POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

# Physics-Informed Neural Network for damage localization using Lamb waves

Tesi di Laurea Magistrale in
Aeronautical Engineering - Ingegneria Aeronautica

**Jacopo Ghellero, 996039**

**Advisor:**
Prof. Francesco Cadini

**Co-advisors:**
Dr. Luca Lomazzi
Eng. Lucio Pinello

**Academic year:**
2022-2023

**Abstract:** Lamb waves have been widely utilized for assessing structural damage due to their sensitivity to defects. Despite their ease of excitation and acquisition, significant processing is often necessary to derive single-valued indicators, known as damage indices, from the acquired signals. Traditionally, damage indices have been developed using tomographic algorithms to create damage probability maps, though this approach is subject to limitations. Recently, machine learning has been employed to enhance the accuracy of guided wave frameworks for damage diagnosis. However, many existing methods still involve extracting damage indices from the acquired signals, potentially leading to the loss of diagnostic information and decreased accuracy. Recently, a new approach within the machine learning field has risen in popularity for its flexibility and explainability: physics-informed neural networks. These networks allow embedding some known physical laws in the algorithm to make sure the predictions adhere to the physics of the problem. However, little to no applications can be found in the field of damage diagnosis. In this context, the present work aims to present a physics-informed framework to perform damage diagnosis using Lamb waves avoiding damage indices extraction. Various case studies were considered for evaluating the performance of the proposed framework.

**Key-words:** PINN, Machine Learning, Neural Network, SHM

## 1. Introduction

Every component during its lifespan is subjected to different types of loads that inevitably lead to damages and safety concerns. In order to operate engineering structures in safe conditions, different strategies have been developed and tested during years of technological advancement. The first one employed is preventive maintenance which is a time-based approach where equipment or systems are regularly serviced or replaced regardless of their current condition. The primary goal is to prevent breakdowns and extend the lifespan of assets. This strategy involves scheduled inspections, routine replacements of components, and general maintenance activities. Following this approach, unnecessary shutdown and replacement are often performed in order to avoid risks that are not quantifiable due to the limited information on component status available. In recent years thanks to technological development a new approach has raised in popularity that is condition-based maintenance, particularly in the context of Structural Health Monitoring (SHM), which relies on real-time data and sensor information to assess the current health of equipment or structures. Instead of performing

maintenance at predetermined intervals, actions are taken based on the actual condition of the asset. This new way of approaching maintenance allows both cost reduction and increased safety. SHM is becoming very common in safe critical applications, e.g., for dealing with oil and gas pipelines, wind turbines, aeroplanes and many more, where the correct operation of a component is crucial to avoid accidents. Also, SHM has been adopted for cost-critical applications where the cost related to replacing or repairing the component is large enough that minimizing downtimes is crucial to save money.

