},$$

and with:

$$Q = \frac{1}{2} (\nu_y D_x + \nu_x D_y + 4D_{xy}).$$

□

Boundary conditions depend on the geometry of $\partial\mathcal{U}$. For simplicity, consider the quarter-plane $\mathcal{U} = \{(x, y) \in \mathbb{R}^2 \mid x \geq 0, y \geq 0\}$.

Definition 6.8. The boundary conditions are expressed through a contour integral:

$$\begin{aligned} \int_{\partial\mathcal{U}} \mathbf{B}(u) \, d\partial\mathcal{U} &= \int_{\mathbb{R}_0^+} \left(\frac{\partial u}{\partial t} q_x + \frac{\partial^2 u}{\partial t \partial x} \mu_x \right) \Big|_{x=0} dy \\ &\quad + \int_{\mathbb{R}_0^+} \left(\frac{\partial u}{\partial t} q_y + \frac{\partial^2 u}{\partial t \partial y} \mu_y \right) \Big|_{y=0} dx, \end{aligned} \quad (91)$$

where the lateral edge force and moment are defined as:

$$q_x := -D_x \left(\frac{\partial^3 u}{\partial x^3} + \left(\frac{4D_{xy}}{D_x} + \nu_y \right) \frac{\partial^3 u}{\partial x \partial y^2} \right), \quad (92a)$$

$$\mu_x := D_x \left(\frac{\partial^2 u}{\partial x^2} + \nu_y \frac{\partial^2 u}{\partial y^2} \right), \quad (92b)$$

with analogous expressions for q_y , μ_y (interchanging x and y indices, and noting $D_{yx} = D_{xy}$). A corner condition arises at $(x, y) = (0, 0)$:

$$\frac{\partial^2 u}{\partial x \partial y} = 0, \quad (93)$$

due to a discontinuity in the twisting moment, see (Gérardin and Rixen, 2015, Ch. 4) □

These integrals recover standard boundary conditions for free, clamped, and simply supported edges:

$$\text{free: } \mu_n = q_n = 0, \quad \text{clamped: } u = \frac{\partial u}{\partial n} = 0, \quad \text{simply-sup: } \mu_n = u_n = 0.$$

where n is the direction normal to the boundary. These generalise the boundary conditions given for the beam in Section 5.2.

Plates often exhibit elastic boundary supports, such as edge beams lacking torsional rigidity (Takabatake and Nagareda (1999)) or partial supports as seen in musical instrument soundboards (Fletcher and Rossing (2012)). These are modelled using boundary elastic forces governed by flexural and rotational springs. In (91), the boundary force and moment become:

$$q_x(0, y, t) = K_{x0}u(0, y, t), \quad \mu_x(0, y, t) = R_{x0} \frac{\partial u(0, y, t)}{\partial x}, \quad (94)$$

where K_{x0} and R_{x0} are spring constants. Similar expressions hold for $y = 0$, with constants K_{y0} , R_{y0} . Free, clamped, and simply supported edges correspond to $K_o = R_o = 0$, $K_o \rightarrow \infty, R_o \rightarrow \infty$, and $K_o \rightarrow \infty, R_o = 0$, respectively.

Definition 6.9. The eigenvalue problem for the Kirchhoff-Love plate equation is obtained after substituting the transformed solution $u(x, y, t) = e^{j\omega t}\hat{u}(x, y)$ in (90). One then gets:

$$\mathcal{L}\hat{u} = \omega^2\hat{u}, \quad \mathcal{L} := (\rho\zeta)^{-1} \left(D_x \frac{\partial^4}{\partial x^4} + 2Q \frac{\partial^4}{\partial x^2 \partial y^2} + D_y \frac{\partial^4}{\partial y^4} \right), \quad (95)$$

and where \hat{u} satisfies a set of boundary conditions as per (94) \square

Example 6.10. Consider a rectangular plate with edge lengths L_x, L_y and under simply-supported edges. The boundary conditions reduce to $\hat{u} = \frac{\partial^2 \hat{u}}{\partial n^2} = 0$, and an analytic solution to the eigenvalue problem exists as:

$$\hat{u}^{(p)} = A^{(p)} \sin(\gamma^{(p_x)}x) \sin(\gamma^{(p_y)}y), \quad (96)$$

with:

$$\gamma^{(p_x)} := \frac{p_x \pi}{L_x}, \quad \gamma^{(p_y)} := \frac{p_y \pi}{L_y} \quad (97)$$

being the quantised wavenumbers along x and y . Above, $(p_x, p_y) \in \mathbb{N}_0$ and $A^{(p)}$ is a constant of normalisation. The corresponding eigenfrequencies are obtained as:

$$(\omega^{(p)})^2 = (\rho\zeta)^{-1} \left(D_x (\gamma^{(p_x)})^4 + 2Q (\gamma^{(p_x)})^2 (\gamma^{(p_y)})^2 + D_y (\gamma^{(p_y)})^4 \right). \quad (98)$$

This solution generalises the solution for the simply-supported beam illustrated in Example 5.11 \square

Modal shapes and frequencies for different boundary conditions are harder to obtain, and one must resort to an appropriate numerical approximation. In the following, a rectangular domain \mathcal{U} is assumed throughout.

6.2 Finite difference operators in two dimensions

The finite difference method can be easily extended to treat two-dimensional problems.

Definition 6.11. A two-dimensional grid \mathfrak{U} is the collection of grid points x_m, y_n and related cells $\{[x_m, x_{m+1}] \times [y_n, y_{n+1}]\}$ such that

$$\begin{aligned} 0 = x_0 &\leq x_1 \leq \dots \leq x_m \leq \dots \leq x_M = L_x \\ 0 = y_0 &\leq y_1 \leq \dots \leq y_n \leq \dots \leq y_N = L_y \end{aligned}$$

Like the one-dimensional case, various kinds of grids can be defined, such as the uniform, smooth and non-smooth. In the following, we will only consider a uniform grid for which $x_{m+1} - x_m = \Delta_x \forall m, y_{n+1} - y_n = \Delta_y \forall n$ \square

Definition 6.12. A two-dimensional grid function $u_{m,n}$ is any $(M+1) \times (N+1)$ tensor defined at the grid points (x_m, y_n) . In practice, the grid is often recombined into a vector by stacking consecutive strips of points. As such, the two-dimensional grid function becomes the vector $\mathbf{u} : \mathfrak{U} \rightarrow \mathbb{R}^{(M+1)(N+1)}$. Like the one-dimensional case, the finite difference method is used to solve a differential problem in $u(x, y)$ by computing a grid function \mathbf{u} such that:

$$u_{m,n} = u(x_m, y_n) + \mathcal{O}(\Delta_x^p) + \mathcal{O}(\Delta_y^q), \quad (99)$$

where p, q are the orders of convergence in x and y . This definition implies that, as the grid spacings Δ_x, Δ_y are made smaller, the approximate grid function tends to the true solution at the grid points \square

