ElasticMatrix: A MATLAB Toolbox for Anisotropic Elastic Wave Propagation in Layered Media

A Preprint

Danny R. Ramasawmy
Department of Medical Physics
and Biomedical Engineering
University College London
London, UK

Ben T. Cox
Department of Medical Physics
and Biomedical Engineering
University College London
London, UK

Bradley E. Treeby
Department of Medical Physics
and Biomedical Engineering
University College London
London, UK

September 27, 2019

Abstract

Simulating the propagation of elastic waves in multi-layered media has many applications. A common approach is to use matrix methods where the elastic wave-field within each material layer is represented by a sum of partial-waves along with boundary conditions imposed at each interface. While these methods are well-known, coding the required matrix formation, inversion, and analysis for general multi-layered systems is non-trivial and time-consuming. Here, a new open-source toolbox called `ElasticMatrix` is described which solves the problem of acoustic and elastic wave propagation in multi-layered media for isotropic and transverse-isotropic materials where the wave propagation occurs in a material plane of symmetry. The toolbox is implemented in MATLAB using an object oriented programming framework and is designed to be easy to use and extend. Methods are provided for calculating and plotting dispersion curves, displacement and stress fields, reflection and transmission coefficients, and slowness profiles.

Keywords partial-wave method · global matrix method · elastic waves

1 Motivation and Significance

Matrix models of wave propagation in multi-layered elastic solids have had a significant contribution to research areas such as acoustics, geophysics and electromagnetics. A few examples include: structural health monitoring [1], characterisation of interface bonding [2], detection of debonding in joints [3], measuring material properties [4], designing composite layered structures [5], mode sorting of guided waves [6], the physical interpretation of guided wave structures [7], modelling the directional response of Fabry-Pérot ultrasound sensors [8], reflection and transmission of plane waves [9], elastography of layered soft tissues [10], and ice detection on wind turbines [11].

Matrix methods, in particular the partial-wave and global matrix method, represent the stress and displacement fields as a sum of partial-waves for each material of the layered-structure. Each partial-wave represents an upward or downward travelling (quasi-)compressional or (quasi-)shear wave. By invoking boundary conditions at the interfaces of adjacent layers, the partial-wave amplitudes and field properties of the first layer can be related to the last in the form of a 'global' matrix. The resulting matrix equation can be used in two different ways. Firstly, the roots of the equation can be found which give the modal solutions or dispersion curves. Secondly, a subset of partial-wave amplitudes can be defined and the remaining amplitudes solved for. This can be used to calculate the displacement and stress fields within the multi-layered structure when a plane wave is incident. This method will be discussed further in Section 2.

Despite its usefulness, there are few available implementations of the partial-wave method. The current state-of-the-art implementation is Disperse [12]. This software has been in development since 1990 and is primarily focused on calculating the dispersion solutions for multi-layered structures. The Disperse software was originally based the partial-wave method, however, is currently being updated to use the spectral collocation method [12, 13, 14, 15, 16]. The main limitation with Disperse is that it is closed-source. For this reason it is not easily adaptable for applications

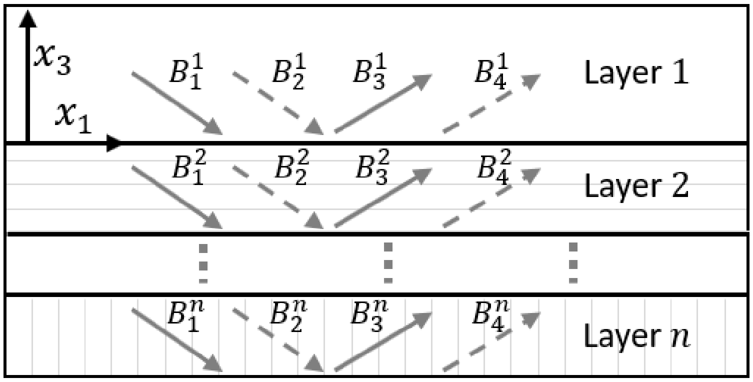
x_3
x_1
B_1^1 B_2^1 B_3^1 B_4^1
Layer 1
B_1^2 B_2^2 B_3^2 B_4^2
Layer 2
B_1^n B_2^n B_3^n B_4^n
Layer n

Figure 1: Diagram for an n-layered elastic medium. In the 2D plane there are four partial-waves with amplitude B_i^n, these represent (quasi-)compressional (solid arrows) and (quasi-)shear (dashed arrows) waves travelling upwards and downwards in each layer.

that are not dispersion analysis, for example, extracting reflection coefficients or slowness profiles. There currently is only a single open-source code modelling the partial-wave method, (LAMB [17]), however, this is limited to modelling only an isotropic plate.

In this paper, a new open-source toolbox called `ElasticMatrix` is introduced which uses the partial-wave method for multi-layered structures with an arbitrary number of isotropic and transverse-isotropic materials. Where possible, it is validated against existing literature and has been implemented so that it is both easy to use and extend. Some potential uses of this software are: 1) plotting the slowness profiles of materials, 2) determining the reflection and transmission coefficients of multi-layered structures, 3) finding the dispersion curves of multi-layered structures, 4) plotting the displacement and stress fields, 5) extending the toolbox for other applications, for example modelling the directional response of Fabry-Perot ultrasound sensors [8, 18].

A brief overview of the underlying mathematical model is described in Section 2. A selection of code snippets and examples are shown in Section 3. (More extensive examples are available with the toolbox documentation.) The impact and conclusions are described in Section 4.

2 Model Description

2.1 Overview

`ElasticMatrix` uses the partial-wave method to model wave propagation in multi-layered elastic solids. The method describes elastic plane-wave propagation along a plane of symmetry for n-layers of rigidly bonded transverse-isotropic materials. An example of an isotropic material is glass, where the material properties are the same when measured from every direction. An example of a transverse-isotropic material is a bundle of fibres, where the properties have translational symmetry axially along the fibre, and are isotropic in the plane perpendicular to this. `ElasticMatrix` can model layered transverse-isotropic materials if they are aligned such that they have rotational symmetry about the axis perpendicular to the plane of each layer and the wave-vector of the propagating wave lies in the plane of symmetry. In this case, the multi-layered structure can be modelled in two-dimensions. This is illustrated in Fig. 1.