In this context, among the many SHM methods proposed in the literature, Lamb wave-based algorithms for damage diagnosis have shown satisfactory performance when thin-walled structures are considered [1–6]. Typically, these waves are generated and detected by placing a network of piezoelectric (PZT) devices on the structure [7–13]. The process of diagnosing damage generally involves two stages. Initially, signals are recorded when the structure is undamaged, forming a dataset known as the healthy baseline. Subsequently, current measurements are compared to the baseline to detect any deviations indicative of damage [14, 15]. Many methods following this approach utilize diagnostic algorithms based on tomographic reconstruction to detect and pinpoint damage [15]. A notable example is the probabilistic inspection of damage reconstruction algorithm (RAPID) [16]. However, tomographic algorithms require that features, or damage indices (DIs), be extracted from the diagnostic signals [5, 17–26]. This traditional approach has some limitations: (i) due to uneven sensing network density some artefacts can appear on damage probability maps, (ii) some parameters need to be selected subjectively, thus strongly affecting damage diagnosis performance, and (iii) information on damage quantification are hardly retrievable. In order to address such issues, different machine learning (ML) approaches have been developed recently. Among the first ML-based methods proposed in the literature, the work in Ref. [27, 28] is worth mentioning. Here, feed-forward neural networks (FFNNs) were leveraged to identify and characterise damage. However, those simple networks still need DIs extraction in order to achieve good performance [27, 28]. Damage diagnosis performance using ML-based frameworks has been improved using deep learning algorithms such as convolutional neural networks (CNNs), for their capability of dealing with multi-dimensional signals [29–32]. Although using classification to localise and quantify damage makes the task easier, prediction accuracy is limited by the resolution used to discretise damage position and extent. This limitation has been overcome in [33], where a CNN for regression has been proposed to localise damage in Aluminum plates. Despite the satisfactory performance, the CNN used in this framework does not deal with unprocessed acquired signals, but it requires previous extraction of DIs. Different ML methods have been tested and compared in [34]. CNNs turned out to be the best-performing tools for damage diagnosis in thin-walled structures. However, such networks belong to the supervised learning scheme, which has been identified as costly and unfeasible in some real scenarios. Unsupervised algorithms that can solve the aforementioned issues are the Physics-Informed Neural Networks (PINNs). This class of algorithms aims to solve partial differential equations where the solution field is approximated with a neural network whose weights and biases are learned by minimizing the residual of a differential equation evaluated at collocation points. This general formulation allows for solving a large variety of problems, drastically reducing the required data compared to a traditional supervised surrogate model. Using PINNs as forward solvers, i.e., to solve partial differential equations, is not as efficient as the classical well-established solvers, e.g., the finite element and the finite difference methods. However, in particular, in contexts where classical methods show some limitations, PINNs are still promising [35]. Moreover, PINNs are considered grey boxes, i.e., models that contain both data and theoretical models, thus rendering the model itself more understandable to the user. From the SHM point of view, the solution to the inversion problem, i.e., the reconstruction of material properties starting from signals of Lamb waves measured by the sensor network arranged on the component without the need for any pre-processing or extraction of DIs from the sensors signals. Several attempts to use PINNs to solve inversion problems have been presented in the literature, with satisfactory results. In the field of SHM, the work in [36] proposed to use two nets working together, one that reconstructs the continuous displacement field, and one that predicts the material properties. Despite the satisfactory performance, the optimization problem that needs to be solved for training the networks is complex and excessively time consuming. In order to overcome these limitations, a finite difference solver was embedded in the method, and only a network predicting the material distribution was developed. That is, a network predicted the material distribution, which was taken in by the forward solver to compute the expected displacement. This choice allowed for simplification of the optimization process and increased the method performance [37]. However, the wave type considered was not suitable for analysing thin-walled structures, like panels and other elements of aeronautical devices.

In this context, the goal of the present work is (i) to develop tools and methods in a PINNs framework to perform damage diagnosis using Lamb waves, without extraction of DIs from the signals, and (ii) to assess the possibility of deploying PINNs for SHM applications. Two different approaches have been tested, i.e., a more traditional physics-informed approach and a promising new technique called the Physics-informed method coupled with a finite difference solver. The wave excited to perform damage detection was a symmetric mode Lamb wave using a tone-burst, commonly used in the SHM field. The approaches were later verified using various case studies:

- **Damage localization in a string**: the two methods proposed were tested on a one-dimensional and

non-homogeneous domain with clamped extremities. The wave excited with a tone-burst is applied on the middle of the domain.

- **Damage localization in a plate section**: The physics-informed method coupled with a finite difference solver was further tested on a plate subjected to plane strain in one direction and clamped and both ends. Again, the wave propagated from the middle.
- **Damage localization in a full plate**: The physics-informed method coupled with a finite difference solver was also tested on a 3D plate subjected to transverse loading. The plate was clamped at the extremities and the force was placed in the middle.

The results showed that the proposed method allows characterizing damage using Lamb waves.

This paper is organized as follows. In Section 2 the theoretical background on Lamb waves, inversion problems and PINNs is provided. In the same section, the tools and methods developed are presented showing the general workflow of the paper. The tools were verified in 3 against validated numerical simulations carried out using the Abaqus software package. Finally, in Section 4 the conclusions are drawn out and possible future developments are presented.

# 2. Methodology

## 2.1. Theoretical Background

Lamb waves are ultrasonic-guided waves that propagate through thin-walled structures, such as plates and shells. Unlike bulk waves, which travel through a material in all directions, Lamb waves are confined to the thickness of the structure, allowing them to interact with defects, boundaries, and other features within the material. This property makes them highly valuable for detecting hidden flaws or assessing the structural integrity of a wide range of engineering components, from aircraft wings to pipelines and bridges. Lamb waves can be further categorized into different modes, such as the symmetric mode (S0) and the anti-symmetric mode (A0) as shown in Figure 1 and Figure 2 respectively, each with its own unique characteristics and applications.