Operators analogous to those reported in Table 2 can be defined acting on the grid function $u_{m,n}$, such that:

$$\begin{aligned} \delta_{xx} u_{m,n} &:= \frac{u_{m+1,n} - 2u_{m,n} + u_{m-1,n}}{\Delta_x^2} \\ \delta_{yy} u_{m,n} &:= \frac{u_{m,n+1} - 2u_{m,n} + u_{m,n-1}}{\Delta_y^2}. \end{aligned}$$

The expressions for various difference operators are obtained similarly. In vector form, the difference operators are represented by sparse matrices, so that:

$$\delta_{xx} u_{m,n} \rightarrow \mathbf{D}_{xx} \mathbf{u}, \quad \delta_{yy} u_{m,n} \rightarrow \mathbf{D}_{yy} \mathbf{u}, \quad (100)$$

and similarly for other difference operators. The matrices are $(M+1)(N+1) \times (M+1)(N+1)$ and take the following forms:

$$\mathbf{D}_{xx} := (\Delta_x)^{-2} \begin{bmatrix} \ddots & \ddots & \ddots & \ddots & \ddots & \\ \cdots & \mathbf{I} & -2\mathbf{I} & \mathbf{I} & \mathbf{0} & \cdots \\ \cdots & \mathbf{0} & \mathbf{I} & -2\mathbf{I} & \mathbf{I} & \cdots \\ & & \ddots & \ddots & \ddots & \ddots \end{bmatrix},$$

$$\mathbf{D}_{yy} := (\Delta_y)^{-2} \begin{bmatrix} \ddots & \ddots & \ddots & \ddots & \ddots & \\ \cdots & 1 & -2 & 1 & 0 & \cdots \\ \cdots & 0 & 1 & -2 & 1 & \cdots \\ & & \ddots & \ddots & \ddots & \ddots \end{bmatrix},$$

where \mathbf{I} is the $(N+1) \times (N+1)$ identity matrix. At interior points, the fourth difference matrices are obtained by simply multiplying the second difference matrices by themselves, so that, for example, $(\mathbf{D}_{xxxx}\mathbf{u})_l = (\mathbf{D}_{xx}^2\mathbf{u})_l$ if l is sufficiently distant from the boundary. However, this relationship breaks down when the stencil of the fourth difference operator involves points on the boundary or beyond. A specialised form for the operators near the boundary is recovered once the ghost points are set via a suitable discretisation of the boundary conditions. Consider the expressions (92) for the shear force and moment for the edge at $x = 0$. Using centred operators, the following expressions are recovered:

$$\begin{aligned} q_x &= -D_x \left(\delta_{xxx} u_{0,n} + \left(\frac{4D_{xy}}{D_x} + \nu_y \right) \delta_{yy} \delta_x u_{0,n} \right), \\ \mu_x &= D_x (\delta_{xx} u_{0,n} + \nu_y \delta_{yy} u_{0,n}), \end{aligned}$$

and the elastic boundary conditions are thus set as:

$$q_x = K_{x0} u_{0,n}, \quad \mu_x = R_{x0} \delta_x u_{0,n}. \quad (101)$$

One may impose similar conditions along all four edges of the plate and use a centred discretisation of the corner condition (93):

$$\delta_x \delta_y u_{m,n} = 0 \quad \text{at each corner.} \quad (102)$$

The numerical boundary conditions allow for the setting of all ghost points and, hence, the specialisation of the difference matrix around the boundary. Analytic expressions for the resulting matrices are cumbersome and omitted here for brevity. These are found in the script `orthoplate_biharm_build.m` uploaded to the GitHub repository.

Definition 6.13. A discrete eigenvalue problem for the Kirchhoff-Love results as:

$$(\rho\zeta)^{-1} (D_x \mathbf{D}_{xxxx} + 2Q \mathbf{D}_{xx} \mathbf{D}_{yy} + D_y \mathbf{D}_{yyyy}) \hat{\mathbf{u}} = \omega^2 \hat{\mathbf{u}}. \quad (103)$$

This problem will be solved in the next subsection using the MAGPIE toolbox \square

6.3 The MAGPIE toolbox for plate analysis

MAGPIE is an open-source toolbox to solve the orthotropic plate equation in the time and frequency domains. It is currently hosted on GitHub and is available as Matlab code. The toolbox can be used for three main purposes:

- Frequency domain analysis producing eigenfunctions and frequencies of orthotropic plates under elastic boundary support.
- Time domain simulation under various dynamic input loads, including sinusoidal, impulsive, and externally-recorded.
- Inverse estimation of the elastic constant of an experimental plate.

Furthermore, MAGPIE allows attaching ribs and static lumped loads to the surface. The model equations have not been laid out in these notes, but the ribs are modelled dynamically as Euler-Bernoulli beams constrained to have the same displacement as the plate along their lengths. See Ducceschi (2024) for details.

The frequency domain analysis User-adjustable parameters are configured in the `FreqDomainLauncher` script. General parameters are specified first:

```
%-----
%-- General parameters
fmax   = 2000 ; % Maximum frequency to compute
ppw    = 3 ;   % Points per wavelength at maximum frequency
Nmodes = 6 ;   % Number of modes to compute ([] = all modes)
%-----
```

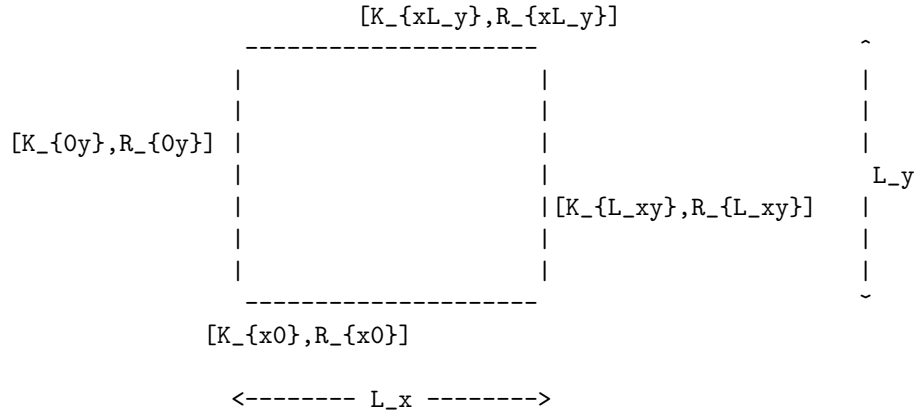
The parameter `fmax` determines the highest frequency f^{\max} used in defining the mesh size, as per (80). `ppw` specifies the points per wavelength at f^{\max} , and `Nmodes` defines the number of modes to be computed. If `Nmodes = []`, all available modes up to `fmax` are included.