Each partial-wave represents the superposition of waves that have been multiply reflected or transmitted at the interfaces between each layer in a steady-state. The polarisation vector and wave-vector of each of these partial-waves can be found from the Christoffel Equation which is described in Section 2.2. The degree of reflection and transmission depends on the boundary conditions at the interfaces and material properties of each layer. The coupled equations that arise from the boundary conditions can be combined into a ‘global-matrix’ which allows them to be solved simultaneously, which is discussed in Section 2.3. This global matrix approach can be used to tackle various problems in elastic wave propagation. For example, the singularities of the global matrix give the dispersion curves, and by specifying an incident wave, the resulting wave-field throughout the structure can be calculated. More detailed descriptions of the partial-wave and global-matrix method can be found in [4, 5, 19, 20, 21, 22, 23, 24].

2.2 Wave-vectors and Polarisation

Firstly, the solution for a plane wave propagating in an unbounded medium is derived. This is needed to calculate the polarisation and wave-vectors for each partial-wave component and the process is repeated independently for every layer. The wave-equation for an anisotropic unbounded medium is

$$\rho \frac{\partial^2 u_i}{\partial t^2} = C_{ijkl} \frac{\partial^2 u_l}{\partial x_j \partial x_k}, \tag{1}$$

where the indices $i, j, k, l \in \{1, 2, 3\}$, x and t are the spatial and temporal variables and Einstein summation notation is used. The variable u_i is the displacement in direction i. The elastic properties of each material are described by the density ρ and the stiffness-tensor C_{ijkl}. The stiffness tensor has 81 components which can be reduced to 21 independent coefficients to describe a fully-anisotropic medium [5]. Here, the analysis is restricted to materials that are either isotropic or transverse-isotropic, which reduces the number of independent coefficients further. As described previously, the wave-vectors of the partial-waves lie in a plane of material symmetry, (x_1, x_3). Here $\partial/\partial x_2 = 0$ and the expanded form of the wave-equation Eq. (1) is

$$\begin{aligned}
\rho \frac{\partial^2 u_1}{\partial t^2} &= C_{11} \frac{\partial^2 u_1}{\partial x_1^2} + C_{55} \frac{\partial^2 u_1}{\partial x_3^2} + (C_{13} + C_{55}) \left(\frac{\partial^2 u_3}{\partial x_1 \partial x_3} \right) \\
\rho \frac{\partial^2 u_2}{\partial t^2} &= C_{66} \frac{\partial^2 u_2}{\partial x_1^2} + C_{44} \frac{\partial^2 u_2}{\partial x_3^2} \\
\rho \frac{\partial^2 u_3}{\partial t^2} &= C_{55} \frac{\partial^2 u_3}{\partial x_1^2} + C_{33} \frac{\partial^2 u_3}{\partial x_3^2} + (C_{13} + C_{55}) \left(\frac{\partial^2 u_1}{\partial x_1 \partial x_2} \right),
\end{aligned} \tag{2}$$

where Voigt notation has been used to contract the indices of the stiffness-matrix (where $11 \to 1, 22 \to 2, 33 \to 3, 23 \to 4, 13 \to 5, 12 \to 6$). A single-frequency plane wave can be written in the form

$$u_i = A_i \exp(\mathrm{i}(\zeta x_1 + \zeta \alpha x_3 - \omega t)), \tag{3}$$

where $i \in \{1, 2, 3\}$, ω is the circular frequency, α is the ratio of the vertical and horizontal (ζ) wavenumbers, and A_i is the polarisation unit vector which describes the direction of displacement relative to the direction of wave propagation. Substituting Eq. (3) into Eq. (2) gives the Christoffel equation

$$\Gamma_{ij}(\alpha) A_j = 0, \tag{4}$$

where the components of the Christoffel matrix (Γ) are

$$\begin{aligned}
&\Gamma_{11} = (C_{11} - \rho\nu^2 + C_{55}\alpha^2) && \Gamma_{22} = (C_{66} - \rho\nu^2 + C_{44}\alpha^2) \\
&\Gamma_{33} = (C_{55} - \rho\nu^2 + C_{33}\alpha^2) && \Gamma_{13} = \Gamma_{31} = (C_{13} + C_{55})\alpha \\
&\Gamma_{12} = \Gamma_{21} = \Gamma_{32} = \Gamma_{23} = 0.
\end{aligned}$$

The phase velocity ν along the x_1 axis is calculated from the relation $\nu = \omega/\zeta$. Solving Eq. (4) admits three solutions for α^2 and therefore 6 solutions for α. From here, the notation α_q, where $q \in \{1, 2, ..., 6\}$, will be used to indicate each solution.

It can be seen from Eq. (4) that the plane wave component A_2 is only dependent on Γ_{22}, hence displacement occurring in the (x_1, x_3) plane is independent of displacement in x_2. Four solutions $(q = 1, 2, 3, 4)$ of α_q Eq. (4) can be found when

$$\det \begin{vmatrix} \Gamma_{11}(\alpha) & \Gamma_{13}(\alpha) \\ \Gamma_{13}(\alpha) & \Gamma_{33}(\alpha) \end{vmatrix} = 0. \tag{5}$$

These describe upwards and downward travelling quasi-shear-vertical (qSV) and quasi-longitudinal (qL) waves with the displacement restricted to the (x_1, x_3) plane. The remaining two solutions $(q = 5, 6)$ are found from $\Gamma_{22} = 0$, and correspond to upward- and downward- travelling quasi-shear-horizontal (qSV) waves. The notation A_{iq} will be used to indicate polarisation vector for each solution q. The displacement field can now be written as

$$u_i = \sum_q A_{iq} B_q \exp(\mathrm{i}(\zeta x_1 + \zeta \alpha_q x_3 - \omega t)), \tag{6}$$

where B_q is the amplitude of each partial-wave. Additionally, the stress field within the unbounded medium can be found using Hooke's law

$$\sigma_{ij} = C_{ijkl} \left(\frac{\partial u_k}{\partial x_l} + \frac{\partial u_l}{\partial x_k} \right), \tag{7}$$

where $i, j, k, l \in \{1, 2, 3\}$. For a multi-layered medium, the Christoffel equation Eq. (4) is solved independently for every layer to calculate the polarisation vector and wave-vector of each partial-wave. However, the amplitude B_q of each partial-wave is solved by invoking the boundary conditions at the interfaces of adjacent layers. This is discussed in the following section.

