A guide to numerical dispersion curve calculations: explanation, interpretation and basic MATLAB code

Vanessa Cool[a,b], Elke Deckers[b,c], Lucas Van Belle[a,b], Claus Claeys[a,b,]

[a] *KU Leuven, Department of Mechanical Engineering, Celestijnenlaan 300 - box 2420, Heverlee, Belgium*
[b] *Flanders Make@KU Leuven, Belgium*
[c] *KU Leuven Campus Diepenbeek, Department of Mechanical Engineering, Wetenschapspark 27, 3590 Diepenbeek, Belgium*

Abstract

Dispersion curves or band diagrams play a crucial role in examining, analyzing and designing wave propagation in periodic structures. Despite their ubiquity and current research interest, introductory papers and reference scripting tailored to novel researchers in the field are lacking. This paper aims to address this gap, by presenting a comprehensive educational resource for researchers starting in the field of periodic structures and more specifically on the study of dispersion curves. The objective is twofold. A first objective is to give a detailed explanation of dispersion curves, with graphical illustrations. Secondly, a documented MATLAB code is provided to compute dispersion curves of 3D structures with 2D periodicity using the so-called inverse approach. The dispersion curves are obtained with numerical simulations using the finite element method. The code is written for elastic wave propagation and orthogonal periodicity directions, but can be extended to other types of linear wave propagation, non-orthogonal periodicity directions or 1D and 3D periodicity. The aim of this code is to serve as a starting point for novice researchers in the field, to facilitate their understanding of different aspects of dispersion curves and serve as a stepping stone in their future research.

Keywords: Dispersion Curves, Educational Code, Metamaterials, Periodic Media, MATLAB, Bloch's theorem, Irreducible Brillouin contour

Email address: `claus.claeys@kuleuven.be` (Claus Claeys)

Abbreviations

BMS	Bloch mode synthesis
DOF	Degree-of-freedom
FBZ	First Brillouin zone
FE	Finite elements
GBMS	Generalized Bloch mode synthesis
IBC	Irreducible Brillouin contour
IBZ	Irreducible Brillouin zone
MOR	Model order reduction
PCG	Plane crystallographic group
STL	Sound transmission loss
UC	Unit cell
WFE	Wave finite elements

1. Introduction

This paper is intended for educational use and for researchers who are new to the field of wave propagation in periodic media, particularly when dealing with dispersion curves. The objective of this paper is twofold: to facilitate a more accessible approach to dispersion curve calculations and to offer guidance for students and novice researchers in comprehending and cultivating an intuition for interpreting dispersion curves. To accomplish this dual objective, the paper is organized into two parts: (i) a theoretical and modeling part, supported by ample visualizations, intended to equip the reader with crucial insights into the interpretation of dispersion curves - a perspective that is presently scarce in existing literature; (ii) a documented MATLAB code, providing a tool that can directly be used to explore various case studies and which can be straightforwardly extended. Summarized, the focus of this paper is to discuss and provide the theoretical background, the modeling and the implementation of dispersion curves in an accessible manner, while it is not the purpose to provide a review of all current literature on the topic.

Dispersion curves serve as a frequently employed tool in the examination of wave propagation phenomena. They are visual representations of dispersion relations, which give the relation between frequency and wave number for different waves and wave types. Their popularity has grown significantly, especially in the investigation of intricate wave phenomena, in particular in the context of periodic media. The concept originally emerged in the field of electromagnetism, with seminal works of Floquet [1] and Bloch [2] describing wave propagation in periodic media. Meanwhile the concept has spread to other branches of science such as optics, acoustics and elastodynamics. Periodicity can lead to frequency ranges in which free wave propagation is forbidden, called stop bands or band gaps. Also other complex phenomena such as wave veering, locking and coupling may occur [3–7]. Dispersion curves provide a useful tool to identify and gain insight in such phenomena. Furthermore, as the interest in manipulating and precisely engineering wave phenomena grows, along with the expanding possibilities to do so through concepts like phononic crystals and metamaterials, dispersion curves have become a widespread and go-to tool in the analysis and design of these (often) periodic, complex media.

While very simple systems can be described by analytical dispersion relations, this is no longer the case for the nowadays considered plenitude of complex periodic structures. Instead, a variety of numerical approaches for dispersion curve computation have emerged, including plane wave expansion, multiple scattering, finite difference and finite element (FE) methods. Although the computation of dispersion curves has been discussed to some extent in literature, e.g. [8], currently there is no adequate educational paper that focuses on introducing novice researchers to the interpretation, reading, and numerical generation of dispersion curves. This work aims to close this gap. Given the background of the authors' metamaterial research, the paper approaches the explanation of dispersion curves in the context of elastodynamics using FE modeling. Nevertheless, the introduced concepts are also more generally applicable to other fields of research.

The remainder of this paper is structured as follows. Sec. 2 introduces the modeling of infinite periodic structures. Next, Sec. 3 explains how to go from the dispersion eigenvalue problem to the commonly used dispersion curves. Afterwards, a basic MATLAB code to compute dispersion curves is discussed in Sec. 4 and followed by some case study discussions with results generated using the code in Sec. 5. Afterwards, a brief overview of possible extensions to the code is given in Sec. 6. The main contributions and key points discussed in this paper are summarized in Sec. 7.

2. Modeling infinite periodic structures

Infinite periodic structures can be fully described by a reference unit cell (UC), which is the smallest non-repetitive part. Based on a model of this UC, the dispersion curves for an infinite periodic structure can be calculated. In this section, a brief introduction to Bloch's theorem[1] is given, which is the fundamental theorem that allows studying wave propagation based on a UC. This theorem is afterwards further detailed for the case of FE-based UC modeling.

2.1. Bloch's theorem

This paper focuses on structures with 2D periodicity as schematically represented in Fig. 1. These consists of a UC with dimensions $L_x \times L_y$, indicated in orange in Fig. 1, which periodically repeats in the xy-plane along two orthogonal directions, denoted by $\mathbf{d}_x$ and $\mathbf{d}_y$. Since the structure is periodic, the position of each point P in the structure, denoted as $\mathbf{r}_\mathrm{P}$, can be expressed with respect to the corresponding point U in the reference UC, translated n_x cells along $\mathbf{d}_x$ and n_y cells along $\mathbf{d}_y$:

$$\mathbf{r}_\mathrm{P} = \mathbf{r}_\mathrm{U} + n_x\mathbf{d}_x + n_y\mathbf{d}_y. \tag{1}$$

Bloch's theorem [2, 9] states that wave propagation in an infinite periodic structure can be expressed in terms of the response of a reference UC, and an exponential term

[1]Note that in the literature the terms Bloch's theorem, Bloch-Floquet theorem or Floquet-Bloch principles are all applied. In this work, the term Bloch's theorem is used.

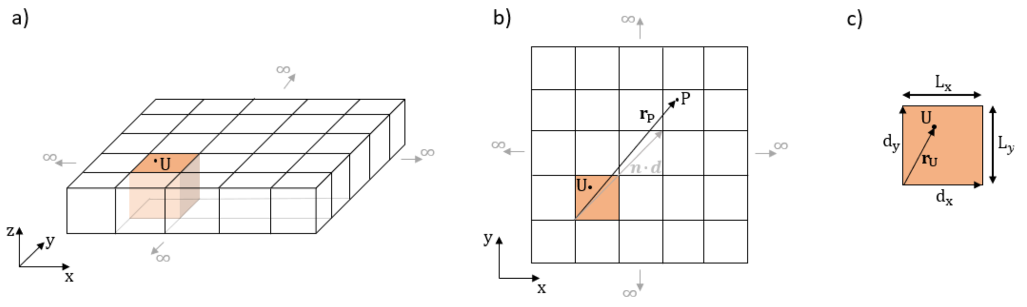


Figure 1: Schematic representation of an infinite 2D periodic structure. a) 3D representation, one UC is highlighted in orange. b) Top view of the infinite periodic structure. c) Corresponding UC, as the fundamental building block of the infinite periodic structure.

defining the relative amplitude and phase change as the wave propagates from one UC to the next. The change in phase and wave amplitude occurring from UC to UC does not depend on the UC location within the periodic structure. Hence, the wave response at a point P of the periodic structure, denoted as $\mathbf{q}(\mathbf{r}_\mathrm{P}, \mathbf{k}, \omega)$ can be expressed as function of the response at the corresponding point U in the reference UC, denoted as $\mathbf{q}_{ref}(\mathbf{r}_\mathrm{U}, \mathbf{k}, \omega)$:

$$\mathbf{q}(\mathbf{r}_\mathrm{P}, \mathbf{k}, \omega) = \mathbf{q}_{ref}(\mathbf{r}_\mathrm{U}, \mathbf{k}, \omega) e^{\mathrm{i}\mathbf{k}\cdot(n_x \mathbf{d}_x + n_y \mathbf{d}_y)}, \tag{2}$$

in which '·' denotes the scalar product. $\mathbf{k} = (k_x, k_y)$ is the (complex) wave vector, composed of the wave numbers in the x- and y-direction, and gives the amplitude decay and phase change of a wave per meter in each direction. In Appendix A, more details are given on how Eq. (2) is obtained. Bloch's theorem can also be written in terms of a vector $\boldsymbol{\mu}$ which consists of the (frequency dependent) wave propagation constants (μ_x, μ_y):

$$\mathbf{q}(\mathbf{r}_\mathrm{P}, \boldsymbol{\mu}, \omega) = \mathbf{q}_{ref}(\mathbf{r}_\mathrm{U}, \boldsymbol{\mu}, \omega) e^{\mathrm{i}\boldsymbol{\mu}\cdot\mathbf{n}}, \tag{3}$$

with

$$\boldsymbol{\mu} = (\mu_x, \mu_y) = (\mathbf{k}\cdot\mathbf{d}_x, \mathbf{k}\cdot\mathbf{d}_y) = (k_x L_x, k_y L_y). \tag{4}$$

The introduction of these propagation constants brings the advantage that wave propagation is expressed in terms of variation per UC instead of per meter. Indeed, the real and imaginary part of the (complex) propagation constants denote, respectively, the relative change in amplitude and the change in phase when moving across one UC in the x- or y-direction.

2.2. FE UC modeling

Considering the wave motion in the infinite periodic structure, Bloch's theorem states that the displacements and forces on the boundaries of a UC are scaled with a factor $e^{\mathrm{i}\mu_x}$ and $e^{\mathrm{i}\mu_y}$ when moving from one UC to the next in the $\mathbf{d}_x$- respectively, $\mathbf{d}_y$-direction. Hence, to model the wave propagation based on a (possibly geometrically complex) UC, a suitable UC modeling strategy is required. Several methods exist to model the UC, i.e. the plane wave expansion, multiple scattering, finite difference and FE methods. Due to its versatility to discretize complex geometries, the FE method is a popular and

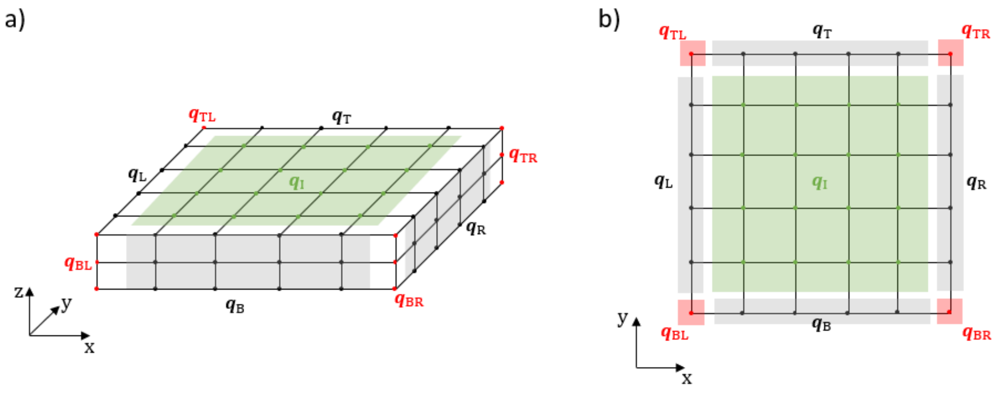


Figure 2: Visualization of the different node groups. a) 3D UC representation discretized with $5 \times 5 \times 2$ elements and notation of the different node groups. b) Corresponding top view.

commonly used method [10]. Using the standard FE method[2], Bloch's theorem can be translated into appropriate periodic boundary conditions on the UC. Fig. 2a shows a UC discretized with $5 \times 5 \times 2$ solid elements. The resulting discrete linear system of equations of the UC with N degrees-of-freedom (DOFs), assuming time-harmonic motion with $e^{\mathrm{i}\omega t}$ dependency, is given here directly in the frequency domain without its time-domain counterpart [10, 11]:

$$(\mathbf{K} - \omega^2\mathbf{M})\mathbf{q} = \mathbf{f}, \tag{5}$$

with ω the radial frequency, $\mathbf{K}$, $\mathbf{M} \in \mathbb{R}^{N\times N}$ the stiffness and mass system matrices, and $\mathbf{q}$, $\mathbf{f} \in \mathbb{R}^{N\times 1}$ the nodal displacement DOFs of the UC and the external nodal forces applied on the UC, respectively. Note that no structural or viscous damping is included here, although including damping is possible [12]. To simplify the application of the periodicity boundary conditions, the UC DOFs $\mathbf{q}$ are partitioned according to the UC boundaries and interior (Fig. 2):

$$\mathbf{q} = \begin{bmatrix}\mathbf{q}_I^T \; \mathbf{q}_L^T \; \mathbf{q}_R^T \; \mathbf{q}_B^T \; \mathbf{q}_T^T \; \mathbf{q}_{BL}^T \; \mathbf{q}_{BR}^T \; \mathbf{q}_{TL}^T \; \mathbf{q}_{TR}^T\end{bmatrix}^T, \tag{6}$$

in which subscripts I, L, R, B, T denote the interior, left, right, bottom and top parts, respectively, while BL, BR, TL and TR are the corner DOFs. By defining[3] $\lambda_x = e^{\mathrm{i}\mu_x}$ and $\lambda_y = e^{\mathrm{i}\mu_y}$, Bloch's theorem results in the following relations for the UC DOFs [14]:

$$\begin{gathered}\mathbf{q}_R = \lambda_x\mathbf{q}_L, \quad \mathbf{q}_T = \lambda_y\mathbf{q}_B, \\ \mathbf{q}_{BR} = \lambda_x\mathbf{q}_{BL}, \quad \mathbf{q}_{TL} = \lambda_y\mathbf{q}_{BL}, \quad \mathbf{q}_{TR} = \lambda_x\lambda_y\mathbf{q}_{BL}.\end{gathered} \tag{7}$$

[2]Although adapted FE methods exist, which start from a modified weak formulation embedding the Bloch theorem, the implementation of this method is less straightforward and thus not considered here.

[3]Note that λ_x and λ_y are sometimes denoted as the propagation constants, e.g. [13], this is however not done in this paper to avoid confusion with μ_x and μ_y.

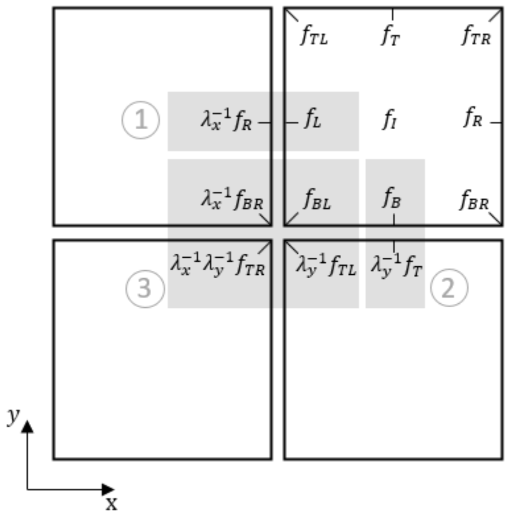


Figure 3: Equilibrium of the generalized forces at ① the left boundary, ② the bottom boundary and ③ the bottom-left corner of consecutive UCs [14].