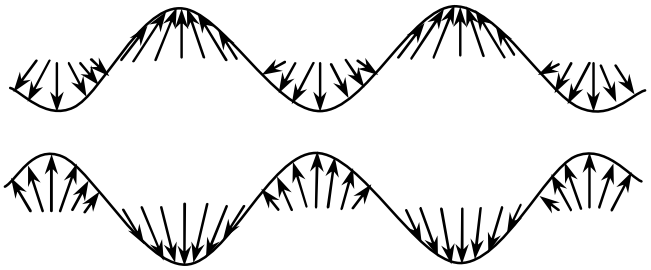

Figure 1: Symmetric mode particle movement (S0)

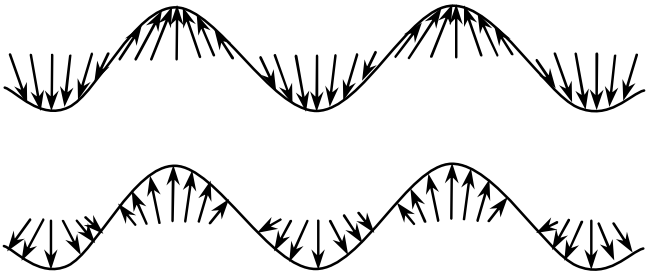

Figure 2: Anti-symmetric mode particle movement (A0)

The equation governing elastic waves, and in particular Lamb waves, propagation is represented by the equation (1):

$$\rho \boldsymbol{u}_{tt} = \boldsymbol{\nabla} \cdot (\boldsymbol{\sigma}) + \boldsymbol{f} \tag{1}$$

where material properties of the domain are considered trough elastic modulus $E$ and Poisson ratio $\nu$ inside the constitutive relationship for elastic material hence relating stress tensor $\boldsymbol{\sigma}$ and displacement vector $\boldsymbol{u}$. The density is represented by $\rho$, forcing vector by $\boldsymbol{f}$ and $\boldsymbol{\nabla}\cdot$ is the divergence operator applied to stress tensor $\boldsymbol{\sigma}$.

From the general equation (1) can be derived three different models: one-dimensional, two-dimensional and three-dimensional. The equations presented assume an homogeneous material, making it impractical to accurately describe domains where material properties vary. The introduction of variable material properties becomes essential for representing real scenarios such as damages and holes within the domain. To non-destructively estimate material properties, addressing the full wave inversion problem is crucial. The solution involves introducing a parameterization $\gamma$ for material properties, which is then multiplied by the material properties to determine the material distribution within the domain.

First in order of complexity is the 1D wave, where the wave equation in its simplest form is solved, an axial wave that propagates along the coordinate $x$ through a one-dimensional string with the two ends fully constrained. Parametrization used to solve full inversion problem for one-dimensional wave equation is the one written in equation (2):

$$c(x)^2 = \gamma(x)c_0^2 = \gamma(x)\frac{E}{\rho} \tag{2}$$

where $c_0$ is the constant wave speed defined as a function of density $\rho$ and elastic modulus $E$, instead $\gamma(x)$ is the parameter that is a function of the position $x$ and multiplied with the material property $c_0$, gives the actual material property of the domain in that position. Introducing this parametrization general wave equation (1) in one-dimensional form with variable material property is obtained:

$$u_{tt} = c_0^2\frac{\partial}{\partial x}(\gamma(x)\frac{\partial u}{\partial x}) + \frac{f}{\rho} \tag{3}$$

where $u$ is the displacement in x direction and $f$ is the force applied in the same direction.

The one-dimensional equation proves inadequate for adequately describing the behaviour of Lamb waves. To address this limitation, the wave equation transforms into a system of partial differential equations. In the simplest case, the two dimensions under consideration are length $x$ and thickness $z$ of a plate section with plane strain applied on the dimension $y$. This constraint renders the third equation zero, simplifying the model. To enhance our comprehension of the domain's orientation and shape, Figure (3) displays the axes and the plate. It's important to note that in this two-dimensional context, the $y$ direction is disregarded due to the imposition of plane strain, thereby rendering the partial differential equation in that direction negligible.

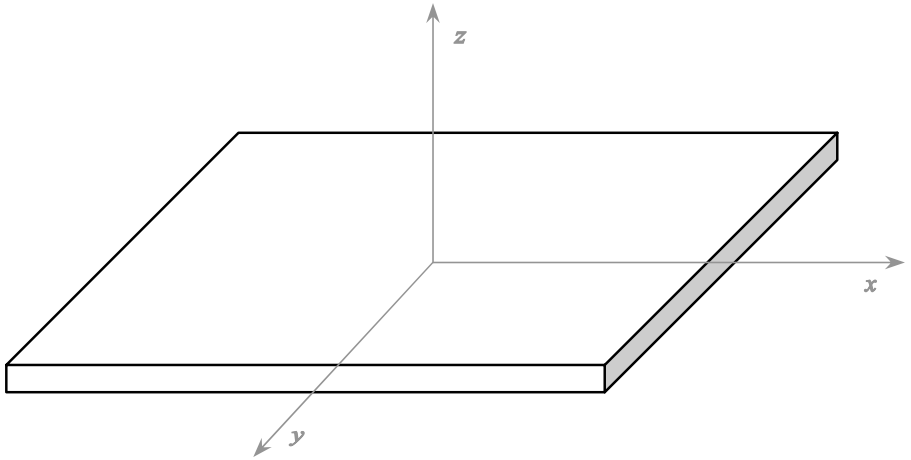


Figure 3: Plate coordinate system valid both for 2D and 3D

To simplify the treatment is useful to define $c_L$ and $c_T$ longitudinal wave speed and shear wave speed respectively:

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)} \tag{4}$$

$$c_L^2 = \frac{\lambda + 2\mu}{\rho}, \quad c_T^2 = \frac{\mu}{\rho} \tag{5}$$

where $\lambda$ and $\mu$ represent the Lamé parameters, another formulation of material properties in the general wave equation, all these parameters depend on the elastic properties of the materials i.e. elastic modulus $E$ and

Poisson ratio $\nu$. To consider variable material properties shear wave speed and longitudinal wave speed are parametrized as in equations (6).

$$c_L^2(x) = \gamma(x)c_L^2, \quad c_T^2(x) = \gamma(x)c_T^2 \tag{6}$$

Applying this substitution the wave equation in the direction $x$ and $z$ and their associated displacement $u$ and $w$ becomes the following system of equations:

$$\begin{aligned} u_{tt} &= (c_L^2\gamma u_x + (c_L^2 - 2c_T^2)\gamma v_y)_x + c_T^2(\gamma(u_y + v_x))_y + \frac{f_x}{\rho} \\ w_{tt} &= c_T^2(\gamma(u_z + w_x))_x + ((c_L^2 - 2c_T^2)\gamma u_x + c_L^2\gamma w_z)_z + \frac{f_z}{\rho} \end{aligned} \tag{7}$$

The three-dimensional model is obtained by adding the third spatial direction in the equation therefore the solution of this system of equations is the most general wave solution obtainable. Material properties are still a handful to write in the same way as equations (5) to get a clear formulation, therefore the parametrization of the properties is an extension of the two-dimensional case, the new parametrization is shown in equations (8).

$$c_L^2(x, y) = \gamma(x, y)c_L^2, \quad c_T^2(x, y) = \gamma(x, y)c_T^2 \tag{8}$$

Adding a new dimension lead to a system of three equation each for a displacement in a particular direction $u, v, w$ associated with the following direction $x, y, z$. The system of equations has the following form:

$$\begin{aligned} u_{tt} &= (c_L^2\gamma u_x + (c_L^2 - 2c_T^2)\gamma(v_y + w_z))_x + c_T^2(\gamma u_y)_y + c_T^2(\gamma u_z)_z + \frac{f_x}{\rho} \\ v_{tt} &= (c_L^2\gamma v_y + (c_L^2 - 2c_T^2)\gamma(u_x + w_z))_y + c_T^2(\gamma v_x)_x + c_T^2(\gamma v_z)_z + \frac{f_y}{\rho} \\ w_{tt} &= (c_L^2\gamma w_z + (c_L^2 - 2c_T^2)\gamma(v_y + u_x))_z + c_T^2(\gamma w_x)_x + c_T^2(\gamma w_y)_y + \frac{f_z}{\rho} \end{aligned} \tag{9}$$

Note that in both two-dimensional and three-dimensional equations, the actual dimensions of the parametrization are lower than the dimensions of the domain, this is the consequence of the choice made that material does not change along $z$ direction and thus $\gamma(x, y)$, the parameter $\gamma$ depends only on $x$ and $y$ coordinates.

Once introduced the physics of the problem is useful to present the other main argument of this paper which is a recently developed class of ML techniques. This paper is inspired by the application PINN which is a powerful and innovative approach that combines neural networks with physical principles to solve complex scientific and engineering problems. It bridges the gap between traditional physics-based models and data-driven ML techniques. In a PINN, neural networks are used to approximate the solution to a physical system while respecting the governing laws of physics. This is achieved by incorporating the equations describing the underlying physical phenomena as constraints during the training of the neural network. By doing so, PINNs can learn from limited or noisy data and extrapolate to predict the behaviour of the system in regions where data may be scarce. The key advantages of PINNs include their ability to efficiently solve partial differential equations, model complex systems, and adapt to various types of data. The structure of a neural network of this kind is typically an FFNN that receives in input the coordinates of the partial differential equation that is to be solved, the output is the solution of the partial differential equation in each coordinate. In Figure 4 is represented the net with inputs $\boldsymbol{x}$ and outputs $\boldsymbol{u}$ as explained before, at each step of the training both data and physics in form of partial differential equation residual are fed into the optimization algorithm in order to change the weights $\boldsymbol{\theta}$ of the net to best fit the solution of the partial differential equation that is being solved.