2.3 Boundary Conditions and Partial-wave Amplitudes

As mentioned previously, the wave-vector of the plane waves are constrained to a plane of symmetry of the transverse-isotropic material reducing the analysis to two dimensions, (x_1, x_3). The normal and transverse displacement and stress describing (quasi-)longitudinal and (quasi-)shear-vertical waves for a single layer is written in the form

$$\begin{bmatrix} u_1 \\ u_3 \\ \sigma_{33} \\ \sigma_{13} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ D_{11} & D_{12} & D_{13} & D_{14} \\ D_{21} & D_{22} & D_{23} & D_{24} \end{bmatrix} \begin{bmatrix} e_1 & & & \\ & e_2 & & \\ & & e_3 & \\ & & & e_4 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix}, \tag{8}$$

where

$$D_{1q} = (C_{13}A_{1q} + C_{33}\alpha_q A_{3q})(\mathrm{i}\zeta) \qquad D_{2q} = C_{55}(\alpha_q A_{1q} + A_{3q})(\mathrm{i}\zeta),$$
$$e_q = \exp(\mathrm{i}(\zeta x_1 + \zeta\alpha_q x_3 - \omega t)).$$

Only the first four solutions of α_q are needed as the motion is restricted to two dimensions (x_1, x_3). The left hand vector of Eqs. (8) contains the components of the displacement and stress, and the right hand vector contains the amplitude of the partial-wave components. The product of the matrices in Eq. (8) will be written as a field matrix $\mathbf{F}$. At an interface $x_3 = d$ between material layers in welded contact, the normal transverse stress and displacement must be continuous across the interface. Therefore, the product of the field matrix and wave amplitudes at the interface of one layer is set equal to the field matrix and wave amplitudes of the adjacent layer. This process is repeated for every interface of the layered medium. For n-layers, there are $4(n-1)$ boundary conditions and $4n$ wave amplitudes which can be arranged into a global matrix. For example, for a medium consisting of 4 layers, the global matrix equation is written

$$\begin{bmatrix} \mathbf{F}_1^1 & -\mathbf{F}_2^1 & & \\ & \mathbf{F}_2^2 & -\mathbf{F}_3^2 & \\ & & \mathbf{F}_3^3 & -\mathbf{F}_4^3 \end{bmatrix} \begin{bmatrix} \mathbf{B_1} \\ \mathbf{B_2} \\ \mathbf{B_3} \\ \mathbf{B_4} \end{bmatrix} = 0. \tag{9}$$

Here, $\mathbf{F}_n^N$ is the 4×4 field matrix and $\mathbf{B}_n$ a 4×1 vector of partial-wave amplitudes of layer n at interface N. By assigning values to four of the partial-wave amplitudes, Eq. (9), can be rearranged and solved for the remaining partial-wave amplitudes. For example, if a compressional wave in the first medium is incident on the layered-structure, the downward-travelling partial-wave amplitude relating to shear (B_2^1, Figure 1) in the first layer and upward-travelling partial-wave amplitudes relating to compressional (B_3^n) and shear (B_4^n) waves in the last (nth) layer are set to zero. Finally, the downward-travelling partial-wave amplitude relating to a compressional wave in the first layer is set to an arbitrary value ($B_1^1 = 1$). In this case, the solved amplitudes describe the solution for an incident single-frequency plane-wave at an angle θ or wavenumber ζ and frequency f. For a four-layered medium this would be

$$\begin{bmatrix} \mathbf{F}_1^{1+} & -\mathbf{F}_2^1 & & \\ & \mathbf{F}_2^2 & -\mathbf{F}_3^2 & \\ & & \mathbf{F}_3^3 & -\mathbf{F}_4^{3-} \end{bmatrix} \begin{bmatrix} \mathbf{B}_1^+ \\ \mathbf{B_2} \\ \mathbf{B_3} \\ \mathbf{B}_4^- \end{bmatrix} = \begin{bmatrix} -\mathbf{F}_1^{1-} & \\ & \\ & \mathbf{F}_4^{3+} \end{bmatrix} \begin{bmatrix} \mathbf{B}_1^- \\ \\ \mathbf{B}_4^+ \end{bmatrix}. \tag{10}$$

Here, $+$ and $-$ superscripts indicate the upwards and downwards travelling partial-wave amplitudes and their respective columns in the field matrices. For example, $\mathbf{F}_1^{1+}$ would be the third and fourth columns of $\mathbf{F}_1^1$, and $\mathbf{B}_1^+$ would be the third and fourth elements of $\mathbf{B_1}$. In the example described above, the first element of $\mathbf{B}_1^-$ is 1 and all the elements of $\mathbf{B}_4^+$ and the second element of $\mathbf{B}_1^-$ are 0. Additionally, once the remaining wave-amplitudes for each layer are found, Eq. (8) can be used to find the displacement and stress anywhere in the layered structure. Alternatively, the dispersion curves can be extracted from the model by setting the incident wave-amplitudes of the layered structure to zero and finding the frequency-wavenumber pairs in which the resulting left-hand-side matrix of Eq. (10) becomes singular. The algorithm used is described further in Section 2.5

2.4 Shear-Horizontal Waves

Shear-horizontal waves propagate independently of (quasi-)shear-vertical and (quasi-)compressional waves and their propagation is analogous to compressional waves in a liquid [25]. Solutions 5 and 6 of α_q correspond to shear-horizontal waves and the displacement and shear stress can be written in the form

$$\begin{bmatrix} u_2 \\ \sigma_{23} \end{bmatrix} = \begin{bmatrix} A_{25} & A_{26} \\ D_{35} & D_{36} \end{bmatrix} \begin{bmatrix} e_5 & \\ & e_6 \end{bmatrix} \begin{bmatrix} B_5 \\ B_6 \end{bmatrix}, \tag{11}$$

where

$$D_{3q} = C_{44}\alpha_q A_{2q}(i\zeta).$$

By setting the properties of a new medium C' so that $C'_{11}, C'_{22}, C'_{33} = C_{44}$ and setting the remaining coefficients to zero, Eq. (8) reduces to Eq. (11). Hence, the propogation of shear-horizontal waves can be modelled with Eq. (8) by reassigning the relevant components of the stiffness matrix.

2.5 Implementation Details

To construct the global matrix, Eq. (9), a field matrix $\mathbf{F_i}$ must be calculated for each layer. To improve the conditioning of the matrix, rows relating to displacement equations are scaled by ζ the horizontal wavenumber, and rows relating to stress are scaled by $\rho\omega^2$. The system matrix is constructed by looping over each interface, calculating the 4×4 field matrices above and below each interface and arranging them into a single matrix. This leads to a rectangular matrix which has $4n$ columns and $4n - 1$ rows. Additionally there are $4n$ partial-wave amplitudes. Four partial-wave amplitudes are defined and the global-matrix in Eq. (9) is rearranged to be square Eq. (10). The resulting equation is solved using the `mldivide` function in MATLAB. This function solves a system of linear equations using the fastest algorithm based on the matrix structure. However, the global-matrix becomes singular at values of ζ and ω on or close-to dispersion curve solutions.