In matrix-vector notation this reads as follows [13]:

$$\mathbf{q} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \lambda_x \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \lambda_y \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda_x \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda_y \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda_x \lambda_y \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{q}_I \\ \mathbf{q}_L \\ \mathbf{q}_B \\ \mathbf{q}_{BL} \end{bmatrix} = \mathbf{R}\tilde{\mathbf{q}}, \tag{8}$$

in which **R** is the periodicity matrix and $\tilde{\mathbf{q}}$ is the periodic DOF vector. An analogous relation can be written to express the force balance at the left boundary, bottom boundary and bottom-left corner of the UC [14] (Fig. 3):

$$\begin{aligned} &① \quad \mathbf{0} = \mathbf{f}_L + \lambda_x^{-1} \mathbf{f}_R, \\ &② \quad \mathbf{0} = \mathbf{f}_B + \lambda_y^{-1} \mathbf{f}_T, \\ &③ \quad \mathbf{0} = \mathbf{f}_{BL} + \lambda_x^{-1} \mathbf{f}_{BR} + \lambda_y^{-1} \mathbf{f}_{TL} + \lambda_x^{-1} \lambda_y^{-1} \mathbf{f}_{TR}. \end{aligned} \tag{9}$$

The presence of the inverse of λ_x and λ_y stems from the fact that e.g. for the left boundary the force balance reads $\mathbf{f}_R = \lambda_x^{-1} \mathbf{f}_L$ which results in the above equation ①. Eqs. (9) can also be rewritten in matrix-vector notation as follows:

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \lambda_x^{-1}\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \lambda_y^{-1}\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \lambda_x^{-1}\mathbf{I} & \lambda_y^{-1}\mathbf{I} & \lambda_x^{-1}\lambda_y^{-1}\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{f}_I \\ \mathbf{f}_L \\ \mathbf{f}_R \\ \mathbf{f}_B \\ \mathbf{f}_T \\ \mathbf{f}_{BL} \\ \mathbf{f}_{BR} \\ \mathbf{f}_{TL} \\ \mathbf{f}_{TR} \end{bmatrix} = \mathbf{R}'\mathbf{f} = \begin{bmatrix} \mathbf{f}_I \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{10}$$

Note that the periodic force matrix is denoted by $\mathbf{R}'$. Whenever real (μ_x, μ_y) are used, $\mathbf{R}'$ equals the conjugate transpose of $\mathbf{R}$. Substituting Eq. (8) into Eq. (5), pre-multiplying with $\mathbf{R}'$ and assuming that no external forces are applied to the internal DOFs ($\mathbf{f}_I = \mathbf{0}$), the following generalized dispersion eigenvalue problem is obtained [7, 13]:

$$(\tilde{\mathbf{K}} - \omega^2 \tilde{\mathbf{M}})\tilde{\mathbf{q}} = \mathbf{0}, \tag{11}$$

in which:

$$\tilde{\mathbf{K}} = \mathbf{R}'\mathbf{K}\mathbf{R}, \qquad \tilde{\mathbf{M}} = \mathbf{R}'\mathbf{M}\mathbf{R}. \tag{12}$$

By solving this dispersion eigenvalue problem, the dispersion curves can be obtained, as will be further explained in the following section. Note that the corresponding eigenvalue problem contains three unknowns, namely ω, λ_x and λ_y, which can all be complex. As a consequence, different approaches can be followed to solve the eigenvalue problem [13, 15]. The two most commonly applied solution approaches are:

- **Inverse approach** $\omega(\boldsymbol{\mu})$ - real wave propagation constants (μ_x, μ_y) are imposed and the eigenvalue problem is solved toward the frequencies. The method thus considers free wave propagation. The resulting frequencies can generally be real, imaginary or complex representing free, (temporally) evanescent or (temporally) decaying waves, respectively [16]. Since the inverse approach is typically used for undamped UCs, real frequencies will result when imposing real propagation constants. Note that in this manner, stopbands or bandgaps can easily be identified as they correspond to frequency ranges without free wave propagation.

- **Direct approach** $\boldsymbol{\mu}(\omega)$ - real ω are imposed and the eigenvalue problem is solved to the wave propagation constants. Therefore the assumption of time-harmonic wave propagation is made. The resulting wave propagation constants can be real, imaginary or complex representing free, spatially evanescent and spatially decaying wave propagation, respectively. While the implementation of the direct approach is straightforward for 1D periodic problems [17], it becomes cumbersome for 2D periodic problems since two unknowns remain after imposing ω. Different approaches are described in the literature to resolve this [13, 18].

Next to the inverse and direct approach, also mixed approaches exist [15] or approaches which simultaneously solve for complex ω and $\boldsymbol{\mu}$ [19]. In this work, the inverse approach is further applied to obtain the dispersion curves of free wave propagation for undamped structures. Understanding these dispersion curves allows to gain insight which can be further built upon when investigating damped structures, by subsequently making use of the direct approach.

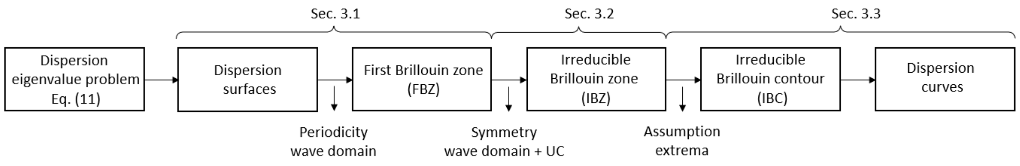

Sec. 3.1
Sec. 3.2
Sec. 3.3
Dispersion eigenvalue problem Eq. (11)
Dispersion surfaces
First Brillouin zone (FBZ)
Irreducible Brillouin zone (IBZ)
Irreducible Brillouin contour (IBC)
Dispersion curves
Periodicity wave domain
Symmetry wave domain + UC
Assumption extrema

Figure 4: Overview of Section 3 on the interpretation and derivation of dispersion curves.

3. Interpreting dispersion curves

This section provides a detailed discussion with graphical explanations on how dispersion curves, as commonly encountered in literature, are obtained. As an example case, the dispersion behavior of a bare plate is used, described by a UC of $0.05 \times 0.05 \times 0.005$ m and with material properties: Young's modulus $E = 210$ GPa, density $\rho = 7800$ kg/m^3 and Poisson's ratio $\nu = 0.3$. To ease the interpretation of the figures, only the out-of-plane bending wave propagation of the plate is considered while the in-plane longitudinal and shear waves are not considered. This is achieved by modeling the plate with plate elements in the FE discretization, which only contain an out-of-plane DOFs.

Fig. 4 gives an overview of the different aspects that are explained in this section. The input is the dispersion eigenvalue problem of Eq. (11), and the section ends with the dispersion curves. The section is divided into three parts: (i) Sec. 3.1 introduces the dispersion surfaces and definition of the first Brillouin zone (FBZ), (ii) Sec. 3.2 discusses the transition from the FBZ to the irreducible Brillouin zone (IBZ), (iii) Sec. 3.3 elaborates on the construction of the irreducible Brillouin contour (IBC) and the generation of the dispersion curves along it.

3.1. Dispersion surfaces and first Brillouin zone

For bending wave propagation in thin isotropic plates, the analytical dispersion relation between wave number k and frequency ω is given by [20]:

$$k^4 = \frac{12\omega^2\rho(1-\nu^2)}{Eh^2} \tag{13}$$

where h is the plate's thickness. Since this relation holds for any propagation direction in the xy-plane, expressing this relation in terms of ω as function of k corresponds to a single paraboloid originating at zero - the dispersion surface of the bending wave propagation. However, in this paper, the aim is to compute the dispersion curves numerically using FE UC modeling. Keeping the above in mind, this section first uncovers how numerically computed dispersion surfaces look like and how they can be understood.

Using the inverse approach, real wave propagation constants (μ_x, μ_y), i.e. free wave propagation, is imposed to the eigenvalue problem, cf. Eq. (11), and correspondingly solved for real frequencies. Solving the eigenvalue problem for different (μ_x, μ_y)-pairs results in surfaces, called the dispersion surfaces. Fig. 5a shows part of the first bending wave dispersion surface, which corresponds to the smallest eigenvalue for each of the imposed (μ_x, μ_y)-combinations covering the region $[0 \quad \pi] \times [0 \quad \pi]$. The reason for choosing this specific region will become clear in the following paragraphs. The resulting surface,

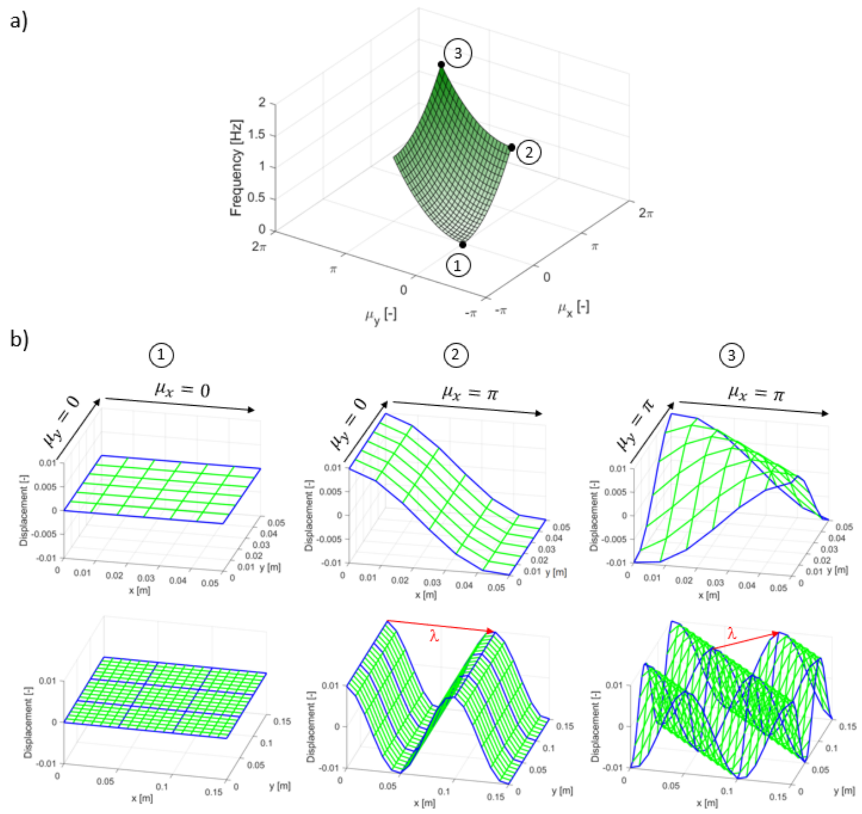


Figure 5: a) First bending wave dispersion surface with b) wave mode visualization for the points ① $\rightarrow (\mu_x = 0, \mu_y = 0)$, ② $\rightarrow (\mu_x = \pi, \mu_y = 0)$, and ③ $\rightarrow (\mu_x = \pi, \mu_y = \pi)$ in the wave space. The wave mode is visualized for one UC as well as for a grid of 3×3 UCs.

depicting frequencies in the wave space (μ_x, μ_y), shows at which frequencies a bending wave can freely propagate through the structure and what the corresponding phase shift per UC will be, being μ_x respectively, μ_y when traveling in the $\mathbf{d}_x$, respectively, $\mathbf{d}_y$ direction. It can be noted that the numerically obtained dispersion surface in this (μ_x, μ_y) range corresponds to a quarter of the paraboloid, which is expected from the analytical bending wave dispersion relation, cf. Eq. (13).

Further insights in the nature of the wave propagation can be gained by also considering the resulting eigenvectors, which correspond to the wave modes. Fig. 5b visualizes the wave modes corresponding to the points ① $\rightarrow (\mu_x = 0, \mu_y = 0)$, ② $\rightarrow (\mu_x = \pi, \mu_y = 0)$, and ③ $\rightarrow (\mu_x = \pi, \mu_y = \pi)$. For each of these points the wave mode is visualized in both one UC as well as in a grid of 3×3 UCs. The wavelength (λ) can also be determined for each solution: e.g., for point ②, the wavelength equals $2L_x$, since $\mu_x = \pi = k_x L_x$ and $\lambda = 2\pi/k_x$ in this case (as $\mu_y = 0$). Physically, $\mu_x = \pi$ means that half a wavelength is imposed over one UC, so a full wavelength covers two UCs. In Fig. 5b, the wave propagation direction and wavelength are shown.

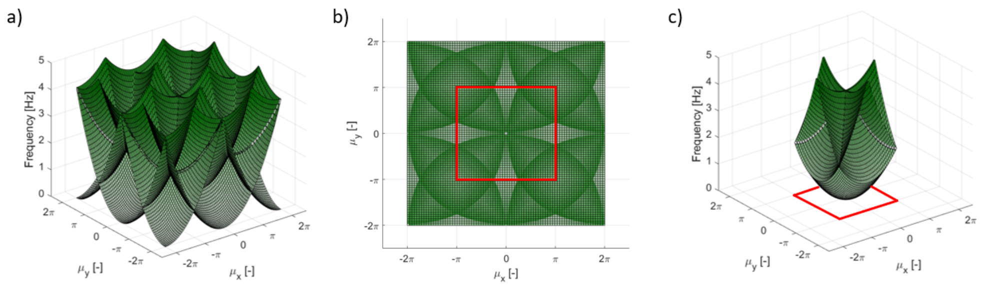


Figure 6: a) Dispersion surfaces for the first three solutions for imposed propagation constants in the domain $[-2\pi \quad 2\pi] \times [-2\pi \quad 2\pi]$, b) Top view of the dispersion surfaces of figure a, c) First three dispersion surfaces within the FBZ $[-\pi \quad \pi] \times [-\pi \quad \pi]$.

Although only one part of the bending wave dispersion surface is shown in Fig. 5, solving Eq. (11) will of course lead to multiple solutions for each (μ_x, μ_y)-pair. For example, when solving for the first three solutions for each (μ_x, μ_y)-pair, three dispersion surfaces (of increasing frequencies) are obtained. When imposing (μ_x, μ_y) in a larger range $[-2\pi \quad 2\pi] \times [-2\pi \quad 2\pi]$, the dispersion surfaces are obtained as shown in Fig. 6a, with a top view in Fig. 6b.

It can be seen that the solutions are 2π-periodic in both μ_x and μ_y and that the dispersion surfaces have multiple intersections. In fact, due to the periodicity of the wave solution, as indicated by Bloch's theorem, also the dispersion surfaces are periodic in the wave space and have periodicity directions which are defined through the periodicity directions of the periodic structure [9]. As a consequence, the dispersion solutions do not need to be calculated for the infinite set of possible (μ_x, μ_y)-pairs, but a confined zone can be defined that entails all dispersion information. Brillouin [9] determined a method for constructing periodicity zones for various periodicity directions, and these zones were later named Brillouin zones. The first Brillouin zone (FBZ) for this case with orthogonal periodicity directions is the domain $[-\pi \quad \pi] \times [-\pi \quad \pi]$, surrounded by the red square in Fig. 6. Hence, all wave dispersion information is contained in Fig. 6c [9].

The periodicity of the dispersion surfaces has been proven by Brillouin [9]. Here, only an intuitive explanation of the proof is given, supported by a graphical representation.