Next, the plate parameters are configured:

```
%-----
%-- Plate parameters
rho    = 390 ;    % Density [kg/m^3]
nux     = 0.39 ;  % Poisson ratio
Ex      = 10.9e9 ; % Young's modulus along x [Pa]
Ey      = 0.64e9 ; % Young's modulus along y [Pa]
Gxy     = 0.58e9 ; % Shear modulus [Pa]
Lx      = 0.6 ;   % Edge length along x [m]
Ly      = 0.6 ;   % Edge length along y [m]
Lz      = 1.0e-3 ; % Thickness [m]
```

The second Poisson's ratio is determined via (83), ensuring symmetry of the elasticity matrix. The boundary conditions are set through the elastic constants:

```
%-- Elastic constants around the edges
KRmat = [1e4, 1e5; % Kx0 Rx0
         1e4, 1e4;  % K0y R0y
         0, 0;      % KxL RxL
         0, 0] ;    % KLy RLy
%-----
```



For the rectangular geometry above, eight constants control the edge lateral force q_o and twisting moment μ_o as per (94). High stiffness values, around 10^{10} , simulate fixed conditions, though excessively large values can lead to ill-conditioned stiffness matrices.

Stiffener parameters are defined as follows:

```

%-----
%-- Brace parameters
Nribs = 4 ; % Number of braces

Eb  = [10e9, 11e9, 10.2e9, 200e9].'; % Young's moduli [Pa]
Lzb = [3e-3, 2e-3, 2e-3, 1e-3].'; % Thicknesses [m]
bb  = [3e-2, 2e-2, 2e-2, 1e-2].'; % Widths of cross-section [m]
rhob = [400, 390, 410, 8000].'; % Densities [kg/m^3]

% Rib coordinates along x (start and end) as a fraction of Lx
x_beam_coord = ...
    [0.2, 0.8;
     0.2, 0.8;
     0.2, 0.8;
     0.2, 0.8] ;

% Rib coordinates along y (start and end) as a fraction of Ly
y_beam_coord = ...
    [0.1, 0.2;
     0.2, 0.3
     0.3, 0.4
     0.4, 0.5] ;
%-----

Nribs corresponds to the total number of stiffeners. Eb, Lzb, bb, and rhob
define the Young's moduli, thicknesses, widths, and densities of the stiffen-
ers. The beams are assumed to have a constant thickness along their length.
Beam domains are set by their  $x$  and  $y$  endpoints, e.g., the first beam spans
 $(0.2L_x, 0.1L_y)$  to  $(0.8L_x, 0.2L_y)$ .

Lumped mass parameters are defined similarly:

%-----
%-- Lumped masses
Nlump      = 2 ;
x_lump_coord = [0.2, 0.4].';
y_lump_coord = [0.7, 0.87].';
Mlump      = [0.2, 0.01].'; % Masses [kg]
%-----

```

Example 6.14. To benchmark, consider an orthotropic plate with parameters as defined above. Eigenfrequencies are computed for fully free and clamped cases, and results are compared with COMSOL Multiphysics simulations in Tables 4 and 5, showing excellent agreement \square

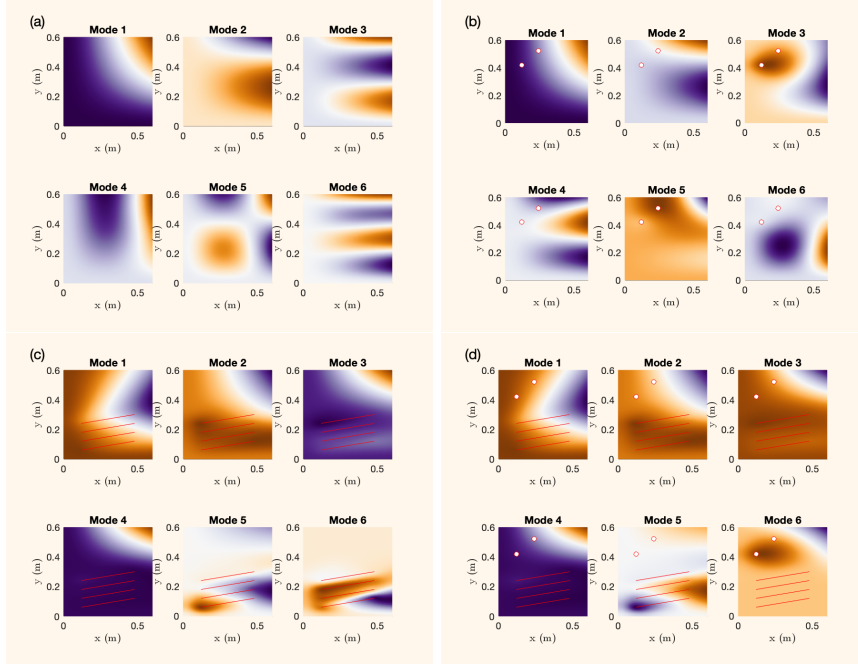


Figure 17. First six eigenmodes of a partially supported plate with lumped masses (white dots) and stiffening beams (red lines). Parameters: $f_{\max} = 2000$, $ppw = 9$, yielding $M = 78$, $N = 158$. (a) No lumped elements or stiffeners; (b) Lumped elements only; (c) Stiffeners only; (d) Both lumped elements and stiffeners

Example 6.15. Fig. 17 illustrates the first six modal shapes for a partially supported orthotropic square plate with and without added masses and ribs. The modal shapes exhibit significant localization near stiffeners, a phenomenon common in piano soundboards, see e.g. Chaigne et al. (2013). Parameters for the braces and lumped masses are as given above \square

7 Modal Sound Synthesis

The chapter on physical modelling by Vasileios Chatziioannou covers time discretisation in detail. Here, wave propagation will be simulated by taking advantage of the numerical eigenvalue solvers developed in the previous section. Simulation techniques in which the system is first projected via the

	ppw = 3	ppw = 5	ppw = 7	ppw = 9	COMSOL
Mode 1	3.53	3.53	3.53	3.53	3.53
Mode 2	3.65	3.66	3.66	3.66	3.66
Mode 3	8.07	8.08	8.09	8.09	8.09
Mode 4	10.1	10.1	10.1	10.1	10.1
Mode 5	14.7	14.8	14.8	14.8	14.9
Mode 6	15.0	15.1	15.1	15.1	15.1

Table 4. Eigenfrequency benchmark against COMSOL Multiphysics. Parameters as per Section 6.3, boundary conditions fully free, $\mathbf{f}_{\max} = 2000$, and varying **ppw**. COMSOL uses quadratic Lagrange shape functions and 9 points per wavelength. Rigid-body frequencies are excluded.

	ppw = 3	ppw = 5	ppw = 7	ppw = 9	COMSOL
Mode 1	16.1	16.1	16.1	16.1	16.1
Mode 2	19.8	19.9	19.9	19.9	19.9
Mode 3	27.5	27.6	27.6	27.6	27.7
Mode 4	39.0	39.3	39.3	39.3	39.4
Mode 5	42.1	42.5	42.6	42.6	42.6
Mode 6	45.2	45.5	45.6	45.7	45.7

Table 5. Same as Table 4, but with fully clamped edges. Stiffness constants set to 10^{10} to ensure clamping.

system’s eigenfunctions (the *modes*) onto a bank of parallel oscillators are collectively known as *modal synthesis*.