The computation of dispersion curves follows the algorithm described in [4]. Firstly, the wavenumber parameter is fixed and the determinant of the global-matrix is found over a range of frequencies. Close to dispersion solutions the determinant of the global-matrix tends to zero. Using these as starting points and taking a limit either side, the exact frequency and wavenumber of the dispersive solution is found using a bisection algorithm. `ElasticMatrix` makes use of MATLAB's `fmincon()` function for this. These solutions are the starting points for each dispersion curve. To find the second point on each dispersion curve, the fixed value of wavenumber is increased and the search is performed again. The algorithm then uses linear interpolation to estimate the location of the third, fourth and fifth points on the dispersion curve, similarly using a bisection algorithm to find the exact frequency-wavenumber pairs. After five points have been found, a higher-order polynomial interpolation scheme is used to more accurately predict points on the dispersion curve. The algorithm implemented in `ElasticMatrix` only searches in the real domain of ζ which is a good-estimate for simple plate structures in a vacuum, however, it may be inaccurate for leaky solutions, for example an plate embedded in soil.

Slowness profiles are calculated by defining a range of phase-speeds and calculating the horizontal and vertical component of wavenumber by solving the Christoffel equation Eq. (5). The values from the calculation may be complex, however, only the real values are plotted.

3 Software Description and Examples

3.1 Overview

The `ElasticMatrix` toolbox implements the partial-wave method using an object-oriented framework in MATLAB. This allows the toolbox to be used with either a simple scripting or command line interface, and makes it easy to use and expand. The software is divided into three classes. The first class, `Medium`, defines the multi-layered geometry and material properties of each layer. The second class `ElasticMatrix` is initialised by a `Medium` object. This class contains the partial-wave method implementation and methods for extracting additional details such as dispersion curves and reflection coefficients. By default, all the calculations use 64 bit precision. The final class, `FabryPerotSensor`, is an example of how numerical models can be built from the `ElasticMatrix` and `Medium` objects. This class inherits `ElasticMatrix` and can be used to model the directional response of a Fabry-Pérot ultrasound sensor. More details can be found in [8, 18]. Each class in the toolbox inherits the MATLAB handle class. Consequently, the object does not need to be reassigned when a method is called. The classes and their respective attributes and methods can be seen in Fig. 2. The toolbox is self contained and has been tested with MATLAB 2016a and above.

This section presents a small selection of code snippets and examples. More detailed examples can be found in the `ElasticMatrix ./examples` folder and `html` documentation can be accessed through the MATLAB help and clicking `ElasticMatrix toolbox`. There are three steps to using the toolbox. Firstly, the geometry of the layered medium must be defined. Secondly, the input parameters to the model should be defined, which are generally a range of angles, frequencies or wavenumbers. Finally, the model can be solved and details such as the reflection coefficients and dispersion curves can be extracted. Note, for clarity in the code implementation, the x_1 and x_3 coordinates are referred to as x and z, respectively.

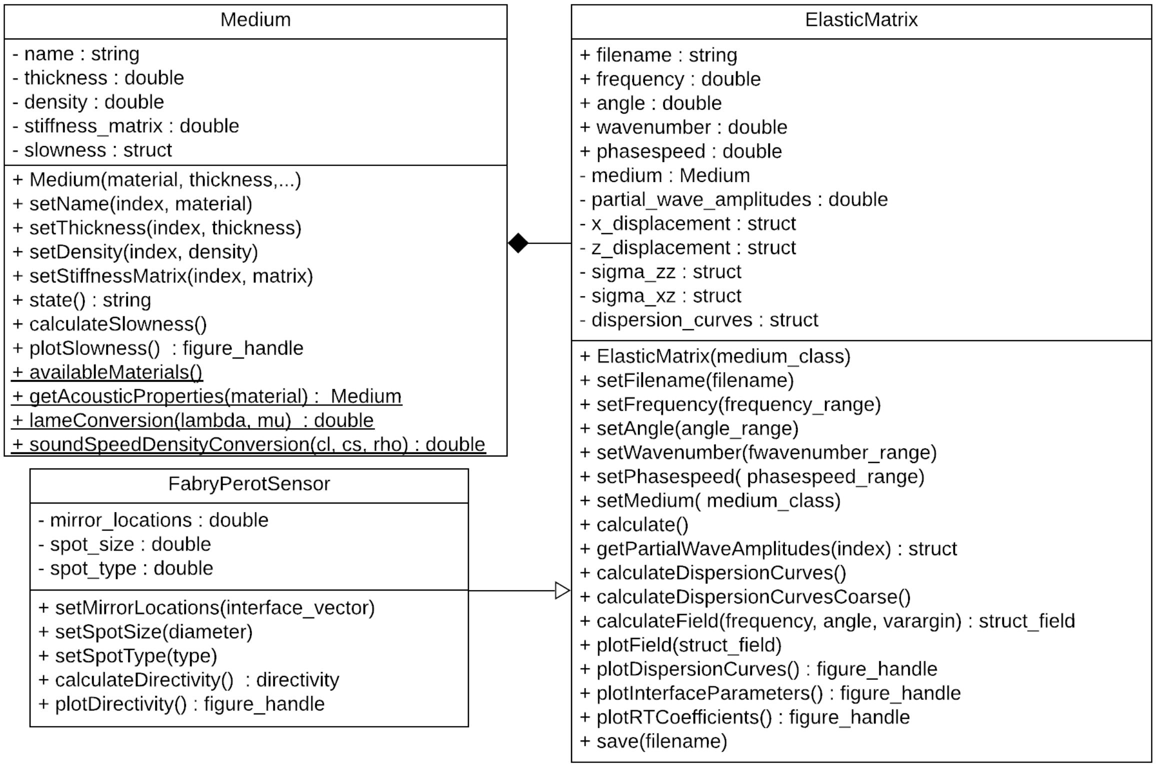
Medium
- name : string
- thickness : double
- density : double
- stiffness_matrix : double
- slowness : struct
+ Medium(material, thickness,...)
+ setName(index, material)
+ setThickness(index, thickness)
+ setDensity(index, density)
+ setStiffnessMatrix(index, matrix)
+ state() : string
+ calculateSlowness()
+ plotSlowness() : figure_handle
+ availableMaterials()
+ getAcousticProperties(material) : Medium
+ lameConversion(lambda, mu) : double
+ soundSpeedDensityConversion(cl, cs, rho) : double
ElasticMatrix
+ filename : string
+ frequency : double
+ angle : double
+ wavenumber : double
+ phasespeed : double
- medium : Medium
- partial_wave_amplitudes : double
- x_displacement : struct
- z_displacement : struct
- sigma_zz : struct
- sigma_xz : struct
- dispersion_curves : struct
+ ElasticMatrix(medium_class)
+ setFilename(filename)
+ setFrequency(frequency_range)
+ setAngle(angle_range)
+ setWavenumber(fwavenumber_range)
+ setPhasespeed(phasespeed_range)
+ setMedium(medium_class)
+ calculate()
+ getPartialWaveAmplitudes(index) : struct
+ calculateDispersionCurves()
+ calculateDispersionCurvesCoarse()
+ calculateField(frequency, angle, varargin) : struct_field
+ plotField(struct_field)
+ plotDispersionCurves() : figure_handle
+ plotInterfaceParameters() : figure_handle
+ plotRTCoefficients() : figure_handle
+ save(filename)
FabryPerotSensor
- mirror_locations : double
- spot_size : double
- spot_type : double
+ setMirrorLocations(interface_vector)
+ setSpotSize(diameter)
+ setSpotType(type)
+ calculateDirectivity() : directivity
+ plotDirectivity() : figure_handle