When calculating the dispersion curves, the function $\omega(\boldsymbol{\mu})$ is obtained through enforcing a relationship between the UC boundary DOFs in terms of multiplications with $\lambda_x = e^{\mathrm{i}\mu_x}$ and/or $\lambda_y = e^{\mathrm{i}\mu_y}$. This relationship is, however, insensitive to a shift of the wave propagation constants μ_x or μ_y with a multiple of 2π, e.g. $\lambda_x = e^{\mathrm{i}\mu_x} = e^{\mathrm{i}(\mu_x+2\pi)}$. As a result, a shift with a multiple of 2π in the wave domain along either μ_x or μ_y, will lead to exactly the same solution of the dispersion relations, and the dispersion surface thus should be 2π-periodic in both μ_x and μ_y.

To illustrate this in more detail, the pair of propagation constants $(\mu_x, \mu_y) = (\pi/2, 0)$ and $(\mu_x, \mu_y) = (\pi/2 - 2\pi, 0) = (-3\pi/2, 0)$ is considered in Fig. 7. Both pairs lie on the $(\omega, \mu_y = 0)$-plane and hence for this analysis the diagram of Fig. 7b is used, which is the result of the evaluation of the eigenvalue problem for $\mu_y = 0$. Visually this diagram can be interpreted as a cross section of the earlier obtained dispersion surface along the

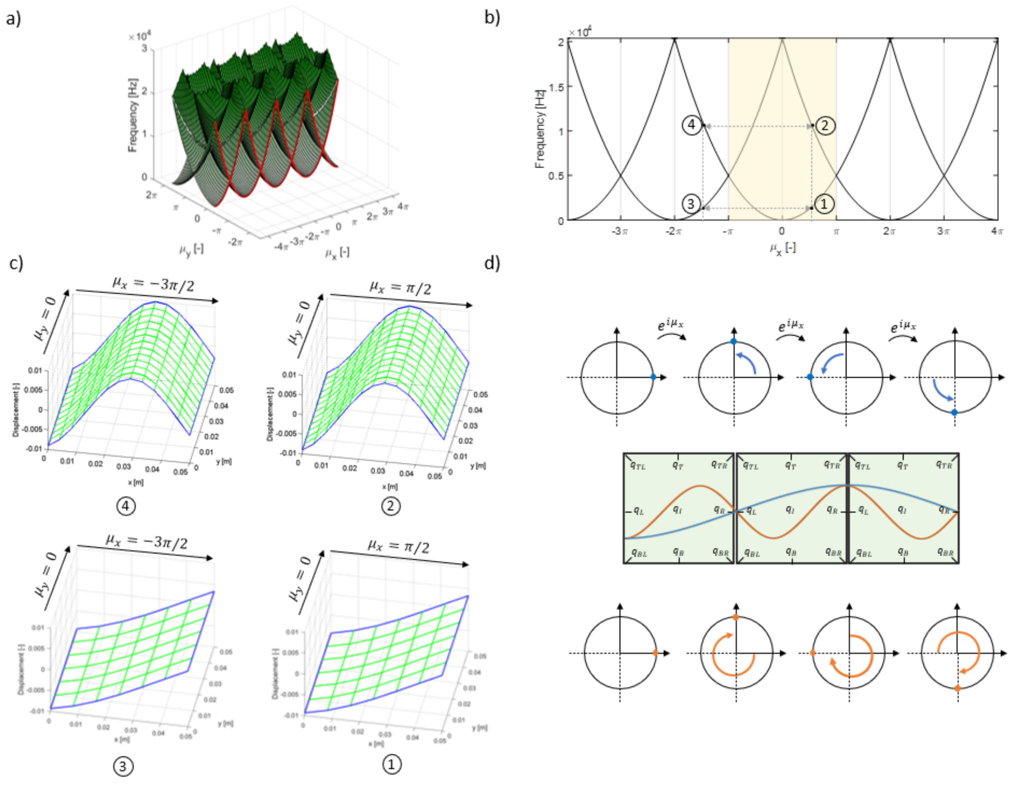


Figure 7: Visualization of periodicity in the wave domain. a) Dispersion surfaces over the domain $[-4\pi \quad 4\pi] \times [0 \quad 2\pi]$, b) 2D cross section of the dispersion surfaces for $\mu_y = 0$ with in yellow the FBZ, c) Wave modes corresponding to the points in subfigure b, d) 1 dimensional representation of the out-of-plane wave mode amplitudes and phase shift across each UC for point ① and ③ (blue) and point ② and ④ (orange).

$(\omega, \mu_y = 0)$-plane, as shown in Fig. 7a, for which the red curves indicates the solutions for $\mu_y = 0$. One can notice, again, the 2π periodicity in the diagram of Fig. 7b: at each 2π-shift with respect to $\mu_x = 0$, a pair of parabolic dispersion curves originates and propagates in positive (right-going wave) and negative (left-going wave) direction.

- When solving the eigenvalue problem for $(\mu_x, \mu_y) = (\pi/2, 0)$, one finds point ① and ② as the first two solutions. Point ① belongs to the dispersion curve starting in the positive direction with as origin $(\mu_x, \mu_y) = (0, 0)$, while it can be seen that point ② belongs to the dispersion curve starting in the negative direction with as origin $(\mu_x, \mu_y) = (2\pi, 0)$.

- When solving the eigenvalue problem for $(\mu_x, \mu_y) = (-3\pi/2, 0)$, one finds points ③ and ④ as the first two solutions. Point ③ is located on the dispersion curve starting in the positive direction with origin $(\mu_x, \mu_y) = (-2\pi, 0)$, while point ④ is located on the dispersion curve starting in the negative direction with as origin $(\mu_x, \mu_y) = (0, 0)$.

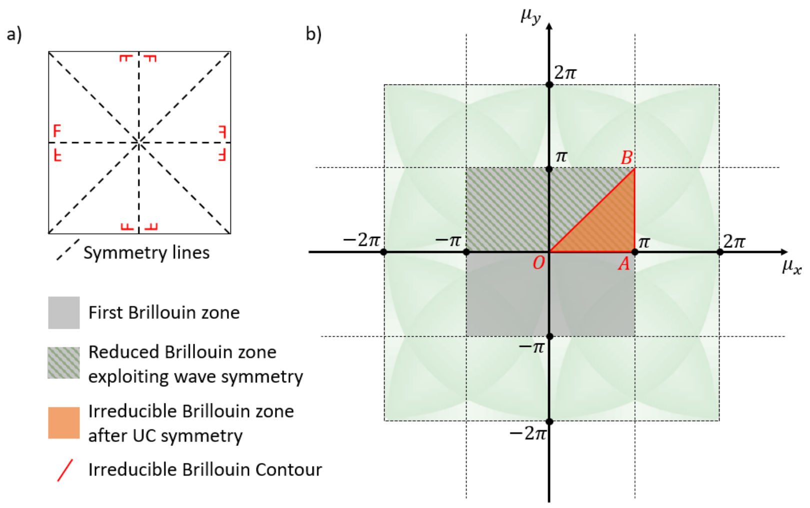

a)
Symmetry lines
First Brillouin zone
Reduced Brillouin zone exploiting wave symmetry
Irreducible Brillouin zone after UC symmetry
Irreducible Brillouin Contour
b)
μ_y
2π
π
B
-2π
$-\pi$
O
A
π
2π
μ_x
$-\pi$
-2π

Figure 8: Example UC belonging to plane crystallographic group p4mm (with 4 symmetry lines) together with a top view of the dispersion surfaces with visualization how to obtain the IBC.

Figure 7c shows the wave mode within one UC for the four derived solutions ①, ②, ③ and ④ together with the corresponding imposed propagation constants. Clearly, solution ① and ③, resp. ② and ④ are identical, and solving only for one of both propagation constants would suffice. When inspecting the wave numbers of the wave modes, intuitively, one might have expected to only find solution ① for propagation constants $(\mu_x, \mu_y) = (\pi/2, 0)$ and solution ④ for propagation constants $(\mu_x, \mu_y) = (-3\pi/2, 0)$ since this would correspond with the wave numbers of the corresponding wave modes. However, since only relations between the boundaries are enforced by the Bloch's theorem, i.e. $\lambda_j = e^{\mathrm{i}\mu_j} = e^{\mathrm{i}(\mu_j+2\pi)}$ with $j = x, y$, one finds identical solutions for $(\mu_x, \mu_y) = (\pi/2, 0)$ and $(\mu_x, \mu_y) = (-3\pi/2, 0)$. Figure 7d schematically explains this further. In this figure, three consecutive UCs in the x-direction are shown and the wave mode for ① (or ③, as they are identical) is represented in 2D with a full blue line whereas the wave mode for ② (or ④) is represented with a full orange line. Above and below the UCs, the wave shifts that occur at each UC boundary are indicated for, respectively, the blue and orange curve. Although the wave amplitude within each UC is different, it is clear that, when only inspecting the wave shift at the boundaries, no distinction can be made between a wave shift $\pi/2$ per UC, i.e., $\mu_x = \pi/2$, and a wave shift $-3\pi/2$ per UC, i.e, $\mu_x = -3\pi/2$, or any other phase shift $\mu_x = \pi/2 \pm 2\pi n$ with $n \in \mathbb{N}$, for that matter. This shows that dispersion surfaces exhibit a 2π-periodicity, that is $\omega(\mu_x, \mu_y) = \omega(\mu_x \pm 2\pi m, \mu_y \pm 2\pi n)$ with $m, n \in \mathbb{N}$, and hence, the information in the FBZ $([-\pi \quad \pi] \times [-\pi \quad \pi])$ indeed contains all the necessary unique dispersion surface results.

3.2. Irreducible Brillouin zone

Although the FBZ $([-\pi \quad \pi] \times [-\pi \quad \pi])$ was shown to contain all dispersion surface information, the wave domain in which the dispersion eigenvalue problem needs to be solved can even be further reduced by exploiting symmetry in (i) the wave propagation

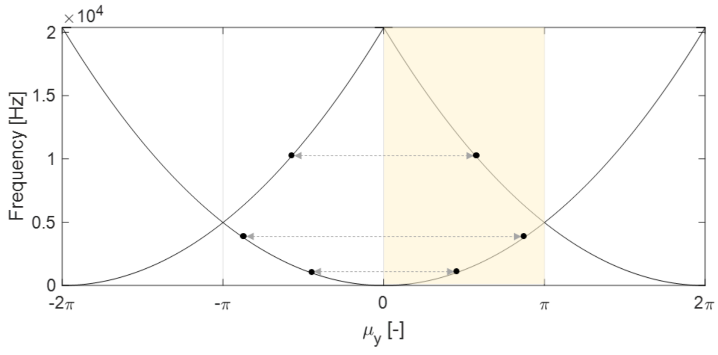


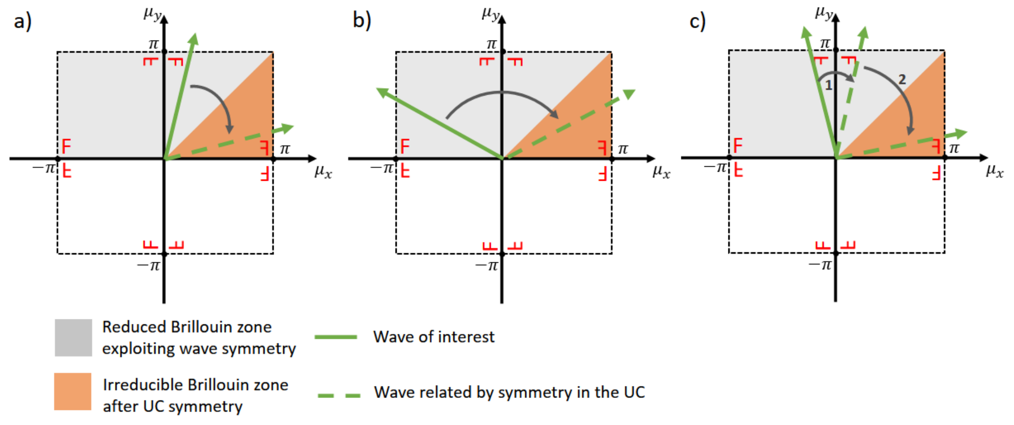
Figure 9: Visualization to show the symmetry between the left and right going waves. 2D dispersion surface plot with connection between right going ($\mu_y = \beta$) and left going ($\mu_y = -\beta$) waves with in yellow the resulting reduced Brillouin zone.

and (ii) the UC geometry. These symmetries lead to symmetries in the wave domain and result in a so-called Irreducible Brillouin zone (IBZ), as illustrated in Fig. 8b for the here considered highly symmetric UC example of Fig. 8a, which has 4 lines of symmetry and belongs to the plane crystallographic group (PCG) p4mm [21]. As a consequence, dispersion surfaces only need to be calculated in the IBZ to characterize the wave propagation in the infinite periodic structure. In the more general case, the shape of the IBZ depends on the type of symmetries present in the UC. To grow a better understanding of how the IBZ is obtained, its construction is next discussed in more detail for the here considered case in two steps: including (i) the wave propagation symmetry and (ii) the symmetry of the UC geometry.

(i) For time-independent linear systems, the properties of a wave propagating along an axis do not depend on the sense of propagation [9]. Hence, the dispersion curves are even functions and only half the FBZ needs to be solved to obtain all dispersion information, visualized in Fig. 9. Although, this reduction can be made along any axis of choice, in this paper it is chosen to use the y-axis and hence reduce the FBZ to the zone $([-\pi \quad \pi] \times [0 \quad \pi])$.

(ii) In case the UC possesses no internal symmetries, the derived reduced Brillouin zone in the previous step will be the IBZ. In case of internal symmetries, the reduced Brillouin zone can be further reduced to its IBZ. Based on the type and combination of symmetries (rotational, glide and mirror symmetry), a classification can be made on the resulting shape of the IBZ [21, 22]. In this work, the example of the highly symmetric UC of Fig. 8a, belonging to PCG p4mm, is considered and the corresponding IBZ is highlighted in orange in Fig. 8b and is defined as the zone surrounded by the wave propagation constants O $\rightarrow (0, 0)$, A $\rightarrow (\pi, 0)$ and B $\rightarrow (\pi, \pi)$.

To understand how this IBZ allows representing all information in the FBZ, three cases of wave propagation are considered in Fig. 10. Each time it will be shown how the considered wave propagation direction, represented by a solid green arrow, can be related to a wave propagation direction studied within the IBZ, represented by a dashed



a)
b)
c)
μ_y
μ_x
π
$-\pi$
1
2
Reduced Brillouin zone exploiting wave symmetry
Irreducible Brillouin zone after UC symmetry
Wave of interest
Wave related by symmetry in the UC

Figure 10: Visualization to illustrate that the IBZ (orange) contains all the required information from within the FBZ. For each wave (solid green) outside the IBZ, a corresponding wave within the IBZ (dashed green) can be found using the geometrical symmetry of the UC (visualized with the red 'F' symbols).

green arrow. Note that in this figure only the direction of the wave is defined, as the argumentation only depends on the ratio and sign of the wave propagation constants (μ_x, μ_y) and not on their absolute amplitude. In this figure, the symmetry of the UC, corresponding to PCG p4mm, which can be mirrored around x, y, and its diagonals, is schematically represented by the addition of a symbol 'F' (cf. Fig. 8).