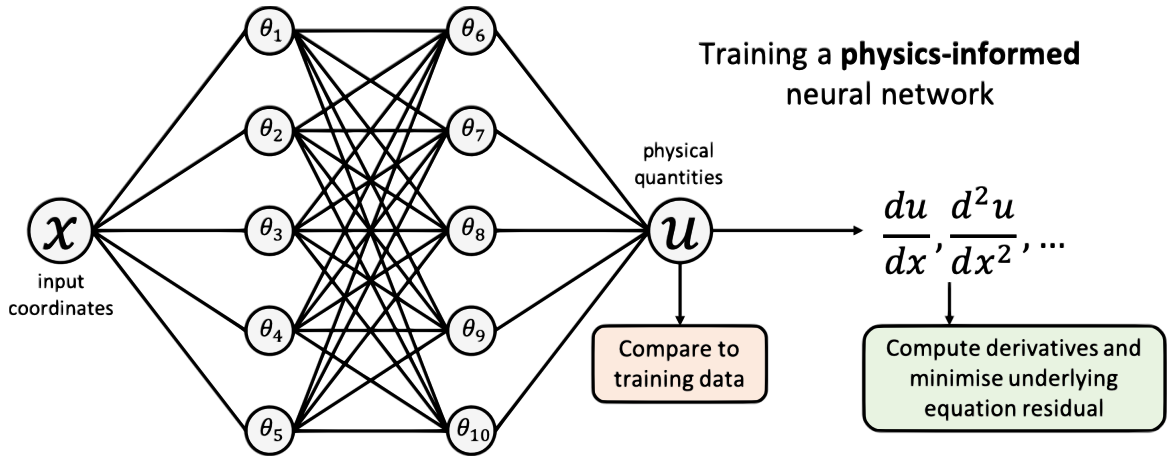


Figure 4: Simple scheme of a PINN that uses both data and physics during training

More complex PINN configurations are able to solve both the forward problem and inversion problem [37], the introduction of another net that predicts material properties is crucial to solving the inversion problem working alongside the net that solves the forward problem. This PINNs approach towards inversion approximate displacement $u$ and material distribution $\gamma$ using two neural networks $A_u$ and $A_\gamma$:

$$\begin{aligned} \boldsymbol{u} &= A_u(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{\theta_u}) \\ \boldsymbol{\gamma} &= A_\gamma(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{\theta_\gamma}) \end{aligned} \tag{10}$$

where $\boldsymbol{\theta} = \{\boldsymbol{\theta_u}, \boldsymbol{\theta_\gamma}\}$ are the model parameters of the neural network, $\boldsymbol{u}$ and $\boldsymbol{\gamma}$ in equation (10) represent the approximation provided by the net of the quantities. In order to set up the loss function for network training is useful to subdivide the loss terms contributions in the respective conditions on the partial differential equation problem:

$$\mathcal{N} = u_{tt}(\boldsymbol{x}, \boldsymbol{t}) - c_0^2 \frac{\partial}{\partial x}(\gamma(\boldsymbol{x}) \frac{\partial u(\boldsymbol{x}, t)}{\partial x}) - \frac{f}{\rho} \tag{11}$$

$$\mathcal{B} = u(\boldsymbol{x}, \boldsymbol{t}) \tag{12}$$

$$\begin{aligned} \mathcal{I}_0 &= u(\boldsymbol{x}, \boldsymbol{t}) \\ \mathcal{I}_1 &= u_t(\boldsymbol{x}, \boldsymbol{t}) \end{aligned} \tag{13}$$

where $\mathcal{N}$ is the residual of the partial differential equation, $\mathcal{B}$ are the boundary conditions and $\mathcal{I}$ are the initial conditions. Those equations are evaluated in $N$ points of the domain called collocation points defined in this way $N = N_\mathcal{N} + N_\mathcal{B} + N_\mathcal{I}$. Once defined these residuals the loss functional $\mathcal{L}$ is then built considering the mean square of those residuals evaluations (the differential equation residual $\mathcal{L}_\mathcal{N}$, the boundary conditions residual $\mathcal{L}_\mathcal{B}$, and the initial condition residual $\mathcal{L}_\mathcal{I}$) at the collocations points:

$$\mathcal{L}_\mathcal{N}(\boldsymbol{u}, \boldsymbol{\gamma}) = \frac{1}{N_\mathcal{N}} \sum_{i=1}^{N_\mathcal{N}} (\mathcal{N}[u_i; \gamma_i])^2 \tag{14}$$

$$\mathcal{L}_\mathcal{B}(\boldsymbol{u}) = \frac{1}{N_\mathcal{B}} \sum_{i=1}^{N_\mathcal{B}} (\mathcal{B}[u_i])^2 \tag{15}$$

$$\mathcal{L}_\mathcal{I}(\boldsymbol{u}) = \frac{1}{N_{\mathcal{I}_0}} \sum_{i=1}^{N_{\mathcal{I}_0}} (\mathcal{I}_0[u_i])^2 + \frac{1}{N_{\mathcal{I}_1}} \sum_{i=1}^{N_{\mathcal{I}_1}} (\mathcal{I}_1[u_i])^2 \tag{16}$$

$$\mathcal{L}(\boldsymbol{u}, \boldsymbol{\gamma}) = \mathcal{L}_\mathcal{N}(\boldsymbol{u}, \boldsymbol{\gamma}) + \mathcal{L}_\mathcal{I}(\boldsymbol{u}) + \mathcal{L}_\mathcal{B}(\boldsymbol{u}) \tag{17}$$

In this way the optimization problem is set for a PINN approach, and the minimization of the loss functional (17) lead to the solution of both inverse and direct problem improving at each step the approximation of both neural networks. This is the traditional PINN approach to solve inverse problems for wave equation as presented in [36].

## 2.2. Workflow

In order to develop a class of methods that solve the inverse problem is necessary to create a finite difference solver that is fast and accurate enough to be embedded in the code and used alongside the ML tools. Three different solvers are presented increasing the spatial dimension considered at each step and thus complexity.

One dimensional finite difference solver is the numerical effort to solve wave equation (3), the domain selected is a one-dimensional string along dimension $x$ with the edges fully constrained and thus the displacement $u$ at those points is set to zero. The wave equation is then solved in the finite difference framework using the following stencils for partial differential equation [38], initial and boundary conditions:

$$u_i^{n+1} = 2u_i^n - u_i^{n-1} + c^2 \frac{dt^2}{2dx^2}[(\gamma_i + \gamma_{i+1})(u_{i+1}^n - u_i^n) - (\gamma i + \gamma_{i-1})(u_i^n - u_{i-1}^n)] + \frac{dt^2}{\rho} f_i^n \tag{18}$$

$$BC: \quad u_0^n = 0, \quad u_{N_x}^n = 0 \tag{19}$$

$$IC: \quad u_i^0 = 0 \tag{20}$$

where $i$ is the index for the space position node and assumes values between $i = 0$ and $i = N_x$, while $n$ is the index for time, varying across $n = 0$ to $n = N_t$. Consider that $N_t + 1$ is the number of time nodes and $N_x + 1$ is the number of spatial nodes of the grid, as a consequence $dx$ and $dt$ are defined as the step in respectively space and time, computed as follows:

$$dx = \frac{L_x}{N_x}, \quad dt = \frac{L_t}{N_t} \tag{21}$$

where $L_x$ is the length of the domain in $x$ direction, while $L_t$ is the end time of the simulation. The stencil of the initial condition on the derivative is not important because the derivative of a null quantity is already zero.

The second solver developed is capable of solving two-dimensional wave equation presented in equations (7), to solve a meaningful problem that allow Lamb waves propagation is necessary to define the domain as mentioned in Subsection 2.1. Lamb waves are a phenomenon typical of thin plates and thus boundary condition are chosen accordingly, the plate is constrained on the two edges and thus no displacement both $u$ and $w$ is null while the upper and lower surface are free to move i.e. $z = 0$ and $z = N_z$ surfaces. Those boundaries require special treatment to impose stress-free conditions represented by the following equations:

$$\begin{gathered} \sigma_{zz}|_{z=0} = 0 \quad becomes \quad (c_L^2 - 2c_T^2)w_z + c_L^2 u_x = 0 \\ \sigma_{zx}|_{z=0} = 0 \quad becomes \quad u_z + w_x = 0 \end{gathered} \tag{22}$$

$$\begin{gathered} \sigma_{zz}|_{z=N_z} = 0 \quad becomes \quad (c_L^2 - 2c_T^2)w_z + c_L^2 u_x = 0 \\ \sigma_{zx}|_{z=N_z} = 0 \quad becomes \quad u_z + w_x = 0 \end{gathered} \tag{23}$$