Figure 2: UML class diagram for `Medium`, `ElasticMatrix` and `FabryPerotSensor`. The top field for each box indicates the name of the class, the second field lists the properties, and the third field lists the methods. Here, `ElasticMatrix` is composed from `Medium` and `FabryPerotSensor` inherits `ElasticMatrix`. The $(-)$ indicates a private method or property and $(+)$ indicates a public method or property. Underlined methods are static.

3.2 `Medium`

The `Medium` class is used to define the material properties and thickness of each layer. The class is initialised by calling the class constructor with input arguments of the material name followed by its thickness. However, the thickness of the first and last layers are semi-infinite and their values should be set with the `Inf` keyword. The `Medium` class will automatically set the thickness of the first and last layer to `Inf` if another value is used. An example is given below.

```
my_medium = Medium('water', Inf, 'blank', 3e-3, 'PVDF', 1e-3, 'glass', Inf);
```

Here, `my_medium` is an object array and every index in the object array corresponds to a different layer in the medium. In the current example, `my_medium(3)` will return a object with the material properties and thickness associated with PVDF. The `'blank'` keyword can used for a material which is not predefined. The material properties and names can be set using their respective `set` functions. User defined materials can be added to the script `materialList.m`.

3.3 Slowness profiles

Slowness profiles are a plot of the inverse-phase velocity of each bulk wave component. They can be used to determine the angles of reflection and transmission between multi-layered media as well as the direction of energy propagation and skew angle [22]. Slowness profiles are found by solving the Christoffel equation, Eq. 4, and only depend on the material properties of each material. The method `.calculateSlowness` is part of the `Medium` class, and calls the function

```
 calculateAlphaCoefficients(...)
```

which is an implementation of Eqs. (4) and (5). This takes input arguments of the material properties and phase-velocity and returns the polarisation and wave-vectors. The slowness profiles given by this function are plotted in terms of k_x/ω vs k_z/ω. For an isotropic material, the slowness profiles for each bulk wave are spherical, however, this is not true for an anisotropic material. An example of the slowness profiles for isotropic-glass and transverse-isotropic beryl is shown in Fig. 3. This figure has been reproduced from [26]. The slowness profiles of the (quasi-)longitudinal, (quasi-)shear-vertical and (quasi-)shear-horizontal bulk waves are shown. As glass is an isotropic material, the slowness profiles are spherical and the magnitudes of L,SV and SH when $k_x/\omega = 0$ or $k_z/\omega = 0$ are equal to the reciprocal of the compressional- and shear-speeds of glass. For the transverse-isotropic case, when $k_x/\omega = 0$, the value of qL is equal to $\sqrt{\rho/C_{33}}$ and qSV is equal to $\sqrt{\rho/C_{55}}$. When $k_z/\omega = 0$, the value of qL is equal to $\sqrt{\rho/C_{11}}$ and the value of qSV is equal to $\sqrt{\rho/C_{55}}$. These have been checked in the toolbox example script and all are within numerical precision.

```
my_medium = Medium('glass', Inf, 'beryl', Inf);
my_medium.calculateSlowness;
my_medium.plotSlowness;
```

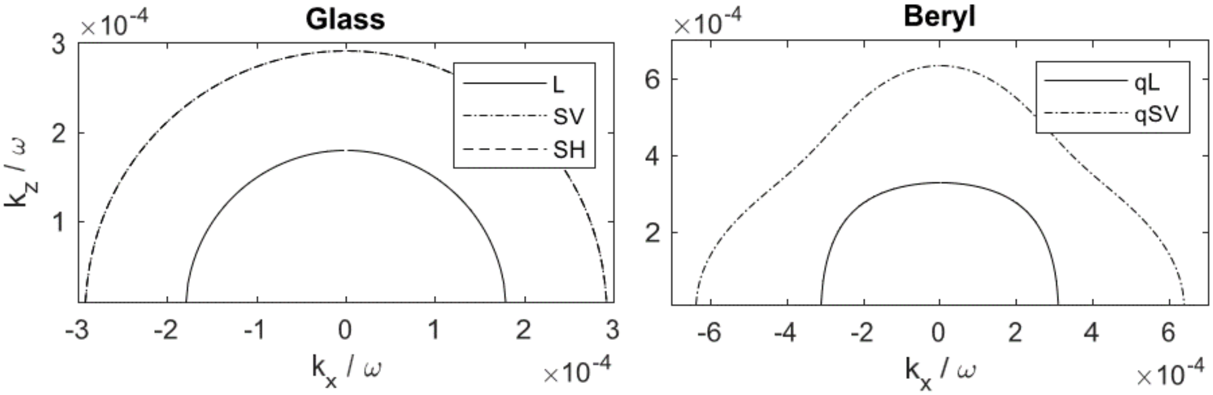

Figure 3: The slowness curves for isotropic-glass and transverse-isotropic beryl materials where (q)L, (q)SV, (q)SH correspond to the reciprocal of the (quasi-)longitudinal, (quasi-)shear-vertical and (quasi-)shear-horizontal partial-wave speeds.