- Fig. 10a shows the case for a wave in the first quadrant but not in the IBZ. Due to the diagonal symmetry of the UC (Fig. 8a), this wave will contain the same information as the wave represented by the green dotted arrow in the IBZ.
- Fig. 10b, shows the case for a wave in the second quadrant. Due to the vertical symmetry of the UC, this wave will contain the same information as the wave represented by the green dotted arrow in the IBZ.
- Fig. 10c, shows again a case for a wave in the second quadrant. In this case, the green arrow lies in the first quadrant after applying vertical symmetry (step 1), but not yet in the IBZ. Therefore, the diagonal symmetry needs to be applied as well (step 2), after which the projection will end up in the IBZ. After these two steps, it can be seen that the original arrow will contain the same information as the wave represented by the green dotted arrow in the IBZ.
- Lastly, cases could be considered for which the waves lie in the third or fourth quadrant. Each of these cases could be mapped on one of the previously treated cases by considering the symmetry in the wave propagation. For this UC, which also is symmetric around the y-axis, the horizontal symmetry could also be applied instead of the symmetry in the wave propagation.

This indeed illustrates that the IBZ contains all unique information of the FBZ. For a more detailed discussion on UCs with other symmetries, the reader is referred to [21].

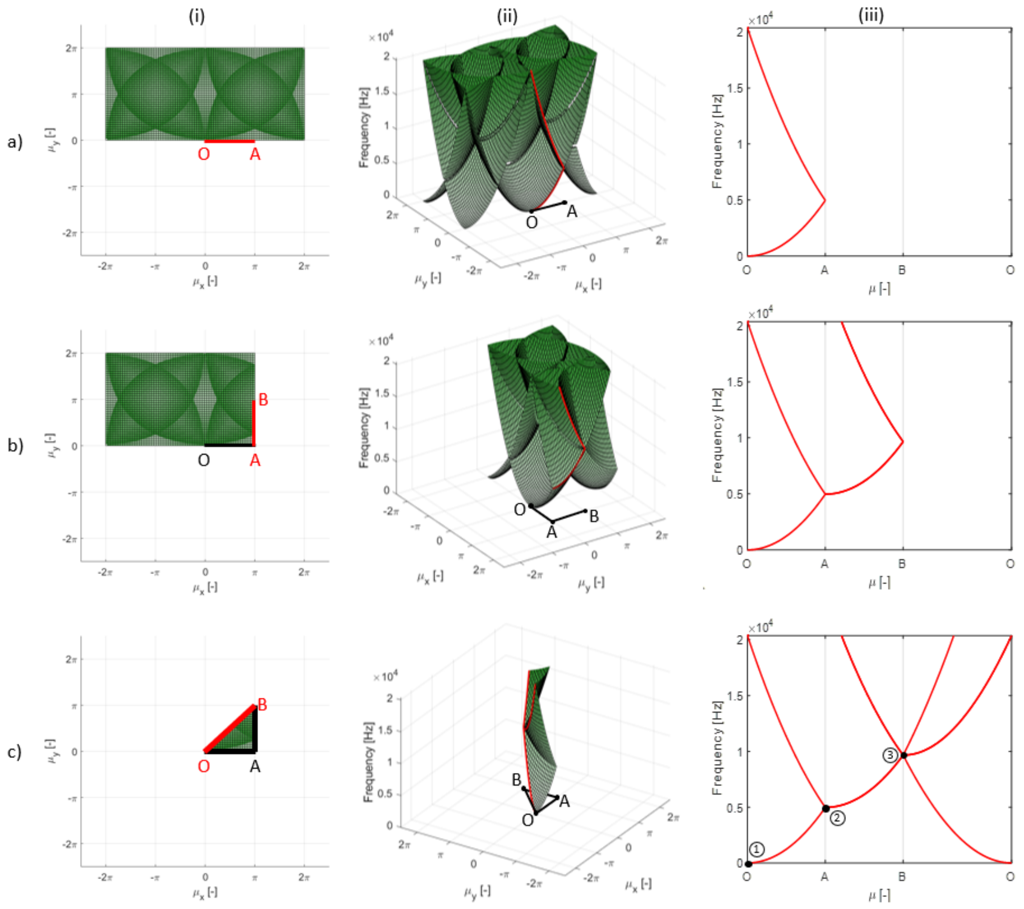


Figure 11: Graphical derivation to go from the dispersion surfaces in the IBZ to the dispersion curves on the IBC. (i) top view of the dispersion surfaces, (ii) 3D view of the dispersion surfaces, (iii) dispersion curves, for respectively, a) section OA, b) section AB and c) section OB.

3.3. Irreducible Brillouin contour

At this point, solving the dispersion eigenvalue problem for imposed wave propagation constants in the IBZ still results in a series of dispersion surfaces, which can take considerable computation time and is not always clear to interpret. Often, one is not interested in the complete dispersion surfaces, but, e.g., only in the minima and maxima on each dispersion surface, as will be shown in an example later.

For this reason, the dispersion surfaces are often only evaluated along the boundary of the IBZ (Fig. 8), named the irreducible Brillouin contour (IBC). This approach is based on the common assumption that all minima and maxima of the dispersion surfaces within the IBZ occur on the boundaries of this IBZ. Although no explicit proof exists, Maurin et al. gave a stochastic proof [21]. Intuitively the evaluation of the solutions on the IBC to find extremes on a dispersion surface can be motivated by the fact that this contour connects the wave propagation constants with a phase difference of π which are the solutions for which (multiple) half wavelength across a UC occur. These points correspond to solutions with a high chance of possible wave interference which enable interesting phenomena, as discussed later.

The remainder of this section gives a visual interpretation of the relationship between the dispersion surfaces in the wave space and the dispersion diagram along the IBC. This with the goal to ease the interpretation of the dispersion diagrams which result from evaluating the eigenvalue problem of Eq. (11) along the IBC. For the highly symmetric UC belonging to PCG p4mm (Fig. 8a), the IBC is denoted by the OABO trajectory consisting of wave propagation constants (μ_x, μ_y):

$$(0,0) \longrightarrow (\pi,0) \longrightarrow (\pi,\pi) \longrightarrow (0,0). \tag{14}$$

The dispersion diagram along this curve can be envisioned as subsequent cuts that are made between the dispersion surfaces and a plane along the OA, AB and BO edges of the IBC. The resulting intersecting curves for OA, AB and BO are next piece-wise plotted on a 2D plot of frequency versus wave propagation constants OABO, effectively establishing the dispersion curves along the IBC. This process is visualized in Fig. 11 for each of the sections OA, AB and BO along the IBC, with the final dispersion curves shown in Fig. 11c.iii. The points ①, ② and ③ from Fig. 5a are indicated in the final dispersion curves as well. Hence, it can be seen that each point on the dispersion curve represents a certain wave propagating freely through the infinite periodic structure along the direction corresponding to the imposed wave propagation constants, and at a certain frequency.

In conclusion, the preceding paragraphs provided an in-detail description of how the evaluation of infinitely large and infinitely many dispersion surfaces, can be limited to the evaluation of dispersion diagrams which capture the main information needed to understand the wave propagation in an infinite periodic structure. Of course, in practice these dispersion curves are computed directly by solving the eigenvalue problem Eq. (11) only for wave propagation constants (μ_x, μ_y) along the IBC. In the next section, a Matlab implementation is discussed which performs exactly this calculation.

4. Matlab implementation

Having established a basic background in calculating and interpreting dispersion curves in the preceding sections, the purpose of this section is to dive into the provided Matlab implementation and explain its details to compute dispersion curves for an FE UC model. The Matlab code is provided in Appendix B and the corresponding Matlab scripts are available on the GitHub repository `github.com/LMSD-KULeuven/2D_InverseUndamped_DispersionCurves`. The code is written from a didactical point of view, with the premise of readability and understandability, not computational efficiency. In what follows, the different sections of the Matlab code are discussed in detail.

4.1. Input data (lines 4-25)

The first section of the code collects the required user inputs which define the problem. A 2D periodic UC is considered which consists of a base plate with dimensions `Lx`, `Ly`, `Lz` in the x-, y- and z-direction, respectively (Fig. 12a). The base plate is assumed to be composed of a linear isotropic elastic material with Young's modulus `E` [Pa], density

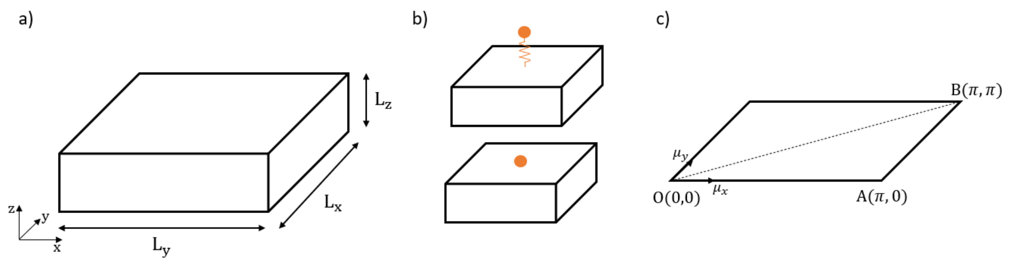


Figure 12: Schematic overview of the inputs. a) UC definition. b) Options for resonator or mass addition to the UC. c) IBC convention.

`rho` [kg/m^3] and Poisson's ratio `v` [-]. Although in the previous section solely bending waves were considered, and consequently plate elements were used, the base plate is discretized in MATLAB using linear elastic hexahedral solid elements due to their generality to represent any topology and 1D, 2D and 3D periodicity. The mesh is defined by the variables `n_elx`, `n_ely`, `n_elz`, which represent the number of elements in the x-, y- and z-direction, respectively. Note that in the code, the FE-part is hard-coded for hexahedral solid elements with 3 DOFs per node, and which are all of equal size. Generally speaking, however, a UC FE mesh can of course be composed of elements of different shapes and sizes. The main requirement for the FE mesh is that the nodes on the opposing sides of the UC (left-right, top-bottom and corners) correspond such that the periodic boundary conditions can be applied, cf. Eqs. (12). Using equal sized elements simplifies the FE implementation and satisfies this requirement.

The possibility to introduce a lumped scatterer in the center of the UC is foreseen in order to obtain a truly periodic structure with associated interesting effects on the dispersion curves. To this end, either a mass-spring resonator with a z-direction displacement DOF or a point mass can be added to the middle of the UC's top face (Fig. 12b) by changing the variable `scatterer`, to `resonator` or `mass`, respectively. If one is interested in computing the dispersion curves of the base plate only, `scatterer` should equal `none`. With `f_res` [Hz] and `m_ratio` [-] the added scatterer can be tuned: `m_ratio` defines the relative mass of the resonator or point mass, as compared to the base plate mass, `f_res` defines the frequency to which the mass-spring system is tuned and is relevant when `scatterer` equals `resonator`.

Before proceeding, the IBC should be defined. Fig. 12c gives the convention which is followed for the 2D periodic case belonging to PCG p4mm as visualized in Fig. 8a. Note that when an uneven number of elements `n_elx`, `n_ely` is chosen, a possible scatterer will not be placed exactly in the middle of the plate's top face so the symmetry will be lost and the IBC needs to be adapted correspondingly. The variable `cont_name` defines the coordinates names of the IBC, which are later used for plotting. The IBC coordinates, consisting of a matrix with the wave propagation constants (μ_x, μ_y) per row, are defined by the variable `cont_co` and are used for the calculation of the dispersion diagrams. For example, when defining the IBC as OABO, the code reads as follows:

```
cont_name = {'O','A','B','O'};
contco =  [0,0;  pi ,0;  pi,pi;  0,0];
```

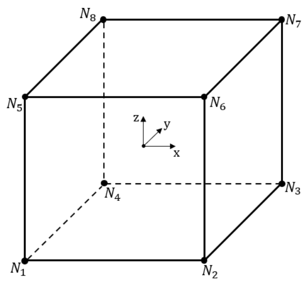


Figure 13: Numbering of the shape functions to derive the element stiffness and mass matrix.

The user can also chose the resolution with which the dispersion curves will be computed, given by the variable `step_size`. A smaller `step_size` means a finer resolution, but also a higher computation cost. Finally, the variable `n_curves` represents the number of computed wave modes per calculation point on the IBC.

4.2. FE preprocessing (lines 27-44)

This next section in the code prepares the FE discretization. First, a number of dependent variables are derived, namely the element sizes in x-, y- and z-direction, denoted by `ax`, `ay` and `az`, respectively. Also information of the UC FE model is derived: `n_elem` the number of elements in the mesh, `n_nodes` the number of nodes, `n_DOFs` the number of DOFs in the FE mesh and `mass_UC` the mass of the UC which is required if a scatterer is added to the base plate. The next step is to compute the element stiffness (`KE`) and mass (`ME`) matrices, which is done by calling the function `KM` (defined on lines 132-167):

```
39  [KE,ME] = KM(E,v,rho,ax,ay,az)
```

Since a structured mesh is used with elements of equal size, only one element stiffness and mass matrix needs to be calculated. Generally, each element will have different element matrices. In the `KM` function, the constitutive matrix (`C`) is first defined. Next, the shape function sequence is given as shown in Fig. 13. `KE` and `ME` are derived by full integration over the element with the symbolic MATLAB toolbox[4] (lines 150-157 and 162-166). This could also be implemented with a numerical integration scheme using e.g. Gauss quadrature [23, 24]. Care should be taken when linear solid elements are used to model thin plates if the elements' aspect ratio differs from 1, since a shear locking phenomenon can occur, resulting in a non-physical stiffness overestimation in bending. To deal with this shear locking, so-called incompatible mode elements are used in this work. The required changes to the element stiffness matrix are computed with the function `FEM_incompatible_modes` on line 158, inspired by the open access code available in [25]. For completeness, the function `FEM_incompatible_modes` is added to the *Supplementary*

[4]For readers without access to the symbolic toolbox of MATLAB, the element stiffness matrix on line 157 can be adapted to the first output kuu of the function FEM_incompatible_modes. The mass matrix should be implemented analogously.

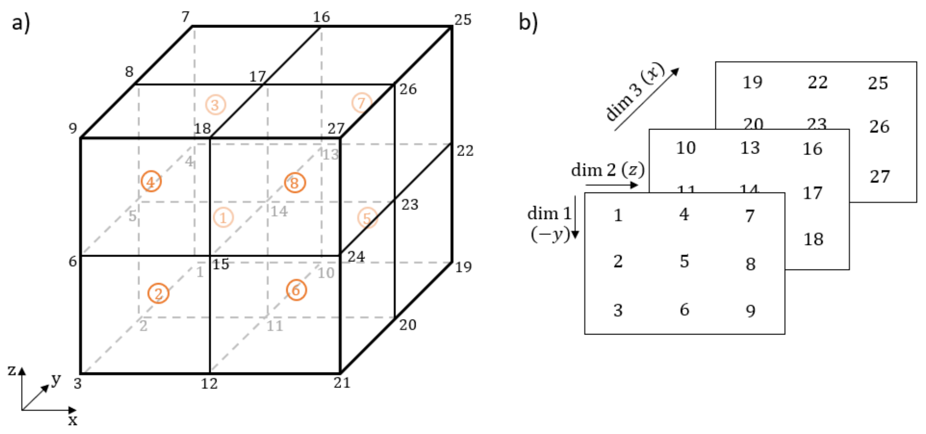


Figure 14: a) Visualization of the elements and node numbering definition. b) Visualization of the variable nodeNrs including the node numbering.

Material of this paper. For more information regarding the derivation of the element matrices, the reader is referred to standard FE reference works, e.g. [23, 24].