Modal methods were initially developed in the context of engineering, where they were used to predict the behaviour of mechanical structures (Meirovitch (1970)). In the late 20th century, as computer music emerged as a field, researchers began to adapt these principles for sound synthesis. Pioneers such as Joseph Morrison and Jean-Marie Adrien at IRCAM formalised modal synthesis techniques, using them to simulate acoustic objects (Adrien (1991); Morrison and Adrien (1993)). By the 1990s, advancements in computational power allowed for more sophisticated applications, enabling the simulation of complex instruments like bells (Debut and Antunes (2014); Carvalho et al. (2021)), plates (Ducceschi and Touzé (2015a); Ducceschi et al. (2012); Ducceschi and Touzé (2015b)) and drums (Kirby and Sandler (2021)), in both linear and nonlinear regimes. Modal synthesis became especially appealing due to its ability to recreate lifelike sounds

by treating them as the sum of damped oscillators, each corresponding to a specific mode of vibration, yielding a kind of synthesis akin to additive but rooted in physical principles, for which most synthesis parameters are naturally assigned by the system’s physical properties. As digital tools matured, software like IRCAM’s Modalys[†] made it easier for musicians and sound designers to craft musically appealing textures.

7.1 Time domain models with input forcing and dissipation

Simulating wave propagation requires the injection of energy into the system, whether it is plucking a string, striking a bar or tapping a plate. The input is concentrated on a small area of the instrument, and, in a first approximation, a Dirac delta suffices to model the spatial extent of the external input forcing. Furthermore, energy is drawn from the system due to various forms of loss. Losses play a fundamental role in shaping the sound of a synthesis model, but a thorough discussion of the physical mechanisms producing losses in musical instruments is beyond the current scope. See, e.g., Valette (1995); Woodhouse (2004) for strings as well as Arcas and Chaigne (2010) for plates. A general discussion on damping in structures is given by Woodhouse (1998). Here, a rather simple model due to Rayleigh is implemented (Vandiver (2011)), which has the advantage of yielding a diagonal form for the losses using two user-selectable parameters.

Definition 7.1. In one dimension, a model for wave propagation incorporating losses and external, pointwise input forcing is given as:

$$\frac{\partial^2 u(t, x)}{\partial t^2} = -\mathcal{L}u(x, t) - (a + b\mathcal{L})\frac{\partial u(t, x)}{\partial t} + \delta(x - \bar{x})f(t), \quad (104)$$

where \mathcal{L} is a differential operator, such as $\mathcal{L} = -c^2 \frac{\partial^2}{\partial x^2}$ for the simple wave equation, or $\mathcal{L} = \kappa^4 \frac{\partial^4}{\partial x^4}$ for the bar equation with constant thickness; a and b are user-defined loss parameters; \bar{x} is the forcing point; $f(t)$ is an externally-supplied time-dependent scaled input. A model in two dimensions, such as for the Kirchhoff-Love equation, is obtained analogously \square

In one dimension, the Vandermonde interpolation described in Section 2.2 returns a strategy to represent the delta function using a subset of points sampled from the continuous function $u(x)$.

Definition 7.2. Consider a subset of equally spaced sampled points $u(x_m)$, with grid spacing Δ_x . An implicit definition of the discrete delta (also known

[†]<https://forum.ircam.fr/projects/detail/modalys/> (accessed May 2025)

as a *spreading* operator) is as follows:

$$u(\bar{x}) = \int_{\mathcal{I}} \delta(x - \bar{x})u(x) = \Delta_x \boldsymbol{\delta}^T \mathbf{u} + \mathcal{O}(\Delta_x^p), \quad (105)$$

where \mathbf{u} is the vector of sampled points, and where $\boldsymbol{\delta}$ is a discrete version of the delta function. The approximation is p^{th} -order accurate in the mesh size Δ_x \square

Example 7.3. The Vandermonde matrix method can be used to find expressions for the spreading vector. First, consider two sample points $u(x_m)$ and $u(x_{m+1})$, and suppose $x_m \leq \bar{x} \leq x_{m+1}$. From (105), one has:

$$u(\bar{x}) \approx \Delta_x (\delta_m u(x_m) + \delta_{m+1} u(x_{m+1})). \quad (106)$$

Using the Vandermonde matrix method as explained in Section 2.2 to build the polynomial $p(x)$, one has:

$$\begin{bmatrix} 1 & x_m - \bar{x} \\ 1 & x_{m+1} - \bar{x} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} u(x_m) \\ u(x_{m+1}) \end{bmatrix}, \quad (107)$$

from which:

$$\alpha_0 = (1 - \beta)u(x_m) + \beta u(x_{m+1}), \text{ with } \beta := \Delta_x^{-1}(\bar{x} - x_m). \quad (108)$$

This allows concluding that:

$$\delta_m = \Delta_x^{-1}(1 - \beta), \quad \delta_{m+1} = \Delta_x^{-1}\beta. \quad (109)$$

As one might expect, this approximation is only first-order accurate. Higher-order approximations may be obtained by using more sample points drawn from $u(x)$, and centred approximations \square

Example 7.4. Spreading operators can be constructed similarly in two dimensions. Approximating the delta function at (\bar{x}, \bar{y}) (a point contained within the square with corners $(u_m, y_n), (u_{m+1}, y_n), (u_m, y_{n+1}), (u_{m+1}, y_{n+1})$) is immediate from the previous example. Let the grid spacings along x and y be Δ_x and Δ_y , respectively. One has:

$$\begin{aligned} \delta_{m,n} &= \Delta_x^{-1} \Delta_y^{-1} (1 - \beta_x)(1 - \beta_y), \\ \delta_{m+1,n} &= \Delta_x^{-1} \Delta_y^{-1} \beta_x (1 - \beta_y), \\ \delta_{m,n+1} &= \Delta_x^{-1} \Delta_y^{-1} (1 - \beta_x) \beta_y, \\ \delta_{m+1,n+1} &= \Delta_x^{-1} \Delta_y^{-1} \beta_x \beta_y, \end{aligned} \quad (110)$$

with $\beta_x := \Delta_x^{-1}(\bar{x} - x_m)$, $\beta_y := \Delta_y^{-1}(\bar{y} - y_n)$ \square

In the previous sections, as per Definition 5.1, the continuous operator \mathcal{L} was discretised using a finite difference approximation, such as the continuous eigenvalue problem was turned into a numerical eigenvalue problem as follows:

$$\mathcal{L}\hat{u} = \omega^2 \hat{u} \rightarrow \mathbf{L}\hat{\mathbf{u}} = \omega^2 \hat{\mathbf{u}}, \quad (111)$$

where \mathbf{L} , a sparse matrix, is the finite difference approximation to \mathcal{L} , $\hat{\mathbf{u}}$ is an eigenvector, and ω^2 , a real, non-negative number, is the corresponding eigenvalue[‡]. When discretised via the finite difference method, a matrix decomposition in the form of eigenvalues and eigenvectors results for the linear operator.