3.4 `ElasticMatrix`

The medium class is used to initialise the `ElasticMatrix` class which runs the partial-wave method over a range of frequencies, wavenumbers, phasespeeds and angles. Two of these must be defined using the `.set` functions. The `.calculate` method is then used to run the partial-wave procedure. If the properties are not set before the `.calculate` method is called, the model will run over a predefined range of frequencies and angles. The `.calculate` method constructs, rearranges and solves the global-matrix, Eq. (10), using the function

```
 calculateMatrixMethod(...)
```

This function takes input arguments of the material properties and the parameters to calculate over (angles, frequencies, wavenumbers). It returns the determinant of the system matrix and the stresses and displacements at the layer interfaces. Each individual field-matrix is calculated using the function

```
 calculateFieldMatrixAnisotropic(...)
```

which is an implementation of Eq. (8). This takes input arguments of the material properties, the wave-vector components, polarisation components and the phase velocity and returns the field-matrix. The default calculation is to find the partial-wave amplitudes and interface stresses and displacements when there is a single-frequency compressional wave incident on the structure from the first layer. An example is given below for a titanium plate.

```
my_medium = Medium('water', Inf, 'titanium', 1e-3, 'water', Inf);
my_model = ElasticMatrix(my_medium); % initialise class
my_model.setFrequency(linspace(1e6, 5e6, 100));
my_model.setAngle(linspace(0, 45, 100));
my_model.calculate;
```

3.5 Reflection and Transmission Coefficients

For a plane wave incident at an oblique angle on a multi-layered structure, the reflection and transmission coefficients determine the amplitude of the wave that is reflected and transmitted at each interface. Knowing these coefficients is useful for a number of applications. For example, selecting the appropriate launch angle when coupling energy into particular modes in a wave-guide, or determining the thickness and material properties of matching layers for ultrasonic transducers [22, 27].

The angles of refraction at the interfaces between multi-layered media can be found by studying the slowness profiles. However, slowness profiles do not take into account the boundary conditions at the interfaces. Consequently, the magnitude of each of the refracted waves cannot be calculated directly. For a plane wave incident on a multi-layered structure, the magnitude of the reflection and transmission coefficients are found by normalising the partial-wave amplitudes B_i^n by the incident plane wave amplitude B_1^1. This is automatically calculated when using the `.calculate` method.

An example of the reflection and transmission coefficients at a PVDF-aluminium interface is given below and shown in Fig. 4. For a plane compressional wave incident on a PVDF-aluminium interface, there are four resulting refracted waves. These are a reflected R and transmitted T compressional L and shear S wave. The reflection and transmission coefficients have been compared to the analytic solutions for a two-layered elastic-medium from [22] and have an average error less than $1e^{-15}$ which is within numerical precision for a 64 bit floating point number. For clarity the analytical solutions have not been plotted but can be seen in the toolbox examples folder.

```
my_medium = Medium(‘PVDF’, Inf, ‘aluminium’, Inf);
my_model = ElasticMatrix(my_medium);
my_model.setFrequency(1e6);
my_model.setAngle(linspace(0, 90, 90));
my_model.calculate;
my_model.plotRTCoefficients;
```

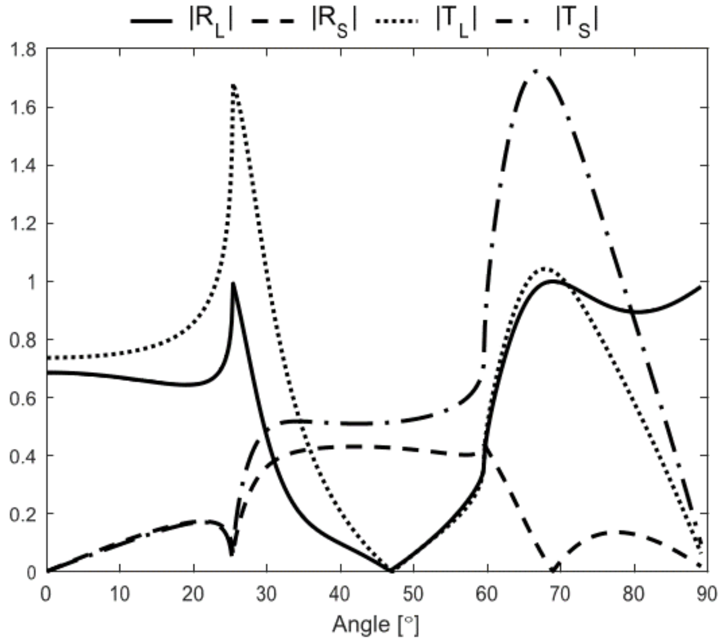

Figure 4: Longitudinal L and shear S reflection R and transmission T coefficients for a PVDF-Aluminium interface.

3.6 Dispersion Curves

Dispersion curves describe the modal solutions of the multilayer structure and describe a wave-mode which propagates parallel to the layer-interfaces independently of a bulk wave. As one example, exciting these modes is essential in ultrasonic inspection. Knowledge of the dispersion curves is useful for determining the most appropriate modes to excite and for optimising the inspection process.

As mentioned in Section 2.5, the modal solutions are found when the global matrix becomes singular. The `ElasticMatrix` software can calculate dispersion curves for simple layered structures (i.e., a plate in a vacuum or water). However, it is not robust for very-leaky cases, for example a plate embedded in soil. For these types of cases either Disperse, or other techniques based on the spectral-collocation method or semi-analytic finite element method are more appropriate [12, 13, 14, 15, 16, 28]. An example of the dispersion curves for a 1 mm titanium plate in a vacuum is shown in Fig. 5(a). The dispersion curves are plotted on a graph of frequency vs wavenumber and show the first three symmetric S and anti-symmetric A Lamb modes. The results from Disperse are also plotted and have excellent agreement.

```
my_medium = Medium('vacuum', Inf, 'titanium', 0.001, 'vacuum', Inf);
my_model = ElasticMatrix(my_medium);
my_model.setFrequency(linspace(0.5e6, 5e6, 100));
my_model.calculateDispersionCurves;
my_model.plotDispersionCurves;
```