Next, the information required for assembling the FE model from the element matrices is derived. This code is based on the work of Ferrari et al. [26]. In Fig. 14a, the numbering of the elements (circled numbers) and nodes (black numbers) is given. The numbering increments according to the $-y$-, z- and then x-direction. The information of the node numbering is included in the `nodeNrs` tensor, which is visualized for a $2 \times 2 \times 2$ mesh in Fig. 14b. This is a three-dimensional tensor in which the first, second and third dimension follow the nodes in the $-y$-, z- and x-direction, respectively. Next, the connectivity matrix `cMat` is derived on lines 42-44 which contains for each element a row with the DOFs in the right order corresponding to the element matrix definition (cf. Fig. 13). For the $2 \times 2 \times 2$ mesh in Fig. 14a, a snapshot of the `cMat` is given by:

$$\text{cMat} = \begin{bmatrix} 4 & 5 & 6 & 31 & 32 & 33 & \dots & 37 & 38 & 39 & 10 & 11 & 12 \\ 7 & 8 & 9 & 34 & 35 & 36 & \dots & 40 & 41 & 42 & 13 & 14 & 15 \\ & & & & & & \vdots & & & & & & \\ 40 & 41 & 42 & 67 & 68 & 69 & \dots & 73 & 74 & 75 & 46 & 47 & 48 \\ 43 & 44 & 45 & 70 & 71 & 72 & \dots & 76 & 77 & 78 & 49 & 50 & 51 \end{bmatrix}. \tag{15}$$

For example, the first row is composed of the DOFs corresponding to the first element. As can be seen in Fig. 14, the nodes $1, 2, 4, 5, 10, 11, 13, 14$ belong to this element. However, following the sequence of the shape functions (Fig. 13), this becomes $2, 11, 10, 1, 5, 14, 13, 4$. Each node contains 3 DOFs, so node 1 corresponds to DOFs $1, 2, 3$, node 2 to DOFs $4, 5, 6$ etc. cMat now contains these DOFs in the right order. Note, only the DOFs for nodes $2, 11, 13, 4$ are visualized here in the first row of cMat to keep the matrix visualization digestible. With the `cMat` construction completed, the full stiffness and mass matrices of the system can be assembled.

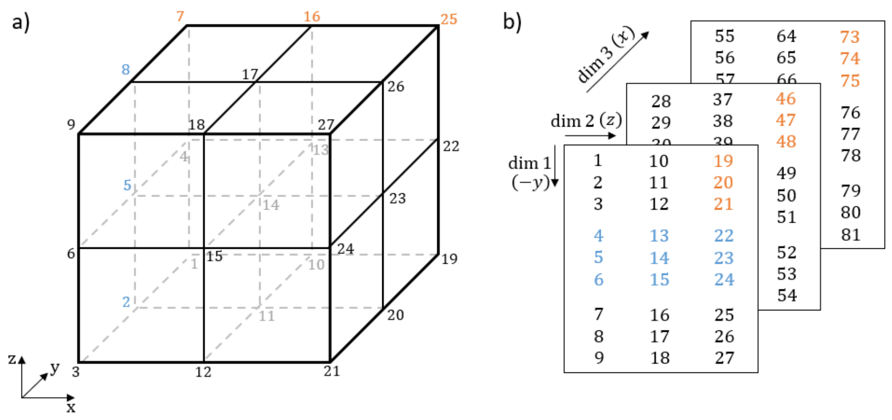


Figure 15: a) Node numbering visualization with two groups colored in blue and orange, b) visualization of the DofNrs tensor with the corresponding DOFs of the colored node groups.

4.3. DOFs partitioning (lines 46-56)

In order to apply the periodicity boundary conditions, a partitioning of the DOFs is required, cf. Eq. (6). This is facilitated by defining the three-dimensional tensor `DofNrs` in a similar way as `nodeNrs`. The `DofNrs` contains an ordering of the DOFs in the $-y$, z, x-direction respectively for dimension one, two and three, as illustrated in Fig. 15b for the $2 \times 2 \times 2$ example of Fig. 15a. This structure provides an easy way to select the DOFs which belong to certain faces or edges. This is visualized by the blue and orange node group in Fig. 15a which results in the blue and orange DOFs highlighted in Fig. 15b. More specifically, the blue nodes correspond to the middle y-node, first x-node and all z-nodes. This leads to the selection of the corresponding DOFs as rows $4-6$ in the first dimension, all z columns in the second dimension and the first face in the third dimension, i.e. `DofNrs(4:6,:,1)`. Analogously, the DOF groups needed in Eq. (6) can be easily derived. Two examples are given here:

```
dofs.L  = DofNrs(nDOF+1:end−nDOF,:,1); dofs.L = dofs.L(:);
dofs.TR = DofNrs(1:nDOF,:,end); dofs.TR = dofs.TR(:);
```

in which the left DOFs `dofs.L` (colored blue in Fig. 15b) are defined as the DOFs corresponding to all nodes except the first and last one according to the y-direction, all nodes in the z-direction and the first node in the x-direction (colored blue in Fig. 15a) . Correspondingly the top-right DOFs are defined as the DOFs corresponding to the first node in the -y-direction, all nodes in the z-direction and the last node in the x-direction.

4.4. System matrix assembly (lines 58-86)

After the FE preprocessing, the required system matrices (`K`,`M`) of the UC FE model are assembled. First the system matrices of the bare plate are derived on lines 60-65, after which the information of the scatterer is added to the system matrices on lines 67-86 in case `scatterer` equals `resonator` or `mass`. The system matrices of the bare plate are assembled using the MATLAB function `sparse`:

```
64 K = sparse(iL,jL,sK);
65 M = sparse(iL,jL,sM);
```

in which `sK` and `sM` contain all the coefficients of the element matrices over all elements in a column vector for the stiffness and mass matrix, respectively. `iL` and `jL` are a set of indices which map the information of `sK` and `sM` to the right position within the full system matrices `K` and `M`, i.e. sK(i) is located in the global matrix $\mathrm{K}(\mathrm{iL}(\mathrm{i}), \mathrm{jL}(\mathrm{i}))$ [26].

Next, the mass or resonator are added to the bare plate system matrices whenever the variable `scatterer` does not equal `none`. For the point mass, this is done in lines 69-71:

```
69 mass = m_ratio*mass_UC;
70 dofM = 3*nodeNrs(floor(end/2),end,floor(end/2));
71 M(dofM,dofM) = M(dofM,dofM) + mass;
```

in which the first line determines the mass of the added point mass. The second line determines the DOF to which the point mass will be added, i.e. here the z DOF of the middle node at the top face of the plate. The corresponding node is easily determined with the `nodeNrs` structure, since now the node is selected in the middle of the y-axis and x-axis (obtained with `floor(end/2)`), while it is the final z-node (obtained with `end`). The purpose of the 3 is to select the z-DOF of the corresponding selected node. The third line finally adds the selected mass to the right position in the mass matrix.

For the resonator case, a mass-spring system is added to the UC, which corresponds to adding a single DOF system. First, the `K` and `M` matrices are extended with one zero row and one zero column to account for the added displacement DOF of the resonator mass. Next, the mass and stiffness of the mass-spring system are determined, whereby the mass is computed analogously to the point mass and the stiffness is computed using this mass and the selected frequency `f_res`:

```
78 k_res = (f_res*2*pi)^2*mass;
```

Similar to the point mass case, the mass-spring system is connected to the z-DOF of the middle node at the top face, denoted as `dofK`. The resonator DOF is added as the last DOF of the extended `K` and `M` matrices, denoted by `dofM`. Finally, a 2×2 system of equations, representing the resonator and its interaction with the base structure, are added at the correct position within the total system matrices:

```
83 K([dofK dofM],[dofK dofM]) = K([dofK dofM],[dofK dofM]) + [k_res −k_res; −k_res k_res];
84 M([dofK dofM],[dofK dofM]) = M([dofK dofM],[dofK dofM]) + [0 0; 0 mass];
```

Finally, on line 85, the new DOF (`dofM`) is added to the interior part of the DOFs partitioning. If the reader would add a resonator at one of the boundaries of the UC, this line should be adapted to the corresponding DOF group.

4.5. Sampling the IBC (lines 88-104)

Before computing the dispersion curves, all (μ_x, μ_y) pairs of interest are required along the IBC. These are sorted in the variable `mu`. Fig. 16 illustrates this for an example IBC of OABO. The code consists of a for-loop (lines 91-104) which loops over the different sections along the IBC. For the example of the OABO contour, these are the sections OA, AB and BO. This for-loop is preceded by initializing the `mu` vector with the first corner point, which corresponds to the first row of the `cont_co` variable. Also a vector `tot_steps` is initialized, which stores the number of steps for each section. This variable

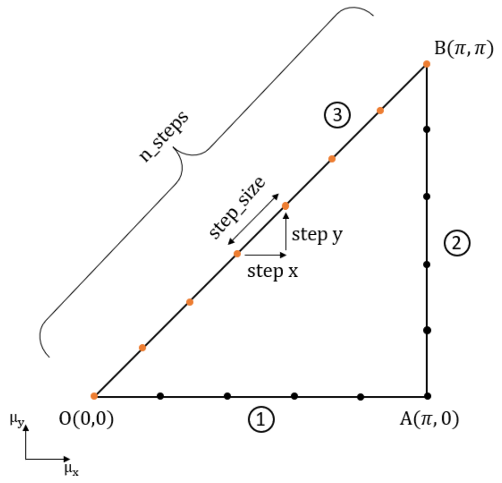


Figure 16: Visualization of how the IBC is sampled in the MATLAB code.

is only used for plotting. In the for-loop, the same computations are repeated for each section. Consider for instance the section BO as an example (Fig. 16). First, the total number of steps in the section is computed, given by `n_steps`:

```
92  n_steps = ceil((sqrt((contco(i,1)-contco(i+1,1))^2+(contco(i,2)-contco(i+1,2))^2))/(
        step_size));
```

The code computes the length of the section and divides it by the `step_size` resolution. Next, the variable `ed` is computed which contains the (μ_x, μ_y) pairs of the particular section, given by the orange dots in Fig. 16. This is done with another for-loop which looks at the μ_x- and μ_y segmentation separately, respectively for `j` equal to one or two:

```
 94  for j = 1:2
 95      step = (contco(i+1,j)-contco(i,j))/(n_steps);
 96      if step == 0
 97          ed(j,:) = contco(i,j)*ones(1,n_steps);
 98      else
 99          ed(j,:) = (contco(i,j)+step):step:contco(i+1,j);
100      end
101  end
```

In this for-loop, first `step` is defined which is the current step-size taken in the μ_x or μ_y orthogonal coordinate space (see Fig. 16). Next, if this step is zero, which occurs in sections OA and AB, the current μ_x or μ_y does not change. If this step is not zero, the vector of μ_x or μ_y are computed on line 99. Finally, the computed `ed` is added to the `mu` variable and the `tot_steps` variable is supplemented. This is done for each section until the entire IBC is discretized. Now that all the to be evaluated (μ_x, μ_y)-pairs are defined, the dispersion curves can be computed.

4.6. Dispersion curve calculation (lines 106-130)

Finally, the dispersion curves are computed along the IBC on lines 108-124. A for-loop is used to loop over the different (μ_x,μ_y)-pairs which are stored in the variable `mu`. Note that the built-in parallel for-loop of MATLAB (`parfor`) can used instead of a regular for-loop to accelerate the computation. For each (μ_x,μ_y)-pair, the dispersion eigenvalue problem, cf. Eq. (11), is solved to obtain the resulting frequencies.

Solving the dispersion eigenvalue problem requires three steps:

(i) On lines 110-116, the periodicity matrix `R`, as defined in Eq. (8), is computed. After initializing an identity matrix, it is filled with the periodicity boundary information, followed by the deletion of the DOFs which are not present in the periodic DOF vector (cf. Eq. (8)).

(ii) The periodicity boundary conditions are imposed via a matrix-multiplication with the periodicity matrix `R` and `R'`, cf. Eq. (12). This is implemented for the stiffness and mass matrices on lines 118 and 119, respectively:

```
118  K_BF=R'*K*R;
119  M_BF=R'*M*R;
```

(iii) Finally the eigenvalue problem, cf. Eq. (11), is solved to the eigenvalues with the built-in sparse MATLAB eigenvalue solver `eigs`:

```
121  [~,s] = eigs(K_BF,M_BF,n_curves,0);
```

in which `n_curves` defines the number of computed eigenvalues and the fourth input indicates the eigenvalues closest to zero are computed. Since the resulting eigenvalues in variable `s` correspond to ω^2, an extra square-root operation allows obtaining the radial frequencies ω, cf. line 123.

After the computations, the dispersion curves are visualized by plotting the frequencies against the evaluated points along the IBC (lines 127-130). A `real` operator is applied to the computed frequencies to delete possible numerical noise which can manifest as spurious imaginary parts of the frequencies. Via the division by 2π, values in [Hz] instead of [rad/s] are obtained. The x-axis labeling uses the variables `tot_steps` and `cont_name` to have a clear visualization of the used IBC.

5. Discussion results

Using the MATLAB code of the previous section, this section aims to discuss some results for the specific case of a bare plate, with a possible addition of a point mass or a mass-spring resonator. These cases are selected based on the fact that such additions lead to interesting changes to the dispersion curves of the bare host structure, relevant for the educational purpose of this paper. While it is not our intention to provide the reader with an in-depth discussion of the physics associated to such structures, we briefly explain the effect of these additions. As mentioned in the introduction, periodic structures can exhibit stop bands, which are frequency zones in which free wave propagation is inhibited. By adding periodic scatterers, e.g. point mass additions, to a host structure, e.g. a plate, a phononic crystal can be obtained, see e.g. [27]. The stop bands in those structures rely on so-called Bragg scattering, where destructive interference occurs between reflected and transmitted waves, and whereby the stop band frequencies are directly linked to the length scale of periodicity. By adding mechanical resonators to a host structure on

UC size [m]	**Material**	**Discretization**	**IBC definition**
`Lx` $= 0.05$	`E` $=$ 210e9 Pa	`n_elx` $= 10$	`cont_name` $=$ O,A,B,O
`Ly` $= 0.05$	`v` $= 0.3$	`n_ely` $= 10$	`cont_co` $=$ [0,0; pi,0; pi,pi; 0,0]
`Lz` $= 0.005$	`rho` $= 7800$ kg/m^3	`n_elz` $= 3$	`step_size` $= 0.01\pi$
		`n_dof` $= 3$	`n_curves` $= 10$

Table 1: Specific inputs for the bare plate used for the results in Sec. 5.

a subwavelength scale, stop bands are obtained due to Fano-type interference between the incoming waves and the waves re-radiated by the resonant cells, see e.g. [28]. The corresponding stop band frequency range is driven by the subwavelength resonators' tuned frequency. Hence, these stop bands do not depend on the periodicity length scale and can be achieved at comparatively lower frequencies without requiring a large spacing, using scales much smaller than the wavelength.

Tab. 1 gives the selected input parameters describing the UC of the bare plate and the definition of the IBC. The point mass is defined by `m_ratio` $= 0.3$, while the mass-spring system is tuned to `f_res` $= 2500$ Hz and the same `m_ratio` $= 0.3$. Fig. 17 shows the obtained dispersion curves. Results are discussed briefly; a detailed discussion for these cases can be found in [29].

Fig. 17a shows the dispersion curves of the bare plate, which is the same case as discussed in Sec. 3. Note that in the code 3D solid elements are used during the FE discretization with 3 DOFs per node. This results in the appearance of longitudinal (L) and shear (S) waves in the dispersion curves, i.e. the steep linear curves indicated in Fig. 17. Note that the added scatterers will have a negligible impact on those wave types in the shown frequency range in what follows.