where $\sigma$ is the stress in the direction indicated by the respective indices evaluated in a particular position. Equations (22) and (23) are directly obtained from constitutive relations that relate stress $\boldsymbol{\sigma}$ and strain $\boldsymbol{\epsilon}$ then is used strain/displacement equation that allows to write those equations in term of displacement $u, w$ and their derivatives. Starting from the partial differential equations described in (7) the finite difference stencil is derived similarly to [39]:

$$\begin{aligned} u_{i,k}^{n+1} = {} & 2u_{i,k}^n - u_{i,k}^{n-1} + \\ & c_L^2 \frac{dt^2}{2dx^2}[(\gamma_{i,k} + \gamma_{i+1,k})(u_{i+1,k}^n - u_{i,k}^n) - (\gamma_{i,k} + \gamma_{i-1,k})(u_{i,k}^n - u_{i-1,k}^n)] + \\ & (c_L^2 - 2c_T^2)\frac{dt^2}{8dxdz}((\gamma_{i,k} + \gamma_{i+1,k})(w_{i+1,k+1}^n - w_{i+1,k-1}^n) - (\gamma_{i,k} + \gamma_{i-1,k})(w_{i-1,k+1}^n - w_{i-1,k-1}^n)) + \\ & c_T^2 \frac{dt^2}{2dz^2}[(\gamma_{i,k} + \gamma_{i,k+1})(u_{i,k+1}^n - u_{i,k}^n) - (\gamma_{i,k} + \gamma_{i,k-1})(u_{i,k}^n - u_{i,k-1}^n)] + \\ & dt^2 \frac{fx_{i,j}^n}{\rho} \end{aligned} \tag{24}$$

$$
\begin{aligned}
w_{i,k}^{n+1} = {} & 2w_{i,k}^{n} - w_{i,k}^{n-1} + \\
& c_L^2 \frac{dt^2}{2dz^2}[(\gamma_{i,k} + \gamma_{i,k+1})(w_{i,k+1}^{n} - w_{i,k}^{n}) - (\gamma_{i,k} + \gamma_{i,k-1})(w_{i,k}^{n} - w_{i,k-1}^{n})] + \\
& (c_L^2 - 2c_T^2)\frac{dt^2}{8dxdz}((\gamma_{i,k} + \gamma_{i,k+1})(u_{i+1,k+1}^{n} - u_{i+1,k-1}^{n}) - (\gamma_{i,k} + \gamma_{i,k-1})(u_{i-1,k+1}^{n} - u_{i-1,k-1}^{n})) + \\
& c_T^2 \frac{dt^2}{2dx^2}[(\gamma_{i,k} + \gamma_{i+1,k})(w_{i+1,k}^{n} - w_{i,k}^{n}) - (\gamma_{i,k} + \gamma_{i-1,k})(w_{i,k}^{n} - w_{i-1,k}^{n})] + \\
& dt^2 \frac{fz_{i,j}^{n}}{\rho}
\end{aligned}
\tag{25}
$$

where $i$ is the spatial position index associated with $x$ direction varying between $i = 0$ and $i = N_x$, similarly $k$ is the spatial position related to direction $z$ assuming values between $k = 0$ and $k = N_z$. Consider that $N_t + 1$ is the number of time nodes, $N_x + 1$ is the number of spatial nodes of the grid in $x$ direction while in thickness direction $z$ the number of nodes is $N_z + 1$, as a consequence $dx$, $dz$ and $dt$ are defined as the step in respectively space and time, computed as follows:

$$
dx = \frac{L_x}{N_x}, \quad dz = \frac{L_z}{N_z}, \quad dt = \frac{L_t}{N_t} \tag{26}
$$

where $L_x$ is the length of the domain in $x$ direction, $L_z$ is the length in $z$ direction and $L_t$ is the end time of the simulation. The time grid is described by index $n$ as before. Concerning boundary conditions the free surfaces discussed before are implemented using the following stencil for both surfaces changing only the $z$ index. Initial conditions are enforced in the same way as the one-dimensional solver and for the sake of simplicity are not reported.

$$
\begin{aligned}
BC: \; u_{i,0}^{n} &= \frac{dz}{2dx}(w_{i+1,1}^{n} - w_{i-1,1}^{n}) + u_{i,1}^{n} \\
w_{i,0}^{n} &= \frac{c_L^2 - 2c_T^2}{c_L^2}\frac{dz}{2dx}(u_{i+1,1}^{n} - u_{i-1,1}^{n}) + w_{i,1}^{n} \\
u_{i,N_z}^{n} &= \frac{dz}{2dx}(w_{i+1,N_z-1}^{n} - w_{i-1,N_z-1}^{n}) + u_{i,N_z-1}^{n} \\
w_{i,N_z}^{n} &= \frac{c_L^2 - 2c_T^2}{c_L^2}\frac{dz}{2dx}(u_{i+1,N_z-1}^{n} - u_{i-1,N_z-1}^{n}) + w_{i,N_z-1}^{n}
\end{aligned}
\tag{27}
$$

$$
u_{0,k}^{n} = 0, \;\; w_{0,k}^{n} = 0, \;\; u_{N_x,k}^{n} = 0, \;\; w_{N_x,k}^{n} = 0 \tag{28}
$$

The group of equations (27) represent the stencil for stress-free boundary conditions while equations (28) represent the null displacement conditions at the sides with normal $x$.