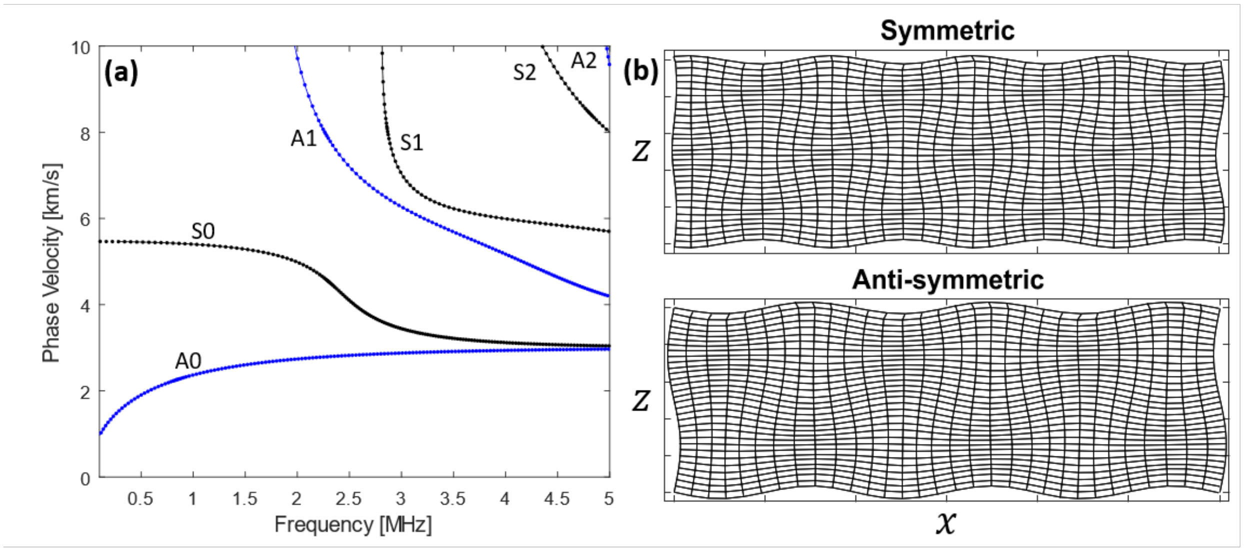

Figure 5: (a) Dispersion curves for a titanium plate in a vacuum. The solid lines are from `ElasticMatrix` and the points are generated using Disperse [12]. The first three symmetric (S, black) and anti-symmetric (A, blue) have been plotted. (b) Displacement field for an anti-symmetric and symmetric mode shape.

3.7 Displacement and Stress Fields

More information about the wave-physics and guided wave structures can be taken from dispersion curves by plotting the displacement and stress fields at different points. In the `ElasticMatrix` software implementation, the x and z ranges over which to plot the displacement or stress fields must be specified. The `.calculateField(...)` method returns a structure with the input ranges and field values at each point of the resulting grid. The values of the structure can be plotted independently or given as an argument to the `.plotField` method. An example is given below for the displacement field within an titanium plate for a symmetric and anti-symmetric mode. The resulting plot can be seen in Fig. 5(b).

```
field_values = myModel.calculateField(freq, angle, {x_range, z_range});
myModel.plotField(field_values, plot_style);
```

3.8 `FabryPerotSensor`

One application of the toolbox is to model the directional response of Fabry-Pérot ultrasound sensors [8]. The `FabryPerotSensor` is a child class of `ElasticMatrix` and is an example of how the `ElasticMatrix` toolbox may be expanded. The are additional inputs to this class which are described in more detail in the `./examples` folder, and a description of the modelling process can be found in [8, 18]. The modelled directional response was found to have good agreement with the measured directional response. An example of the modelled and measured directional response for a glass etalon Fabry-Pérot sensor can be seen in Fig. 6. Features of the directional response correspond to symmetric and anti-symmetric Lamb modes propagating within the sensor.

```
my_medium = Medium(‘water’, Inf, ‘AlMir’, 1e-8, ‘glass’, 175e-6, ‘AlMir’, ...
    1e-8, ‘air’, Inf);
fp_sensor = FabryPerotSensor(my_medium);
fp_sensor.setAngle(linspace(0, 45, 45));
fp_sensor.setFrequency(linspace(0.1e6, 100e6, 100));
fp_sensor.setMirrorLocations([1, 4]);
fp_sensor.calculateDirectivity;
fp_sensor.plotDirectivity;
fp_sensor.calculateDispersionCurves;
```

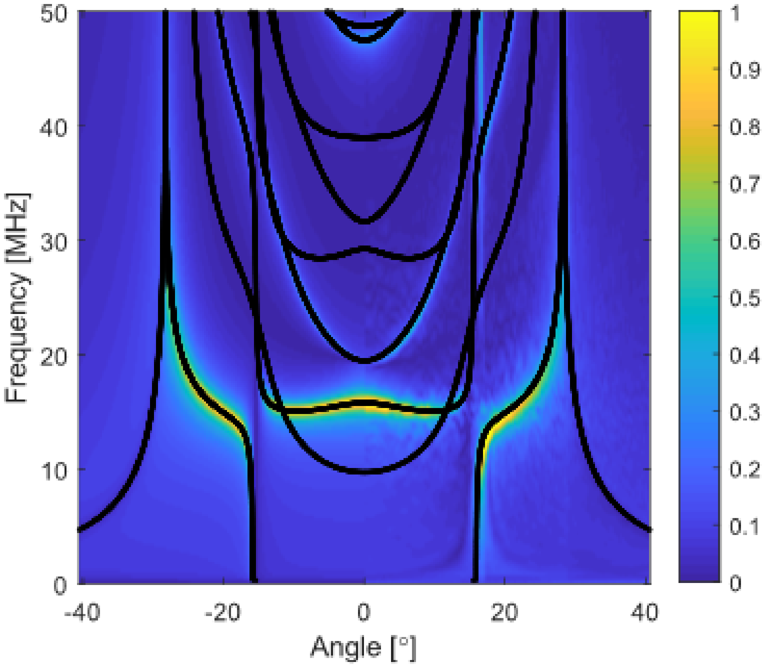

Figure 6: The modelled directional response ($-40^\circ - 0^\circ$) and measured directional response ($0^\circ - 40^\circ$) from [8]. The dispersion curves associated with this sensor are plotted as black points.

4 Impact and Conclusions

This paper introduces a new open-source toolbox called `ElasticMatrix` which models elastic wave propagation in multi-layered media with anisotropic materials with isotropic or transverse-isotropic symmetry. The toolbox uses the partial-wave method which allows the calculation of slowness profiles, reflection and transmission coefficients, dispersion curves and stress and displacement fields. The software has been implemented using the object-oriented capabilities of MATLAB allowing for a simple command line or scripting interface. The implementation allows researchers to add functionality and integrate the software into other projects. For example, this toolbox has already been used to model the directional response of Fabry-Pérot ultrasound sensors [8]. It is anticipated the research user-base will actively contribute to `ElasticMatrix` and add to the functionality.