The frequency for which the bending wavelength in the infinite plate without periodic additions equals twice the UC length in x-direction can be determined analytically using Kirchhoff plate theory:

$$f_{\lambda/2} = \frac{2\pi}{(2L_x)^2}\sqrt{\frac{EL_z^2}{12(1-\nu^2)\rho}}. \tag{16}$$

This correspond with point A where the bending wave dispersion curve folds and is also the frequency for which Bragg interference occurs. For the bare plate discussed above this frequency equals 4933 Hz, which corresponds to the numerically obtained result (Fig. 17a).

The point mass and resonator are scatterers that are explicitly added to control wave propagation. Fig. 17b shows the dispersion curves with the addition of the point mass. A first opening of the dispersion curves occurs exactly at the Bragg interference limit since there destructive interference is possible. The $f_{\lambda/2}$ is therefore the minimum frequency at which phononic crystals will have (partial) bandgaps, as shown in orange. On higher frequencies, other (partial) bandgaps occur as well. Whenever the mass ratio increases, the dispersion curves will open up going from partial bandgaps to full omni-directional bandgaps. The reader can easily check this by increasing the `m_ratio` accordingly.

Fig. 17c shows the result when instead a resonator, tuned to 2500 Hz, is added. Around the tuned frequency, a Fano-type bandgap opens up for the bending waves. Again, the $f_{\lambda/2}$ is important, as resonators have to be added on a subwavelength scale (thus below

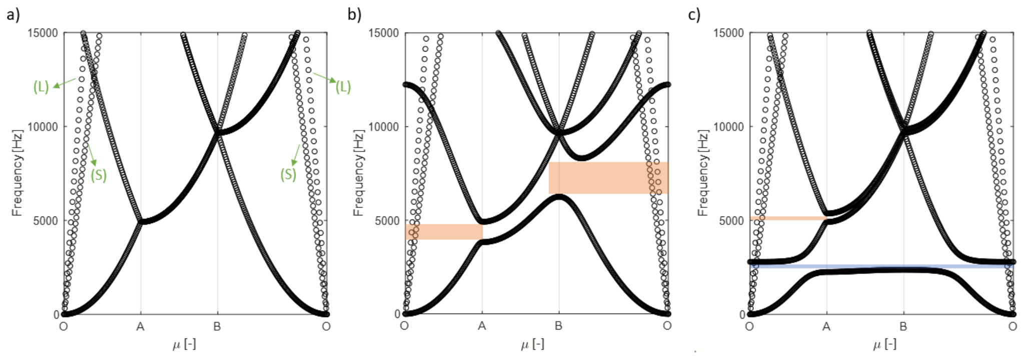


Figure 17: Results obtained with the educational Matlab code with inputs given in Sec. 5. Only the black parts are obtained with the code, while the colored parts are added in post-processing to help the interpretation. a) Dispersion curves of the bare plate with indication of the longitudinal (L) and shear (S) wave modes. b) Dispersion curves when a point mass with mass ratio 0.3 is added. c) Dispersion curves when a mass-spring resonator is added with mass ratio 0.3 and tuned frequency 2500 Hz. In orange, the partial bandgaps for the bending waves are shown. In blue, the full omni-directional bandgap for the bending waves is shown.

this frequency) in order to enable a full omni-directional bandgap of this type [29]. The reader can explore this by changing the `f_res` accordingly. Changing the `m_ratio`, on the other hand, will impact the width of the stop band. It can be noted that, since the resonant additions also act as periodic scatterers, a narrow directional band gap can also be identified around $f_{\lambda/2}$.

6. Extensions

For conciseness, the code is limited to the inverse approach for the computation of dispersion curves of 2D periodic UCs with a possible addition of a point mass or resonator. Starting from this implementation, this last section briefly elaborates on a number of possible extensions, which might be of interest to the reader. No detailed additional code is provided for all extensions, yet sufficient clues and references are given. Subsequently the following aspects are discussed:

1. Wave mode calculation and plotting
2. Adding multiple resonators per UC
3. Computing the dispersion curves with the direct approach
4. Adapting the script from 2D to 1D or 3D periodicity
5. Computing a realizable resonator
6. Accounting for non-orthogonal periodicity axes
7. More comprehensive extensions such as model order reduction (MOR) or vibro-acoustic performance indicator calculations (i.e. sound transmission loss, vibration transmission etc.).

6.1. Wave mode calculation and plotting

In the code, only the eigenvalues of the eigenvalue problem are computed, but not the eigenvectors which correspond to the wave modes. This can be easily extended to also compute and visualize the corresponding wave modes. The wave modes can also be calculated with the built-in function `eigs`. To do so, lines 121 and 122 of the code are adapted to:

```
121 [Q,s] = eigs(K_BF,M_BF,n_curves,0);
122 [s,Idx] = sort(diag(s));
123 modeshapes{i} = Q(:,Idx);
```

The `modeshapes`-variable now contains for each (μ_x, μ_y)-pair the displacement of all periodic DOFs $\tilde{\mathbf{q}}$ corresponding to the eigenvalues. Next, with Eq. (8), the displacement of all DOFs of the UC can be extracted by expanding the wave modes over all UC DOFs.

Plotting these wave modes in MATLAB is straightforward and requires the coordinates of the different nodes. These are easily deduced since a structured hexagonal mesh of equal-sized elements is used. The plotting in MATLAB can be done with the function `scatter` or `patch`. Note that the wave modes can also be plotted over several UCs since the displacement for the consecutive UCs is easily obtained by applying Bloch's theorem using the corresponding values for (μ_x, μ_y), cf. Eq. (3).

6.2. Multiple resonators per UC

In the literature, adding multiple resonators in one UC has been shown to enable several advantages, e.g. [30–32]. When all resonators are represented by idealized mass-spring resonators, these cases can be easily implemented in the MATLAB code. The main adaptation is how the stiffness matrix is adapted when the coefficients for the resonators are added, e.g. lines 73-85. First of all, the obtained `Ks` and `Ms` matrices should be enlarged with additional rows and columns corresponding to the amount of resonators, denoted with `n_Res`, instead of only one:

```
74  K = [K zeros(n_DOFs,n_Res); zeros(n_Res,n_DOFs+1)];
75  M = [M zeros(n_DOFs,n_Res);zeros(n_Res,n_DOFs+1)];
```

Next, the mass, stiffness and attachment DOF for the spring should be defined for each resonator separately, after which their corresponding 2×2 matrices should be added in the UC matrices at the right location. This could be implemented by replacing lines 77-85 in the code with a for-loop over the different resonators:

```
77 for i = 1:n_Res
78     % Define mass and stiffness
79     mass(i) = ...
80     k_res(i) = ...
81     dofK(i) = ...
82     n_DOFs = n_DOFs+1;
83     dofM = n_DOFs;
84     % Add resonator matrices to UC mass, stiffness matrices
85     K([dofK(i) dofM(i)],[dofK(i) dofM(i)]) = K([dofK(i) dofM(i)],[dofK(i) dofM(i)]) + [
           k_res(i) -k_res(i); -k_res(i)  k_res(i)];
86     M([dofK(i) dofM(i)],[dofK(i) dofM(i)]) = M([dofK(i) dofM(i)],[dofK(i) dofM(i)]) + [0
           0; 0 mass(i)];
87     dofs.I = [dofs.I; dofM];
88 end
```

Note that when adding several resonators to the UC at separate locations, the symmetry of the UC can change and the IBC should be adapted accordingly on lines 22 and 23.

6.3. Direct calculation of dispersion curves

Until now, the inverse approach was applied to compute the dispersion curves, whereby the assumption of free wave propagation is made. Whenever the attenuation strength inside a bandgap is of interest, the direct approach $\boldsymbol{\mu}(\omega)$ is typically applied. This approach imposes real frequencies and solves the eigenvalue problem to the (complex) wave propagation constants. The methodology for computing dispersion curves with this approach is elaborated by Mace in [17] and [13], respectively, for 1D periodic and 2D periodic media. In this section, a general overview is given on how to adapt the MATLAB code to compute towards this approach.

First of all, in the input section, the parameter of the dispersion curve `cont_co` should be changed to the frequency range of interest to the user. The corresponding `step_size` is now instead defined as the frequency resolution. The code for the FE discretization and DOFs classification (lines 27-86) remains the same. The sampling of the IBC (lines 88-104) is not required anymore, since now the dispersion eigenvalue problem is solved for given frequencies. The dispersion curve calculation now requires a `for`-loop over the desired frequencies, in which the corresponding eigenvalue problem is constructed for each frequency and subsequently solved to the wave propagation constant. More specifically, with the DOFs classification, sub-matrices can be extracted from the system matrices, e.g.

```
D = K − om^2 M;
D_II = D(dofs.I,dofs.I);
```

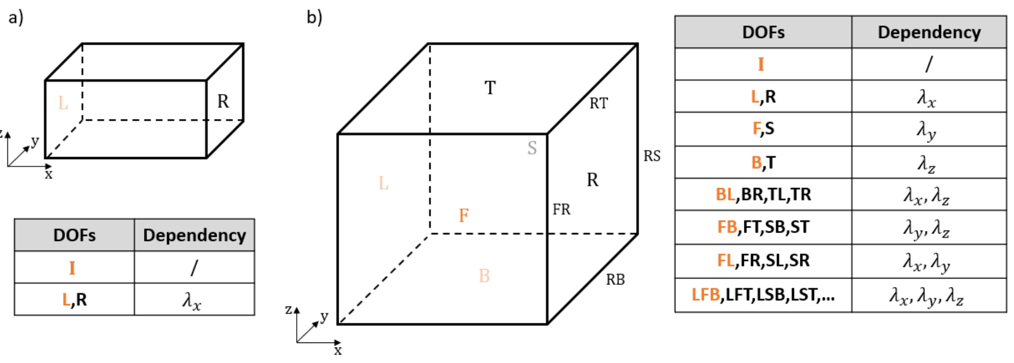
in which `om` is the imposed frequency, `D` is the dynamic stiffness matrix, and `D_II` is the submatrix of `D` which relates to the interior DOFs. The interior DOFs are often first eliminated using dynamic condensation before applying the periodic boundary conditions. More information on the construction of the eigenvalue problem can be found in [17] for 1D periodic media and [13] for 2D periodic media. Note that, for 1D periodic media, after imposing the frequency, only one unknown remains and a quadratic eigenvalue problem needs to be solved. For 2D periodic media, after imposing the frequency, two unknown wave propagation constants remain. Therefore, different strategies have been presented which employ different formulations of the eigenvalue (e.g. polynomial or transcendental).

After solving the eigenvalue problem, the results can be plotted. Again, different visualization strategies are encountered in the literature: (i) a separate plot for the real and imaginary part of the wave numbers with respect to the frequency, e.g. [7, 13, 33], (ii) a 3D representation showing the imaginary part on the x-axis, real part on the y-axis and frequency on the z-axis, e.g. using `scatter3` in MATLAB [12, 33, 34], (iii) a hybrid way, plotting the real part of the wave numbers while coloring the curves according to the imaginary part of the wave number [12, 35].

6.4. 1D, 3D periodicity

While this manuscript explains the dispersion curves and provides code for structures which are periodic in two dimensions, i.e. x- and y-direction, interesting UC designs have been proposed which are 1D or 3D periodic, e.g. [17, 36–38]. For such cases, the code can be adapted as follows:

(i) In the input, mainly the definition of the IBC should be adapted on lines 22 and 23. The `cont_co` matrix should contain the same number of columns as the dimensionality



DOFs	Dependency
I	/
L,R	λ_x

DOFs	Dependency
I	/
L,R	λ_x
F,S	λ_y
B,T	λ_z
BL,BR,TL,TR	λ_x, λ_z
FB,FT,SB,ST	λ_y, λ_z
FL,FR,SL,SR	λ_x, λ_y
LFB,LFT,LSB,LST,...	$\lambda_x, \lambda_y, \lambda_z$

Figure 18: Schematic of the DOF groups and their dependency for the periodic boundary conditions for a) a 1D periodic UC case and b) a 3D periodic UC case.

of the periodicity. Based on the symmetry of the UC under investigation, the correct IBC should also be selected [21, 39].

(ii) The preparation of the FE discretization (lines 27-44) and the assembly of the system matrices (lines 58-65) can remain the same if simple rectangular structures are investigated, meshed with identical elements. Otherwise, for more general and geometrically complex structures, this part should be adapted as described in the previous section. Note that the shear locking adaptations on line 158-159 can be deleted when this is not required.

(iii) While adding the point mass or the mass-spring system, the DOF to which the mass or spring is added can also be adapted. Now this is encoded on line 70 and line 79 as:

```
3*nodeNrs(floor(end/2),end,floor(end/2))
```

which means that the mass or spring is connected to the z-displacement DOF at the node in the middle of the top surface.

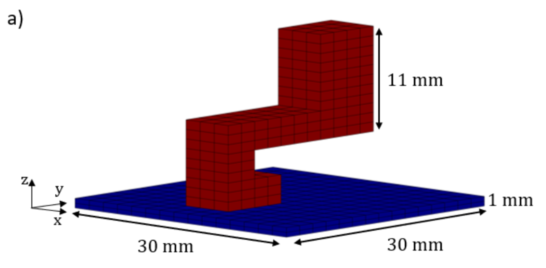
(iv) The structuring of the DOFs should be adapted on lines 47-56. The different DOF groups for the 1D and 3D periodicity are visualized in Fig. 18. For the 1D periodicity this means that lines 50-55 can be deleted while the interior DOFs are now defined as:

```
dofs.I  = DofNrs(:,:,2:end-1); dofs.I = dofs.I(:);
```

For the 3D periodicity, extra relationships between the front and back, between the different edges and between the corner points should be defined, for which the same reasoning as in Sec. 4.3 can be followed.

(v) During the sampling of the IBC (lines 88-104), most of the code can remain the same, apart from the considered dimensionality. On line 92, the difference of `cont_co` between the different directions of periodicity should be taken, i.e. one for the 1D and three for the 3D case. The inner loop should loop across all periodicity directions: `j` equals 1 in the 1D periodic case and should loop from 1 till 3 for the 3D periodic case.

(vi) Finally, during the computation of the dispersion curves, the construction of the `R` matrix (lines 111-115) should be adapted to include all the required periodicity boundary conditions as visualized in Fig. 18. The transformation (lines 118-119) and the computation itself (lines 121-123) remain the same.



b)

	E [GPa]	ρ [kg/m^3]	ν [-]	ax [mm]	ay [mm]	az [mm]
Plate	70	2700	0,3	2	2	0,5
Resonator	7,85	1188	0,3	2	2	1

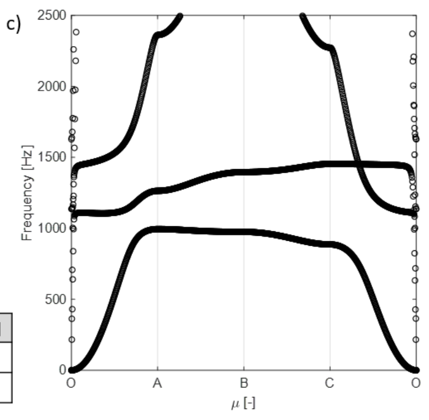


Figure 19: Example of a possible realizable resonator which can be implemented. a) The resonator design in which the different colors indicate the difference in element size and material, listed in the table b). c) The corresponding dispersion curves.