Definition 7.5. The finite difference operator \mathbf{L} is expressed via its eigenvalue decomposition as:

$$\mathbf{L} = \hat{\mathbf{U}}\mathbf{\Omega}^2\hat{\mathbf{U}}^{-1}, \quad (112)$$

where $\hat{\mathbf{U}}$ is a matrix whose columns are the eigenvectors, i.e.

$$\hat{\mathbf{U}} := [\hat{\mathbf{u}}^{(1)}, \hat{\mathbf{u}}^{(2)}, \dots], \quad (113)$$

and where $\mathbf{\Omega}$ is a diagonal matrix of eigenvalues \square

In practice, (104) is first discretised in space using the eigendecomposition of the operator \mathbf{L} and any suitable approximation to the delta function:

$$\ddot{\mathbf{u}} = -\hat{\mathbf{U}}\mathbf{\Omega}^2\hat{\mathbf{U}}^{-1}\mathbf{u} - (a + b\hat{\mathbf{U}}\mathbf{\Omega}^2\hat{\mathbf{U}}^{-1})\dot{\mathbf{u}} + \delta f(t). \quad (114)$$

Then, left-multiplying the system by $\hat{\mathbf{U}}^{-1}$ and defining $\mathbf{v} := \hat{\mathbf{U}}^{-1}\mathbf{u}$, $\boldsymbol{\eta} := \hat{\mathbf{U}}^{-1}\delta$, one finally obtains the modal system:

$$\ddot{\mathbf{v}} = -\mathbf{\Omega}^2\mathbf{v} - (a + b\mathbf{\Omega}^2)\dot{\mathbf{v}} + \boldsymbol{\eta}f(t), \quad (115)$$

which is a fully diagonal system of damped, forced oscillators.

Example 7.6. The coefficients a and b in the formulation of Rayleigh damping can be determined using the decay times at two specific frequencies. Let τ_o represent the 60 dB decay time associated with the frequency

[‡]For the sake of conciness, the notation ω was retained in both the continuous and the discrete eigenvalue problems, but it is intended that the discrete eigenvalue problem returns an *approximation* to the continuous problem's eigenvalues.

ω_\circ . Employing the decay time expression $\tau_\circ = \frac{3 \log(10)}{a + b\omega_\circ^2}$, the parameters a and b can be derived as follows:

$$a = \frac{3 \log(10)}{\omega_2^2 - \omega_1^2} \left(\frac{\omega_2^2}{\tau_1} - \frac{\omega_1^2}{\tau_2} \right), \quad (116a)$$

$$b = \frac{3 \log(10)}{\omega_2^2 - \omega_1^2} \frac{\tau_1 - \tau_2}{\tau_1 \tau_2}. \quad (116b)$$

To ensure energy dissipation across all frequencies, the condition $\tau_2 < \tau_1$ must hold if $\omega_2 > \omega_1$, and $\tau_1 \omega_1^2 < \tau_2 \omega_2^2$. For simplicity, one may choose $\omega_1 = 0$, which corresponds to a frequency-independent decay time τ_1 \square

7.2 Conservative time discretisation

Simulating transient and broadband wave propagation requires time discretisation of the modal system (115). Since the system is fully diagonal, the scalar case of a damped oscillator with forcing can be considered:

$$\ddot{v} = -\omega^2 v - 2\sigma \dot{v} + \eta f(t), \quad \sigma := a + b\omega^2. \quad (117)$$

Multiplying through by \dot{v} yields an energy balance for modal energy scaled by mass:

$$\frac{d}{dt} H = -2\sigma \dot{v}^2 + \dot{v} \eta f(t), \quad H := \frac{\dot{v}^2}{2} + \frac{\omega^2 v^2}{2}. \quad (118)$$

Thus, when $f(t) = 0$, the modal energy decreases.

A discrete version of (117) replaces the continuous function $v(t)$ with a time series $v[k]$, where $t_k = Tk$ for time step T and index k . The time difference operators are defined as follows:

$$\delta_t v[k] := \frac{v[k+1] - v[k-1]}{2T}, \quad (119a)$$

$$\delta_{tt} v[k] := \frac{v[k+1] - 2v[k] + v[k-1]}{T^2}, \quad (119b)$$

$$\nu_{tt} v[k] := \frac{v[k+1] + 2v[k] + v[k-1]}{4}. \quad (119c)$$

This gives the time-discretized equation (see e.g. van Walstijn et al. (2016, 2024)):

$$\delta_{tt} v[k] = -\tilde{\omega}^2 \nu_{tt} v[k] - 2\tilde{\sigma} \delta_t v[k] + \eta f[k], \quad (120)$$

where:

$$\begin{aligned}\tilde{\omega}^2 &:= \frac{4}{T^2} \frac{1 - 2e^{-\sigma T} \cos(T\sqrt{\omega^2 - \sigma^2}) + e^{-2\sigma T}}{1 + 2e^{-\sigma T} \cos(T\sqrt{\omega^2 - \sigma^2}) + e^{-2\sigma T}} \approx \omega^2, \\ \tilde{\sigma} &:= \frac{2}{T} \frac{1 - e^{-\sigma T}}{1 + 2e^{-\sigma T} \cos(T\sqrt{\omega^2 - \sigma^2}) + e^{-2\sigma T}} \approx \sigma.\end{aligned}$$

Multiplying (120) by $\delta_t.v[k]$ produces a discrete energy balance analogous to (118):

$$\frac{\mathfrak{H}[k + \frac{1}{2}] - \mathfrak{H}[k - \frac{1}{2}]}{T} = -2\tilde{\sigma}(\delta_t.v[k])^2 + \delta_t.v[k]\eta f[k], \quad (121)$$

with:

$$\mathfrak{H}[k + \frac{1}{2}] := \frac{(v[k+1] - v[k])^2}{2T^2} + \frac{\tilde{\omega}^2(v[k+1] + v[k])^2}{2}. \quad (122)$$

Scheme (120) effectively avoids numerical warping when $f[k] = 0$ by discretising the continuous solution exactly if T is sufficiently small (Cieřliński (2011)). Moreover, the numerical modal energy remains non-negative (Bilbao, 2009, Ch.3), ensuring unconditional stability. The time step is chosen as:

$$T \approx 2/\omega^{(\max)}, \quad (123)$$

where $\omega^{(\max)}$ is the highest eigenfrequency of the reduced model. This choice corresponds to the stability limit of the classic leapfrog method (LeVeque, 2007, Ch.7). Updating the reduced-order vector $\mathbf{v}[k]$ enables solution reconstruction at grid nodes via $\mathbf{u} = \hat{\mathbf{U}}\mathbf{v}$. The numerical error stems from approximations in the eigenproblem (60), but appropriate mesh selection based on points per wavelength allows for a form of control (Marburg (2002); Langer et al. (2017)).