Acknowledgements

This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC), UK, Grant Nos. EP/P008860/1, EP/L020262/1 and EP/S026371/1, in part by the EPSRC-funded UCL Centre for Doctoral Training in Medical Imaging (EP/L016478/1) and the Department of Health's NIHR-funded Biomedical Research Centre at NIHR Biomedical Research Centre at University College London Hospitals.

References

[1] C. Willberg, S. Duczek, J. Vivar-Perez, Z. Ahmad, Simulation methods for guided wave-based structural health monitoring: a review, Appl. Mech. Rev. 67 (1) (2015) 010803 (2015).

[2] H. Song, J. Popovics, Characterization of steel-concrete interface bonding conditions using attenuation characteristics of guided waves, Cement. Concrete. Comp. 83 (2017) 111–124 (2017).

[3] M. Rucka, E. Wojtczak, J. Lachowicz, Detection of debonding in adhesive joints using lamb wave propagation, in: MATEC. Web. Conf., Vol. 262, EDP Sciences, 2019, p. 10012 (2019).

[4] M. Lowe, Matrix techniques for modeling ultrasonic waves in multilayered media, IEEE Trans. Ultrason. Ferroelect. Freq. Contr. 42 (4) (1995) 525–542 (Jul 1995).

[5] A. H. Nayfeh, Wave propagation in layered anisotropic media: With application to composites, Vol. 39, Elsevier, 1995 (1995).

[6] C. Hakoda, C. J. Lissenden, Using the partial wave method for mode-sorting of elastodynamic guided waves, in: AIP Conf. Proc., Vol. 2102, AIP Publishing, 2019, p. 020014 (2019).

[7] C. Hakoda, C. Lissenden, Using the partial wave method for wave structure calculation and the conceptual interpretation of elastodynamic guided waves, Appl. Sci. 8 (6) (2018) 966 (2018).

[8] D. Ramasawmy, E. Martin, J. Guggenheim, E. Zhang, P. Beard, B. Treeby, B. Cox, Analysis of the directivity of glass etalon Fabry-Pérot ultrasound sensors, IEEE Trans. Ultrason. Ferroelect. Freq. Contr. (2019).

[9] J. Chen, J. Guo, E. Pan, Reflection and transmission of plane wave in multilayered nonlocal magneto-electro-elastic plates immersed in liquid, Compos. Struct. 162 (2017) 401–410 (2017).

[10] G. Li, Y. Zheng, Y. Jiang, Z. Zhang, Y. Cao, Guided wave elastography of layered soft tissues, Acta. Biomater. 84 (2019) 293–304 (2019).

[11] S. Shoja, V. Berbyuk, A. Boström, Investigating the application of guided wave propagation for ice detection on composite materials, in: In Proc. Int. Conf. Eng. Vib., 2015, pp. 152–161 (2015).

[12] B. Pavlakovic, M. Lowe, D. Alleyne, P. Cawley, Disperse: a general purpose program for creating dispersion curves, in: Review of progress in quantitative nondestructive evaluation, Springer, 1997, pp. 185–192 (1997).

[13] F. H. Quintanilla, M. Lowe, R. V. Craster, Full 3d dispersion curve solutions for guided waves in generally anisotropic media, J. Sound. Vib. 363 (2016) 545–559 (2016).

[14] F. H. Quintanilla, Z. Fan, M. Lowe, R. V. Craster, Guided waves' dispersion curves in anisotropic viscoelastic single-and multi-layered media, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 471 (2183) (2015) 20150268 (2015).

[15] F. H. Quintanilla, M. Lowe, R. V. Craster, Modeling guided elastic waves in generally anisotropic media using a spectral collocation method, J. Acoust. Soc. Am. 137 (3) (2015) 1180–1194 (2015).

[16] A. T. Adamou, R. V. Craster, Spectral methods for modelling guided waves in elastic media, J. Acoust. Soc. Am. 116 (3) (2004) 1524–1535 (2004).

[17] J. L. Prego Borges, Lamb: a simulation tool for air-coupled lamb wave based ultrasonic nde systems (2010).

[18] B. T. Cox, P. C. Beard, The frequency-dependent directivity of a planar Fabry-Pérot polymer film ultrasound sensor, IEEE Trans. Ultrason. Ferroelect. Freq. Contr. 54 (2) (2007).

[19] W. T. Thomson, Transmission of elastic waves through a stratified solid medium, J. Appl. Phys. 21 (2) (1950) 89–93 (1950).

[20] A. H. Nayfeh, The general problem of elastic wave propagation in multilayered anisotropic media, J. Acoust. Soc. Am. 89 (4) (1991) 1521–1531 (1991).

[21] L. P. Solie, B. A. Auld, Elastic waves in free anisotropic plates, J. Acoust. Soc. Am. 54 (1) (1973) 50–65 (1973).

[22] J. L. Rose, Ultrasonic Waves in Solid Media, Vol. 107, Cambridge University Press, 2004 (2004).

[23] N. A. Haskell, The dispersion of surface waves on multilayered media, B. Seismol. Soc. Am. 43 (1) (1953) 17–34 (1953).

[24] L. Brekhovskikh, Waves in layered media, Vol. 16, Elsevier, 2012 (2012).

[25] L. Brekhovskikh, O. A. Godin, Acoustics of layered media I: Plane and quasi-plane waves, Vol. 5, Springer Science & Business Media, 2012 (2012).

[26] R. C. Payton, Elastic wave propagation in transversely isotropic media, Vol. 4, Springer Science & Business Media, 2012 (2012).

[27] S. P. Kelly, G. Hayward, T. E. G. Alvarez-Arenas, Characterization and assessment of an integrated matching layer for air-coupled ultrasonic applications, IEEE Trans. Ultrason. Ferroelect. Freq. Contr. 51 (10) (2004) 1314–1323 (2004).

[28] Ş. Sorohan, N. Constantin, M. Găvan, V. Anghel, Extraction of dispersion curves for waves propagating in free complex waveguides by standard finite element codes, Ultrasonics 51 (4) (2011) 503–515 (2011).

Current code version

Nr.	Code metadata description	
C1	Current code version	v1
C2	Permanent link to code/repository used for this code version	github.com/dannyramasawmy/ ElasticMatrix
C3	Legal Code License	GNU Lesser General Public License v3.0
C4	Code versioning system used	git
C5	Software code languages, tools, and services used	MATLAB.
C6	Compilation requirements, operating environments & dependencies	MATLAB 2016a and above
C7	If available Link to developer documentation/manual	github.com/dannyramasawmy/ ElasticMatrix
C8	Support email for questions	dannyramasawmy+elasticmatrix @gmail.com

Table 1: Code metadata