6.5. Realizable resonator

Until now, only bare plates with or without an additional point-mass or ideal mass-spring resonator are included. However, it could be of interest to investigate more complex resonator geometries. An example of a more complex UC design is given in Fig. (19), together with the resulting dispersion curves. This UC is inspired on the work of Van Belle et al. [12]. Due to the generality of the code, this UC (and others) can be implemented while only changing slightly the FE part of the code (lines 27-44 and 58-86). Note, however, that for general complex UC designs, an unstructured body-fitted FE mesh can be of interest, for which the code has to be extended more extensively. All parts corresponding to the FE discretization will need to be extended (lines 27-44, 58-86 and 132-167), while the DOFs partitioning needs to be adapted correspondingly.

In the example of the resonator in Fig. 19a, the structured hexagonal elements can be used with only a small extension to the code. The base plate, shown in blue, and the resonator, shown in red, consist of a different material and different element size, as given in Fig. 19). In the FE analysis this would result in defining two element sizes by adapting lines 29-31, together with obtaining the corresponding element matrices by adapting line 39. Next, the assembly of the full matrices will be more complicated since it needs to take into account the correct placement of the different element matrices which are no longer equal for all elements. Moreover, the `ones(n_elem,1)` command on lines 60 and 61 are not valid anymore since not all possible element positions in the rectangular $L_x \times L_y \times L_z$ domain are filled with material. After the correct `K` and `M` matrices of the UC are obtained, the DOFs structuring, sampling of the IBC and computation of the dispersion curves all remain the same as in the provided MATLAB code.

6.6. Non-orthogonal periodicity axes

In this paper only the case of orthogonal periodicity directions is considered.

In the most general case, a UC is a parallelogon which can take 5 different shapes [39] namely oblique, rectangular, rhombic, square, and hexagonal, with corresponding periodicity directions for each case [21].

For non-orthogonal periodicity directions, the FBZ will become hexagonal shaped and the basis vectors of the direct space ($\mathbf{d_i}$) and the basis vectors of the reciprocal wave space ($\mathbf{e_i}$) are related by $\mathbf{d_i}\mathbf{e_j} = \delta_{ij}$, with δ_{ij} the Kronecker delta function and subscripts i and j taking integer values 1 and 2 for 2D structures [9]. Although this influences somewhat the interpretation of dispersion diagrams, especially since the base vectors $\mathbf{e_1}$ and $\mathbf{e_2}$ in the reciprocal wave space are no longer orthogonal, the reasoning of this paper still applies.

Regarding the Matlab code, the same structure and reasoning can be followed to evaluate the dispersion curves. However, the required adaptations and extensions are more extensive since the simple, cuboid FE discretization cannot be applied anymore. The main parts which should be adapted are lines 27-44, 58-86 and 132-167. This can be implemented more generally with the help of standard FE handbooks [23, 24] or reading in the FE meshes using a commercial preprocessor. Correspondingly, the DOFs partitioning (lines 46-56) should be adapted to the FE mesh. As before, also the coordinates of the IBC will be dependent on the symmetry of the UC [21].

6.7. Elaborate extensions with the code as a starting point

In this section, a few more elaborate extensions are briefly described and appropriate references are listed. Firstly, the computation of the dispersion curves can be computationally demanding, which the reader might discover when experimenting with the code. To alleviate this computational burden, model order reduction (MOR) techniques can be applied. The most commonly used techniques to reduce the computational cost for dispersion curve calculations are the Bloch mode synthesis (BMS) [40] and generalized BMS (GBMS) [41]. Both are model reduction techniques of the Craig-Bampton type, using modal information of the full order UC model, in order to represent it with a less expensive yet approximate reduced model. The BMS technique only reduces the interior UC DOFs, while the GBMS also reduces the boundary UC DOFs. The paper of Krattiger et al. [40] can be followed for the implementation of the BMS and [41] for the GBMS. A brief description on how to the adapt the code is given here for the BMS. Up until the sampling of the IBC (lines 1-104), the basic Matlab implementation remains the same. After this point, the UC DOFs need to be partitioned in to an interior and boundary DOF subset, e.g. by constructing corresponding DOF groups `dofs.I` and `dofs.A`, respectively. Next, the reduction basis is constructed following [40] as:

```
[Phi,~] = eigs(K(dofs.I,dofs.I),M(dofs.I,dofs.I),nRI,0);
Psi = -K(dofs.I,dofs.I)\K(dofs.I,dofs.A);
B = [Phi Psi; zeros(nA,nRI) eye(nA,nA];
```

with `nRI` the number of interior modal DOFs after the reduction, `nA` the (unreduced) number of boundary DOFs, `Phi` the interior normal modes and `Psi` the static constraint boundary modes. Afterwards, the reduced system matrices of the UC are obtained with the appropriate multiplication of the reduction basis `B`:

```
M_red = B.'*M*B;
K_red = B.'*K*B;
```

These reduced UC matrices now should be used when applying the periodicity boundary conditions on line 118 and 119. Note that the construction of the matrix `R` needs to be adapted to the right number and ordering of the DOFs after the reduction. It is noted that for BMS, the amount of boundary DOFs remains te same. For the GBMS, this is not the case and additional care should be taken in renumbering and reordering the DOFs as well as in constructing the matrix `R`.

Next, with the code as a starting point, many FE UC modeling based techniques can be implemented which compute performance indicators other than the dispersion curves. Important is that these techniques employ the wave and finite element (WFE) method which uses the UC of a periodic structure as a reference point and combines the FE technique with the infinite periodic theory. Following calculations are amongst others possible: (i) computation of the forced response of 2D homogeneous media, as studied by Renno et al. [42], (ii) finite structure forced response computation of periodic media as studied by [43], (iii) infinite periodic structure sound transmission loss (STL) e.g. [44–46] etc., (iv) finite structure STL, as studied by Yang et al. [47] and (v) wave mode contributions linking dispersion curves with the infinite periodic structure STL, as studied by Cool et al. [34]. Note that this list is not exhaustive and the reader is referred to the literature to find out more about these and other WFE -based techniques.

7. Conclusion

Understanding and computing dispersion curves can be a daunting challenge for novice researchers. To address this, this manuscript provides a guide and accompanying basic Matlab code for dispersion curve computations.

First, an explanation on how to read and interpret dispersion curves were given together with a Matlab implementation. The focus was put on the inverse approach for dispersion curve calculations of 2D periodic media.

The theoretical part of the paper provided an introduction to the infinite periodic UC modeling. Next, insights were given with a graphically supported explanation on how to read, interpret and derive the dispersion curves. The Matlab code to numerically calculate dispersion curves based on an FE UC description is available at the bottom of this paper, and can be downloaded on the corresponding Github repository (`github.com/LMSD-KULeuven/2D_InverseUndamped_DispersionCurves`). All steps in the code were explained and, where relevant, illustrated graphically. Numerous extensions were provided as inspiration to extend/adapt the code. Users which suggest modifications, extensions or improvements to the code are invited to share them on the public Github repository or by contacting the authors.

Acknowledgments

The research of V. Cool (fellowship no. 11G4421N) is funded by a grant from the Research Foundation - Flanders (FWO). The Research Fund KU Leuven is gratefully acknowledged for its support.

Appendix A. Bloch's theorem

This section gives more details on how Eq. (2) is obtained. A complete discussion on Bloch's theorem can be found in literature, e.g. [2, 9]. In essence, Bloch's theorem governs the wave propagation in periodic media. When a time-harmonic wave propagates through the structure, the theorem states it can be written as:

$$\mathbf{q}(\mathbf{r}, k, \omega) = \tilde{\mathbf{q}}(\mathbf{r}, k) e^{\mathrm{i}\mathbf{k}\cdot\mathbf{r}} e^{\mathrm{i}\omega t}, \tag{A.1}$$

in which $e^{\mathrm{i}\mathbf{k}\cdot\mathbf{r}}$ is a plane wave and $\tilde{\mathbf{q}}$ is a periodic function which has a spatial periodicity due to the underlying periodicity of the structure:

$$\tilde{\mathbf{q}}(\mathbf{r}, k) = \tilde{\mathbf{q}}(\mathbf{r} + n_x \mathbf{d}_x + n_y \mathbf{d}_y, k), \tag{A.2}$$

with n_x and n_y integer numbers denoting the UC positioning. By combining the above two equations, Eq. (2) is obtained, namely:

$$\begin{aligned} \tilde{\mathbf{q}}(\mathbf{r}_P, k, \omega) &= \tilde{\mathbf{q}}(\mathbf{r}_P, k) e^{\mathrm{i}\mathbf{k}\cdot\mathbf{r}_P} e^{\mathrm{i}\omega t} \\ &= \tilde{\mathbf{q}}(\mathbf{r}_U, k) e^{\mathrm{i}\mathbf{k}\cdot\mathbf{r}_U} e^{\mathrm{i}\mathbf{k}\cdot(n_x\mathbf{d}_x + n_y\mathbf{d}_y)} e^{\mathrm{i}\omega t} \\ &= \mathbf{q}_{ref}(\mathbf{r}_U, k, \omega) e^{\mathrm{i}\mathbf{k}\cdot(n_x\mathbf{d}_x + n_y\mathbf{d}_y)}, \end{aligned} \tag{A.3}$$

in which first Eq. (A.1) is used while filling in $\mathbf{r} = \mathbf{r}_P$, next Eq. (A.2) and Eq. (1) are applied, afterwards again Eq. (A.1) is used.

Appendix B. Matlab code

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% INVERSE APPROACH FOR DISPERSION CURVE CALCULATIONS %%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% INPUT DATA (definition of the problem)
% UC size
Lx = 0.05;                             % UC size x-direction [m]
Ly = 0.05;                             % UC size y-direction [m]
Lz = 0.005;                            % UC size z-direction [m]
% Material
E = 210e9;                             % Young's modulus [Pa]
v = 0.3;                               % Poisson ratio [-]
rho = 7800;                            % Mass density [kg/m^3]
% Mesh definition
n_elx = 10;                            % Number of elements in x-dir [-] (>0)
n_ely = 10;                            % Number of elements in y-dir [-] (>0)
n_elz = 3;                             % Number of elements in z-dir [-] (>0)
% Addition to UC bare plate
scatterer = 'none';                    % 'none', 'resonator', 'mass'
f_res = 2500;                          % resonance of resonator [Hz]
m_ratio = 0.3;                         % Scatterer mass ratio vs plate [-]
% Parameters for dispersion curves
cont_name = {'O','A','B','O'};         % Name of the IBC contour
cont_co = [0,0; pi,0; pi,pi; 0,0];     % IBC definition
step_size = 0.01*pi;                   % Resolution propagation constant
n_curves = 10;                         % Number of calculated wave modes

%% FE preprocessing
% Element size
ax = Lx/n_elx;
ay = Ly/n_ely;
az = Lz/n_elz;
% Information UC
n_elem = n_elx*n_ely*n_elz;
n_nodes = (n_elx+1)*(n_ely+1)*(n_elz+1);
nDOF = 3;                             % Number of DOFs per node;
n_DOFs = nDOF*n_nodes;
mass_UC = rho*(n_elx*ax)*(n_ely*ay)*(n_elz*az);
% Element matrices
[KE,ME] = KM(E,v,rho,ax,ay,az);
% FEM information (hard-coded for nDOF = 3)
nodeNrs = reshape(1:n_nodes, 1+n_ely, 1+n_elz, 1+n_elx); % nodes numbering
cMat = reshape(nDOF * nodeNrs(1:n_ely, 1:n_elz, 1:n_elx)+1, n_elem, 1) + ...
      [0,1,2,3*( n_ely+1)*(n_elz+1)+[0,1,2,-3,-2,-1],-3,-2,-1,3*(n_ely + ...
     1)+[0,1,2],3*( n_ely+1)*(n_elz+2)+[0,1,2,-3,-2,-1],3*(n_ely+1)+[-3,-2,-1]]; % connectivity matrix DOFs

%% DOFs partitioning
DofNrs = reshape(1:n_DOFs, nDOF*(n_ely+1), n_elz+1, n_elx+1);
dofs.L  = DofNrs(nDOF+1:end-nDOF,:,1); dofs.L = dofs.L(:);
dofs.R  = DofNrs(nDOF+1:end-nDOF,:,end); dofs.R = dofs.R(:);
dofs.T  = DofNrs(1:nDOF,:,2:end-1); dofs.T = dofs.T(:);
dofs.B  = DofNrs(end-(nDOF-1):end,:,2:end-1); dofs.B = dofs.B(:);
dofs.TL = DofNrs(1:nDOF,:,1); dofs.TL = dofs.TL(:);
dofs.TR = DofNrs(1:nDOF,:,end); dofs.TR = dofs.TR(:);
dofs.BL = DofNrs(end-(nDOF-1):end,:,1); dofs.BL = dofs.BL(:);
dofs.BR = DofNrs(end-(nDOF-1):end,:,end); dofs.BR = dofs.BR(:);
dofs.I  = DofNrs(nDOF+1:end-nDOF,:,2:end-1); dofs.I = dofs.I(:);

%% System matrix assembly
% Assemble bare plate system matrices
iL = reshape(kron(cMat,ones(24,1))',24*24*n_elem,1);
jL = reshape(kron(cMat,ones(1,24))',24*24*n_elem,1);
sK = reshape(KE(:)*(ones(n_elem,1)'),24*24*n_elem,1);
sM = reshape(ME(:)*(ones(n_elem,1)'),24*24*n_elem,1);
K = sparse(iL,jL,sK);
M = sparse(iL,jL,sM);
% Add scatterer (point mass/resonator)
switch scatterer
    case 'mass'
      mass = m_ratio*mass_UC;
      dofM = 3*nodeNrs(floor(end/2),end,floor(end/2));
      M(dofM,dofM) = M(dofM,dofM) + mass;
    case 'resonator'
      % Add extra zero row and column to K, M
      K = [K zeros(n_DOFs,1);zeros(1,n_DOFs) 0];
      M = [M zeros(n_DOFs,1);zeros(1,n_DOFs) 0];
      % Define mass and stiffness
      mass = m_ratio*mass_UC;
      k_res = (f_res*2*pi)^2*mass;
      dofK = 3*nodeNrs(floor(end/2),end,floor(end/2));
      n_DOFs = n_DOFs+1;
      dofM = n_DOFs;
      % Add resonator matrices to bare plate UC system matrices
      K([dofK dofM],[dofK dofM]) = K([dofK dofM],[dofK dofM]) + [k_res -k_res; -k_res k_res];
      M([dofK dofM],[dofK dofM]) = M([dofK dofM],[dofK dofM]) + [0 0; 0 mass];
      dofs.I = [dofs.I; dofM];
end

%% Sampling the IBC
mu = 1i*cont_co(1,:).';
tot_steps = zeros(1,size(cont_co,1));
for i = 1 : size(cont_co,1)-1
    n_steps = ceil((sqrt((cont_co(i,1)-cont_co(i+1,1))^2+(cont_co(i,2)-cont_co(i+1,2))^2))/(step_size));
    ed = zeros(2,n_steps);
    for j = 1:2 % j = 1 -> x; j = 2 -> y
        step = (cont_co(i+1,j)-cont_co(i,j))/(n_steps);
        if step == 0
            ed(j,:) = cont_co(i,j)*ones(1,n_steps);
        else
            ed(j,:) = (cont_co(i,j)+step):step:cont_co(i+1,j);
        end
    end
    mu = [mu, 1i*ed];
    tot_steps(i+1) = tot_steps(i)+n_steps;
end