7.3 Sound Synthesis of Plates in the MAGPIE toolbox

The MAGPIE framework allows performing time domain simulation of orthotropic plates using the modal scheme above. The script `TimeDomainLauncher` manages user-configurable parameters, many of which are similar to those in the frequency domain. However, additional parameters specific to time-domain simulations, such as input/output locations, decay properties, and the type of force applied, are also required. Input/output locations are defined as follows:

```
%-----
%-- input / output locations
```

```

x_p    = [0.513,0.678] ;           % fractional {(Lx Ly)}
outMat = [0.51,0.52;
          0.12,0.76] ;           % fractional {(Lx Ly)},
                                   % rows indicate output points
%-----

```

The variable `x_p` specifies a single input location, while multiple output points can be defined using `outMat`. The loss parameters, configured via the array `dampVec`, are as follows:

```

%-----
%-- loss parameters
dampVec = [0.3, 0.23, 500] ;
% [t60_0Hz (s), t60_f1 (s), f1 (Hz)]
% Stability condition: t60_f1 < t60_0Hz
%-----

```

The decay time constant, `t60_0Hz`, governs frequency-independent losses in the system, as per Example 7.6, assuming $\omega_1 = 0$. The input force configuration is as follows:

```

%-----
%-- forcing
forceType = 1 ; % 1 = impulse, 2 = sinusoid, 3 = input_file
if forceType == 1
    forceParams = [0.0007,5,0.5] ;
    % [contact time (s), max amplitude (N), noise modulation]
elseif forceType == 2
    forceParams = [100,50,0.2] ;
    % [sine frequency (Hz), max amplitude (N), noise modulation]
elseif forceType == 3
    forceParams = 'testInput.wav' ;
end
%-----

```

The `forceType` parameter specifies the input force $f(t)$. For `forceType` = 1, a raised cosine force is applied:

$$f(t) = r(t)(1 + gn(t)), \quad r(t) = \frac{F}{2} \left(1 - \cos \frac{2\pi t}{t_p} \right), \quad 0 \leq t \leq t_p, \quad (124)$$

where $f(t) = 0$ otherwise, F , t_p , and g correspond to elements in `forceParams`. For `forceType` = 2, a sinusoidal force is generated:

$$f(t) = s(t)(1 + gn(t)), \quad s(t) = F \sin(2\pi f_p t), \quad (125)$$

where F , f_p , and g are the respective parameters. The `forceType = 3` setting allows for loading an external input, such as an impact hammer recording. This input is resampled to match the simulation rate.

The real-time plotting feature can be toggled using the `LivePlot` parameter:

```
%-----
%-- plot parameters
cmap      = cmaps(2) ; % colormap options:
                        % 1 = RedBlue, 2 = GreenPurple, etc.
LivePlot   = 0 ;      % 1: enable live plot, 0: disable
RefreshRate = 1 ;      % skip frames for performance
absPlot    = 0 ;      % 1: absolute value plotting
FilmRec    = 0 ;      % 1: record video
%-----
```

Example snapshots of plate dynamics, showing displacement, velocity, and acceleration fields, are displayed in Fig. 18. These snapshots illustrate orthotropic wave propagation, evident in the oval-shaped waveform due to differing speeds along the x - and y -axes, diffraction effects caused by ribs, and high-frequency content in acceleration fields due to time derivatives.

Similar conclusions can be drawn by looking at the example output waveforms in Fig. 19, where the output acceleration presents a clear amplification of shorter wavelengths compared to velocity and displacement.

8 Conclusions

This lecture offered a brief but focused entry point into eigenvalue problems as encountered in musical acoustics, arising in settings such as structural analysis, geometric optimisation, and the identification of material parameters. These problems, ubiquitous across the field, are increasingly tackled using advanced numerical simulation, enabled by software platforms capable of treating complex domains, heterogeneous media, and coupled physical systems.

The examples presented herein are deliberately simple, intended to provide the standard pipeline from continuous models to numerical discretisation. Emphasis was placed on the formulation and solution of eigenvalue problems within this workflow, using tools that are readily generalisable. A solid understanding of the numerical methods involved, particularly eigenvalue routines, is a strong asset for anyone wishing to interpret results from more elaborate commercial packages.

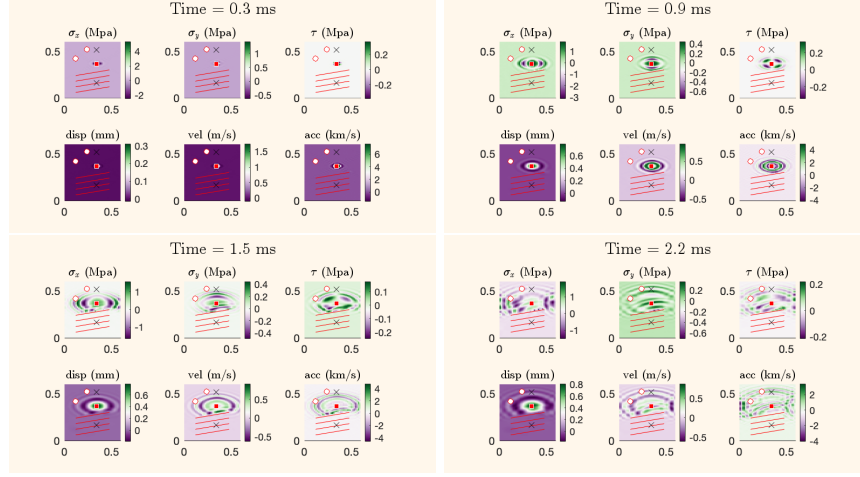


Figure 18. Example snapshots of the stress fields at the plate's surface ($z = \zeta/2$ in (82)) and of the displacement, velocity and acceleration fields obtained with the MAGPIE toolbox. White dots represent the static loads applied to the plate; red lines are the user-configurable ribs; black crosses are the output points.

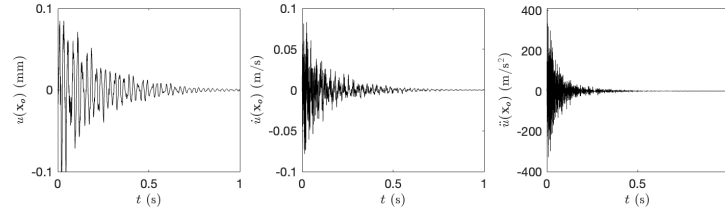


Figure 19. Example output displacement, velocity and acceleration waveforms.

For those inclined, the techniques laid out may serve as a foundation for further development—towards problems posed on curved or irregular domains, or governed by more comprehensive models such as the full equations of 3D elasticity. The accompanying codebase, available on GitHub, is freely shared, and the author encourages its distribution among students, colleagues, and fellow researchers.

Bibliography

- J.-M. Adrien. The missing link: Modal synthesis. In G. De Poli, A. Piccialli, and C. Roads, editors, *Representations of Musical Signals*, pages 269–298. MIT Press, 1991.
- K. Arcas and A. Chaigne. On the quality of plate reverberation. *Appl. Acoust.*, 71(2):147–156, 2010.
- B. Bank. A modal-based real-time piano synthesizer. *IEEE Trans. Audio Speech Lang. Process.*, 18(4):809–821, 2010. doi: 10.1109/TASL.2009.2038077.
- B. Bank and L. Sujbert. Generation of longitudinal vibrations in piano strings: From physics to sound synthesis. *J. Acoust. Soc. Am.*, 117(4):2268–2278, 2005. doi: 10.1121/1.1869074.
- K.-J. Bathe. *Finite Element Procedures*. Prentice Hall, Upper Saddle River, NJ, 1996.
- S. Bilbao. *Numerical Sound Synthesis*. John Wiley & Sons, Ltd, Chichester, UK, 2009.
- S. Bilbao. Immersed boundary methods in wave-based virtual acoustics. *J. Acoust. Soc. Am.*, 151(3):1627–1638, 2022.
- S. Bilbao and B. Hamilton. Modeling of complex geometries and boundary conditions in finite difference/finite volume time domain room acoustics simulation. *IEEE Trans. Audio Speech Lang. Process.*, 21(7):1524–1533, 2013.
- S. Bilbao, O. Thomas, C. Touzé, and M. Ducceschi. Conservative numerical methods for the full von kármán plate equations. *Num. Meth. Part. Diff. Eq.*, 31(6):1948–1970, 2015.
- S. Bilbao, B. Hamilton, J. Botts, and L. Savioja. Finite volume time domain room acoustics simulation under general impedance boundary conditions. *IEEE/ACM Trans. Audio Speech Lang. Process.*, 24(1):161–173, 2016.
- J. Botts and L. Savioja. Effects of sources on time-domain finite difference models. *J. Acoust. Soc. Am.*, 136(1):242–247, 2014.
- M. Carvalho, V. Debut, and J. Antunes. Physical modelling techniques for the dynamical characterization and sound synthesis of bells. *Herit. Sci.*, 9:20, 2021. doi: 10.1186/s40494-021-00620-2.

- J. Chabassier and P. Joly. Time domain simulation of a piano. part 2: Numerical aspects. *ESAIM Math. Model. Numer. Anal.*, 49(5):1331–1372, 2015.
- A. Chaigne. On the use of finite differences for the synthesis of musical transients. application to plucked stringed instruments. *J. Phys. IV*, 2 (C5):187–196, 1992.
- A. Chaigne and A. Askenfelt. Numerical simulations of piano strings. i. a physical model for a struck string using finite difference methods. *J. Acoust. Soc. Am.*, 95(2):1112–1118, 1994.
- A. Chaigne, B. Cotté, and R. Viggiano. Dynamical properties of piano soundboards. *J Acoust Soc Am*, 133(4):2456–2466, 2013.
- J. L. Cieřliński. On the exact discretization of the classical harmonic oscillator equation. *J Difference Eq Appl*, 17(11):1673–1694, 2011.
- G. Cohen, P. Joly, J. E. Roberts, and N. Tordjman. Higher order triangular finite elements with mass lumping for the wave equation. *SIAM J. Numer. Anal.*, 32(4):1449–1467, 1995.
- R. Courant, K. Friedrichs, and H. Lewy. On the partial difference equations of mathematical physics. *Math. Ann.*, 100:32–74, 1928.
- J. l.R. d’Alembert. Recherches sur la courbe que forme une corde tendue mise en vibration. *Mem Acad Sci Berlin*, 3:214–219, 1747.
- P. J. Davis. *Interpolation and Approximation*. Dover Publications, Mineola, NY, 1975.
- V. Debut and J. Antunes. When singing bowls don’t sing: A numerical and experimental investigation on the subtle dynamics of tibetan bowls. In *Proc. Int. Symp. Musical Acoust. (ISMA)*, Le Mans, France, 2014.
- V. Doutaut, D. Matignon, and A. Chaigne. Numerical simulations of xylophones. ii: Time-domain modeling of the resonator and of the radiated sound pressure. *J. Acoust. Soc. Am.*, 104(3):1633–1647, 1998.
- M. Ducceschi. An open-source toolbox for direct and inverse modelling of orthotropic plates. *Preprint*, -(–):1–40, 2024. Available online at https://www.researchgate.net/publication/386210529_An_Open-Source_Toolbox_for_Direct_and_Inverse_Modelling_of_Orthotropic_Plates.
- M. Ducceschi and S. Bilbao. Non-iterative, conservative schemes for geometrically exact nonlinear string vibration. In *Proc. Int. Congr. Acoust. (ICA)*, Aachen, Germany, 2019.
- M. Ducceschi and S. Bilbao. Simulation of the geometrically exact nonlinear string via energy quadratisation. *J. Sound Vib.*, 534:117021, 2022.
- M. Ducceschi and C. Touzé. Modal approach for nonlinear vibrations of damped impacted plates: Application to sound synthesis of gongs and cymbals. *J. Sound Vib.*, 344:313–331, 2015a. doi: 10.1016/j.jsv.2015.02.010.