%% Dispersion curve calculation
omega = zeros(n_curves,size(mu,2));
for i = 1:size(mu,2)
    % Construction of periodicity matrix
    R = eye(n_DOFs,n_DOFs);
    R(dofs.R,dofs.L) = exp(mu(1,i))*eye(length(dofs.R),length(dofs.L));
    R(dofs.T,dofs.B) = exp(mu(2,i))*eye(length(dofs.T),length(dofs.B));
    R(dofs.TL,dofs.BL) = exp(mu(2,i))*eye(length(dofs.TL),length(dofs.BL));
    R(dofs.TR,dofs.BL) = exp(mu(1,i)+mu(2,i))*eye(length(dofs.TR),length(dofs.BL));
    R(dofs.BR,dofs.BL) = exp(mu(1,i))*eye(length(dofs.BR),length(dofs.BL));
    R = sparse(R(:,setdiff(1:n_DOFs,[dofs.R;dofs.T;dofs.TL;dofs.BR;dofs.TR])));
    % Impose periodicity boundary conditions
    K_BF = R'*K*R;
    M_BF = R'*M*R;
    % Compute dispersion curves
    [~,s] = eigs(K_BF,M_BF,n_curves,0);
    [s,~] = sort(diag(s));
    omega(:,i) = sqrt(s);
end

%% Plot
figure
plot(0:tot_steps(end),real(omega(:,:))/(2*pi),'ko','LineWidth',1);
xlabel('Re(\mu) [-]'); ylabel('Frequency [Hz]'); axis tight;
set(gca,'XTick',tot_steps(:),'XTickLabel',cont_name,'XGrid','on')

%% Function of element matrices
function [KE,ME] = KM(E,nu,rho,a,b,c)
syms x y z
C = (1/((1+nu)*(1-2*nu)))*[1-nu,nu,nu,0,0,0;
    nu,1-nu,nu,0,0,0;
    nu,nu,(1-nu),0,0,0;
    0,0,0,(1-2*nu)/2,0,0;
    0,0,0,0,(1-2*nu)/2,0;
    0,0,0,0,0,(1-2*nu)/2];
N1 = (1/(8*a*b*c))*(a-x)*(b-y)*(c-z);
N2 = (1/(8*a*b*c))*(a+x)*(b-y)*(c-z);
```

```
N3 = (1/(8*a*b*c))*(a+x)*(b+y)*(c-z);
N4 = (1/(8*a*b*c))*(a-x)*(b+y)*(c-z);
N5 = (1/(8*a*b*c))*(a-x)*(b-y)*(c+z);
N6 = (1/(8*a*b*c))*(a+x)*(b-y)*(c+z);
N7 = (1/(8*a*b*c))*(a+x)*(b+y)*(c+z);
N8 = (1/(8*a*b*c))*(a-x)*(b+y)*(c+z);

B = [diff(N1,x),0,0, diff(N2,x),0,0, diff(N3,x),0,0, diff(N4,x),0,0, diff(N5,x),0,0, diff(N6,x),0,0, diff(N7,x),0,0, diff(N8,x),0,0;
     0, diff(N1,y),0,0, diff(N2,y),0,0, diff(N3,y),0,0, diff(N4,y),0,0, diff(N5,y),0,0, diff(N6,y),0,0, diff(N7,y),0,0, diff(N8,y),0;
     0,0, diff(N1,z),0,0, diff(N2,z),0,0, diff(N3,z),0,0, diff(N4,z),0,0, diff(N5,z),0,0, diff(N6,z),0,0, diff(N7,z),0,0, diff(N8,z);
     diff(N1,y), diff(N1,x),0, diff(N2,y), diff(N2,x),0, diff(N3,y), diff(N3,x),0, diff(N4,y), diff(N4,x),0, diff(N5,y), diff(N5,x),0,
            diff(N6,y), diff(N6,x),0, diff(N7,y), diff(N7,x),0, diff(N8,y), diff(N8,x),0;
     0, diff(N1,z), diff(N1,y),0, diff(N2,z), diff(N2,y),0, diff(N3,z), diff(N3,y),0, diff(N4,z), diff(N4,y),0, diff(N5,z), diff(N5,y)
            ,0, diff(N6,z), diff(N6,y),0, diff(N7,z), diff(N7,y),0, diff(N8,z), diff(N8,y);
     diff(N1,z),0, diff(N1,x), diff(N2,z),0, diff(N2,x), diff(N3,z),0, diff(N3,x), diff(N4,z),0, diff(N4,x), diff(N5,z),0, diff(N5,x),
            diff(N6,z),0, diff(N6,x), diff(N7,z),0, diff(N7,x), diff(N8,z),0, diff(N8,x)];
KE = int(int(int(B.'*C*B, z,-c,c),y,-b,b),x,-a,a);
KE = E*double(KE);
[~, kua, kaa, kau] = FEM_incompatible_modes(E,nu,a,b,c);
KE = KE - kua*(kaa\kau);
KE = KE/2;

N = [N1,0,0,N2,0,0,N3,0,0,N4,0,0,N5,0,0,N6,0,0,N7,0,0,N8,0,0;
     0,N1,0,0,N2,0,0,N3,0,0,N4,0,0,N5,0,0,N6,0,0,N7,0,0,N8,0;
     0,0,N1,0,0,N2,0,0,N3,0,0,N4,0,0,N5,0,0,N6,0,0,N7,0,0,N8];
ME = int(int(int(N.'*N, z,-c,c),y,-b,b),x,-a,a);
ME = rho*double(ME/8);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% This Matlab code was written in October 2023
%%% by V. Cool, E. Deckers, L. Van Belle and C. Claeys
%%% Department of Mechanical Engineering, 3001 Heverlee, Belgium
%%% Any comments are welcome: claus.claeys@kuleuven.be
%%%
%%% The code is intended for educational purposes.
%%% Theoretical details are discussed in the corresponding paper:
%%% "A guide to numerical dispersion curve calculations:
%%% explanation, interpretation and basic Matlab code"
%%%
%%% This code can be dowloaded from the corresponding Github page:
%%% github.com/LMSD-KULeuven/2D_InverseUndamped_DispersionCurves
%%%
%%%
%%% Copyright (c) 2023 KU Leuven Mecha(tro)nic System Dynamics (LMSD)
%%%
%%% Permission is hereby granted, free of charge, to any person
%%% obtaining a copy of this software and associated documentation
%%% files (the "Software"), to deal in the Software without
%%% restriction, including without limitation the rights to use,
%%% copy, modify, merge, publish, distribute, sublicense, and/or sell
%%% copies of the Software, and to permit persons to whom the
%%% Software is furnished to do so, subject to the following
%%% conditions:
%%%
%%% The above copyright notice and this permission notice shall be
%%% included in all copies or substantial portions of the Software.
%%%
%%% THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
%%% EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
%%% OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
%%% NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
%%% HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
%%% WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
%%% FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
%%% OTHER DEALINGS IN THE SOFTWARE.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

References

[1] G. Floquet, Sur les équations différentielles linéaires à coefficients périodiques, in: Annales scientifiques de l'École normale supérieure, Vol. 12, 1883, pp. 47–88.

[2] F. Bloch, Über die quantenmechanik der elektronen in kristallgittern, Zeitschrift für physik 52 (7-8) (1929) 555–600.

[3] T. Vasileiadis, J. Varghese, V. Babacic, J. Gomis-Bresco, D. Navarro Urrios, B. Graczykowski, Progress and perspectives on phononic crystals, J. Appl. Physics 129 (16).

[4] X. Zhou, X. Liu, G. Hu, Elastic metamaterials with local resonances: an overview, Theor. Appl. Mech. Letters 2 (4) (2012) 041001.

[5] Y. Liu, X. Zhang, Metamaterials: a new frontier of science and technology, Chemical Soc. Rev. 40 (5) (2011) 2494–2507.

[6] C. Lopez, Materials aspects of photonic crystals, Adv. Mat. 15 (20) (2003) 1679–1704.

[7] M. I. Hussein, M. J. Leamy, M. Ruzzene, Dynamics of phononic materials and structures: Historical origins, recent progress, and future outlook, Appl. Mech. Rev. 66 (4) (2014) 040802.

[8] E. Manconi, B. Mace, Modelling wave propagation in two-dimensional structures using a wave/finite element technique, ISVR Techn. Memorandum No: 966.

[9] L. N. Brillouin, Wave propagation in periodic structures, McGraw-Hill Book Company, New York City, 1953.

[10] D. Mead, Wave propagation in continuous periodic structures: research contributions from southampton, 1964–1995, J. Sound Vibr. 190 (3) (1996) 495–524.

[11] D. Mead, A general theory of harmonic wave propagation in linear periodic systems with multiple coupling, J. Sound Vibr. 27 (2) (1973) 235–260.

[12] L. Van Belle, C. Claeys, E. Deckers, W. Desmet, On the impact of damping on the dispersion curves of a locally resonant metamaterial: Modelling and experimental validation, J. Sound Vibr. 409 (2017) 1–23.

[13] B. R. Mace, E. Manconi, Modelling wave propagation in two-dimensional structures using finite element analysis, J. Sound Vibr. 318 (4-5) (2008) 884–902.

[14] R. Langley, A note on the force boundary conditions for two-dimensional periodic structures with corner freedoms, J. Sound Vibr. 167 (2) (1993) 377–381.

[15] B. R. Mace, Discussion of "dynamics of phononic materials and structures: historical origins, recent progress and future outlook", Appl. Mech. Rev. 66 (4) (2014) 045502.

[16] E. Manconi, B. R. Mace, Estimation of the loss factor of viscoelastic laminated panels from finite element analysis, J. Sound Vibr. 329 (19) (2010) 3928–3939.

[17] B. R. Mace, D. Duhamel, M. J. Brennan, L. Hinke, Finite element prediction of wave motion in structural waveguides, J. Acoust. Soc. Am. 117 (5) (2005) 2835–2843.

[18] L. Meng, Z. Shi, Z. Cheng, A new perspective for analyzing complex band structures of phononic crystals, J. Appl. Phys. 123 (9).

[19] M. J. Frazier, M. I. Hussein, Generalized bloch's theorem for viscous metamaterials: Dispersion and effective properties based on frequencies and wavenumbers that are simultaneously complex, Comptes Rendus Physique 17 (5) (2016) 565–577.

[20] F. J. Fahy, Sound and structural vibration: radiation, transmission and response, Elsevier, 2007.

[21] F. Maurin, C. Claeys, E. Deckers, W. Desmet, Probability that a band-gap extremum is located on the irreducible brillouin-zone contour for the 17 different plane crystallographic lattices, Int. J. Solids and Structures 135 (2018) 26–36.

[22] A. Cracknell, Tables of the irreducible representations of the 17 two-dimensional space groups and their relevance to quantum mechanical eigenstates for surfaces and thin films, Thin Solid Films 21 (1) (1974) 107–127.

[23] O. C. Zienkiewicz, R. L. Taylor, The finite element method for solid and structural mechanics, Elsevier, 2005.

[24] R. D. Cook, et al., Concepts and applications of finite element analysis, John Wiley & Sons, 2007.

[25] A. Bower, Applied mechanics of solids: Demonstration FEA codes (2012).
URL `https://solidmechanics.org/FEA.php`

[26] F. Ferrari, O. Sigmund, A new generation 99 line matlab code for compliance topology optimization and its extension to 3d, Struct. Multidiscipl. Opt. 62 (2020) 2211–2228.

[27] M. Sigalas, E. N. Economou, Band structure of elastic waves in two dimensional systems, Solid state communications 86 (3) (1993) 141–143.

[28] C. Goffaux, J. Sánchez-Dehesa, A. L. Yeyati, P. Lambin, A. Khelif, J. Vasseur, B. Djafari-Rouhani, Evidence of fano-like interference phenomena in locally resonant materials, Phys. Rev. letters 88 (22) (2002) 225502.

[29] C. Claeys, K. Vergote, P. Sas, W. Desmet, On the potential of tuned resonators to obtain low-frequency vibrational stop bands in periodic panels, J. Sound Vibr. 332 (6) (2013) 1418–1436.

[30] C. Claeys, E. Deckers, B. Pluymers, W. Desmet, A lightweight vibro-acoustic metamaterial demonstrator: Numerical and experimental investigation, Mech. Systems and Signal Proc. 70 (2016) 853–880.

[31] C. Claeys, N. G. R. de Melo Filho, L. Van Belle, E. Deckers, W. Desmet, Design and validation of metamaterials for multiple structural stop bands in waveguides, Extreme Mech. Letters 12 (2017) 7–22.

[32] S. Janssen, L. Van Belle, N. G. R. de Melo Filho, W. Desmet, C. Claeys, E. Deckers, Improving the noise insulation performance of vibro-acoustic metamaterial panels through multi-resonant design, Appl. Acoust. 213 (2023) 109622.

[33] A. Krushynska, V. Kouznetsova, M. Geers, Visco-elastic effects on wave dispersion in three-phase

acoustic metamaterials, J. Mech. Phys. Solids 96 (2016) 29–47.

[34] V. Cool, R. Boukadia, L. Van Belle, W. Desmet, E. Deckers, Contribution of the wave modes to the sound transmission loss of inhomogeneous periodic structures using a wave and finite element based approach, J. Sound Vibr. 537 (2022) 117183.

[35] Y.-F. Wang, Y.-S. Wang, V. Laude, Wave propagation in two-dimensional viscoelastic metamaterials, Phys. Rev. B 92 (10) (2015) 104110.

[36] Z. Liu, X. Zhang, Y. Mao, Y. Zhu, Z. Yang, C. T. Chan, P. Sheng, Locally resonant sonic materials, Science 289 (5485) (2000) 1734–1736.

[37] T. Delpero, S. Schoenwald, A. Zemp, A. Bergamini, Structural engineering of three-dimensional phononic crystals, J. Sound Vibr. 363 (2016) 156–165.

[38] S. Sorokin, O. Ershova, Plane wave propagation and frequency band gaps in periodic plates and cylindrical shells with and without heavy fluid loading, J. Sound Vibr. 278 (3) (2004) 501–526.

[39] C. Kittel, Introduction to solid state physics, John Wiley & Sons, inc, 2005.

[40] D. Krattiger, M. I. Hussein, Bloch mode synthesis: Ultrafast methodology for elastic band-structure calculations, Phys. Rev. E 90 (6) (2014) 063306.

[41] D. Krattiger, M. I. Hussein, Generalized bloch mode synthesis for accelerated calculation of elastic band structures, J. Comp. Physics 357 (2018) 183–205.

[42] J. M. Renno, B. R. Mace, Calculating the forced response of two-dimensional homogeneous media using the wave and finite element method, J. Sound Vibr. 330 (24) (2011) 5913–5927.

[43] L. Van Belle, C. Claeys, W. Desmet, E. Deckers, Fast vibro-acoustic response computations for finite periodic metamaterial plates using a generalized bloch mode synthesis based sub-structuring approach, Front. Mech. Eng. 8 (2022) 108.

[44] E. Deckers, S. Jonckheere, L. Van Belle, C. Claeys, W. Desmet, Prediction of transmission, reflection and absorption coefficients of periodic structures using a hybrid wave based–finite element unit cell method, J. Comp. Phys. 356 (2018) 282–302.

[45] A. Parrinello, G. Ghiringhelli, Transfer matrix representation for periodic planar media, J. Sound Vibr. 371 (2016) 196–209.

[46] Y. Xiao, J. Cao, S. Wang, J. Guo, J. Wen, H. Zhang, Sound transmission loss of plate-type metastructures: Semi-analytical modeling, elaborate analysis, and experimental validation, Mech. Systems and Signal Proc. 153 (2021) 107487.

[47] Y. Yang, M. J. Kingan, B. R. Mace, A wave and finite element method for calculating sound transmission through rectangular panels, Mechanical Systems and Signal Processing 151 (2021) 107357.