- M. Ducceschi and C. Touzé. Simulations of nonlinear plate dynamics: An accurate and efficient modal algorithm. In *Proc. 18th Int. Conf. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, 2015b.
- M. Ducceschi, C. Touzé, and S. Bilbao. Nonlinear plate vibrations: A modal approach with application to cymbals and gongs. In *Proc. Joint Meeting of the Société Française d'Acoustique and the Institute of Acoustics*, Nantes, France, 2012.
- M. Ducceschi, C. Touzé, and S. Bilbao. Dynamics of the wave turbulence spectrum in vibrating plates: A numerical investigation using a conservative finite difference scheme. *Physica D*, 280–281:73–85, 2014.
- M. Ducceschi, S. Duran, H. Tahvanainen, and L. Ausiello. A method to estimate the rectangular orthotropic plate elastic constants using least-squares and chladni patterns. *Appl. Acoust.*, 220:109949, 2024.
- M. Duruflé, P. Grob, and P. Joly. Influence of gauss and gauss-lobatto quadrature rules on the accuracy of a quadrilateral finite element method in the time domain. *Numer. Methods Partial Differ. Equ.*, 25(3):526–551, 2009.
- N. H. Fletcher and T. D. Rossing. *The physics of musical instruments*. Springer Science & Business Media, 2012.
- M. Géradin and D.J. Rixen. *Mechanical Vibrations: Theory and Applications to Structural Dynamics (Third Edition)*. John Wiley & Sons, Ltd, Chichester, UK, 2015.
- T. Ha-Duong and P. Joly. On the stability analysis of boundary conditions for the wave equation by energy methods. part i: The homogeneous case. *Math. Comput.*, 62(206):539–563, 1994.
- B. Hamilton and S. Bilbao. Optimised 25-point finite difference schemes for the three-dimensional wave equation. In *Proc. Meet. Acoust.*, volume 28, page 015022, 2016.
- R. W. Hamming. *Numerical Methods for Scientists and Engineers*. McGraw-Hill, New York, 2nd edition, 1973.
- Roger A. Horn and Charles R. Johnson. *Matrix Analysis, second edition*. Cambridge University Press, Cambridge, UK, 2013.
- T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1987.
- T. J. R. Hughes and H. S. Tzou. The finite element method in plate bending analysis. *Comput. Struct.*, 7:311–317, 1977.
- P. Joly. Exact boundary conditions for the finite element solution of time-dependent problems. *ESAIM: Math. Model. Numer. Anal.*, 23(3):329–346, 1989.
- P. Joly. Finite element methods with continuous displacement. In Kurt Friedrichs and Hilary Mackenzie, editors, *Effective Computational Methods for Wave Propagation*, pages 247–266. Chapman & Hall/CRC, 2008.

- J. Kirby and M. Sandler. The evolution of drum modes with strike intensity: Analysis and synthesis using the discrete cosine transform. *J. Acoust. Soc. Am.*, 150(1):202–214, 2021.
- K. Kowalczyk and M. van Walstijn. Room acoustics simulation using 3-d compact explicit fdtd schemes. *IEEE Trans. Audio Speech Lang. Process.*, 19(1):34–46, 2011.
- P. Langer, M. Maeder, C. Guist, M. Krause, and S. Marburg. More than six elements per wavelength: The practical use of structural finite element models and their accuracy in comparison with experimental results. *J Comput Acoust*, 25(04):1750025, 2017.
- R.J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations. Steady State and Time Dependent Problems*. SIAM, Philadelphia, USA, 2007.
- E. Maestre, G. P. Scavone, and J. O. Smith. Joint modeling of bridge admittance and body radiativity for efficient synthesis of string instrument sound by digital waveguides. *IEEE/ACM Trans. Audio Speech Lang. Process.*, 25(5):1128–1139, 2017.
- E. Maestre, G. Scavone, and J. O. Smith. Efficient rendering of saxophone sound by modal synthesis and wave scattering. *J. Acoust. Soc. Am.*, 144(3):1752, 2018.
- S. Marburg. Six boundary elements per wavelength: Is that enough? *J Comp Acoust*, 10(01):25–51, 2002.
- M. E. McIntyre, R. T. Schumacher, and J. Woodhouse. On the oscillations of musical instruments. *J. Acoust. Soc. Am.*, 74(5):1325–1345, 1983.
- L. Meirovitch. *Methods of Analytical Dynamics*. McGraw-Hill, New York, 1970.
- J. D. Morrison and J.-M. Adrien. Mosaic: A framework for modal synthesis. *Comput. Music J.*, 17(1):45–56, 1993.
- P. Morse and K.U. Ingard. *Theoretical Acoustics*. Princeton Univ Pr, 1987.
- T. Okuzono and T. Yoshida. High potential of small-room acoustic modeling with 3d time-domain finite element method. *Front. Built Environ.*, 8:1006365, 2022.
- M. Shashkov. *Conservative Finite-Difference Methods on General Grids*. CRC Press, Boca Raton, FL, 1996.
- J. O. Smith. *Physical Audio Signal Processing: For Virtual Musical Instruments and Audio Effects*. W3K Publishing, Palo Alto, CA, 2010. Available online at <https://ccrma.stanford.edu/jos/pasp/>.
- J.O. Smith. *Music Applications of Digital Waveguides*. PhD thesis, Stanford University, 1987.
- F. Soares, J. Antunes, and V. Debut. Multi-modal tuning of vibrating bars with simplified undercuts using an evolutionary optimization algorithm. *Appl. Acoust.*, 173:107704, 2021. doi: 10.1016/j.apacoust.2020.107704.

- J. C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2nd edition, 2004.
- R. Szilard. *Theories and applications of plate analysis*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2004.
- H. Takabatake and Y. Nagareda. A simplified analysis of elastic plates with edge beams. *Comp Struct*, 70(2):129–139, 1999.
- A. Thibault and J. Chabassier. Dissipative time-domain one-dimensional model for viscothermal acoustic propagation in wind instruments. *J. Acoust. Soc. Am.*, 150(2):1165–1175, 2021.
- R. Tournemenne and J. Chabassier. A comparison of a one-dimensional finite element method and the transfer matrix method for the computation of wind music instrument impedance. *Acta Acust. united Acust.*, 105(5):838–849, 2019.
- C. Valette. The mechanics of vibrating strings. *Acustica*, 79(1):105–123, 1995.
- M. van Walstijn, J. Bridges, and S. Mehes. A real-time synthesis oriented tanpura model. In *Proc. Int. Conf. Digital Audio Effects (DAFx-16)*, pages 175–182, 2016.
- M. van Walstijn, V. Chatziioannou, and A. Bhanuprakash. Implicit and explicit schemes for energy-stable simulation of string vibrations with collisions: Refinement, analysis, and comparison. *J Sound Vib*, 569: 117968, 2024.
- J. K. Vandiver. Rayleigh damping and experimental fitting of damping ratios. MIT OpenCourseWare, Engineering Dynamics, Fall 2011, 2011. URL <https://ocw.mit.edu/courses/2-003sc-engineering-dynamics-fall-2011/resources/rayleigh-damping-and-experimental-fitting-of-damping-ratios/>.
- R. Viala, V. Placet, and S. Cogan. Identification of the anisotropic elastic and damping properties of complex shape composite parts using an inverse method based on finite element model updating and 3D velocity fields measurements (FEMU-3DVF): Application to bio-based composite violin soundboards. *Compos Part A-Appl S*, 106:91–103, 2018.
- J. Woodhouse. Linear damping models for structural vibration. *J Sound Vib*, 215(3):547–569, 1998.
- J. Woodhouse. On the synthesis of guitar plucks. *Acta Acust. united Acust.*, 90(6):928–944, 2004.