The three-dimensional equation is solved by means of the finite difference method considering a plate that is constrained at the four sides and therefore no displacement is allowed on that surface, i.e., surfaces with normal $x$ and $y$. Similarly to the two-dimensional case surface free boundary condition has to be applied to the free surface of the plate i.e. surfaces with normal $z$. This condition is implemented employing equations (29) for surface with $z = 0$ and equations (30) for surface with $z = N_z$.

$$
\begin{aligned}
\sigma_{zz}|_{z=0} = 0 \quad becomes \quad (c_L^2 - 2c_T^2)w_z + c_L^2 u_x + c_L^2 v_y = 0 \\
\sigma_{zx}|_{z=0} = 0 \quad becomes \quad w_x + u_z = 0 \\
\sigma_{zy}|_{z=0} = 0 \quad becomes \quad w_y + v_z = 0
\end{aligned}
\tag{29}
$$

$$
\begin{aligned}
\sigma_{zz}|_{z=N_z} = 0 \quad becomes \quad (c_L^2 - 2c_T^2)w_z + c_L^2 u_x + c_L^2 v_y = 0 \\
\sigma_{zx}|_{z=N_z} = 0 \quad becomes \quad w_x + u_z = 0 \\
\sigma_{zy}|_{z=N_z} = 0 \quad becomes \quad w_y + v_z = 0
\end{aligned}
\tag{30}
$$

The stencils of the three-dimensional finite difference solver are not reported for the sake of simplicity, they can be easily derived from the system of equations (9) by applying finite difference schemes to the derivatives. In the same way boundary condition stencils are obtained by performing those operations on equations grouped

in (29) and (30).

Methods developed make use of a neural network that predicts the distribution of the material on the domain, they differ on how the loss function is defined. Prediction of material distribution is devoted to an FFNN with a simple architecture, the spatial position is fed in the network and the output is the material distribution $\boldsymbol{\gamma}$ a function that multiplied to the material property gives the material property in that particular point. A simplified physics-informed method uses displacement measurement $\boldsymbol{u_m}$ coming from sensors placed on the domain and the material distribution prediction $\boldsymbol{\gamma}$ from the neural network $A_\gamma$ to compute the residual of the partial differential equation. This residual represents the loss function that is minimized by the optimization algorithm modifying neural network parameters $\boldsymbol{\theta}$. The loss functional is described by the following equation:

$$loss = \frac{\rho}{dt^2}(u_i^{n+1} - 2u_i^n + u_i^{n-1}) - \rho c^2(\frac{\partial \gamma}{\partial x}\frac{(u_{i+1}^n - u_{i-1}^n)}{2dx} + \frac{u_{i+1}^n - u_i^n + u_{i-1}^n}{dx^2}) - f_i^n \tag{31}$$

$$l_{pde} = \frac{1}{n_m}\sum_{i=0}^{n_m} loss^2 \tag{32}$$

Spatial and temporal derivatives with respect to displacement are computed using a finite difference scheme prone to numerical errors, whereas the one relative to material distribution i.e $\frac{\partial \gamma}{\partial x}$ is computed using automatic differentiation to ensure high accuracy considering that this derivative is the one used by the optimization algorithm to improve $\boldsymbol{\gamma}$ prediction. This very simple and computationally inexpensive method has some limitations in particular derivatives computed with finite differences to be accurate need a fine grid, not feasible on a real component due to limitations on the number of sensors that can be installed on it. A common solution is to use another network with collocation points that predict displacement where sensors are not present, similar to the classical PINNs approach for the inversion problem explained in Subsection 2.1. This approach has been extensively tested by [36]. Figure 5 represents the scheme of the method above where the main steps are in the following order: spatial coordinates of the domain $x$ are fed into the neural network $A_\gamma$, the output is the material distribution $\boldsymbol{\gamma}$ that is then used to compute loss function following equations (31) and (32). Loss value is then necessary to perform the backpropagation step where neural network parameters are optimized to fit with the solution.

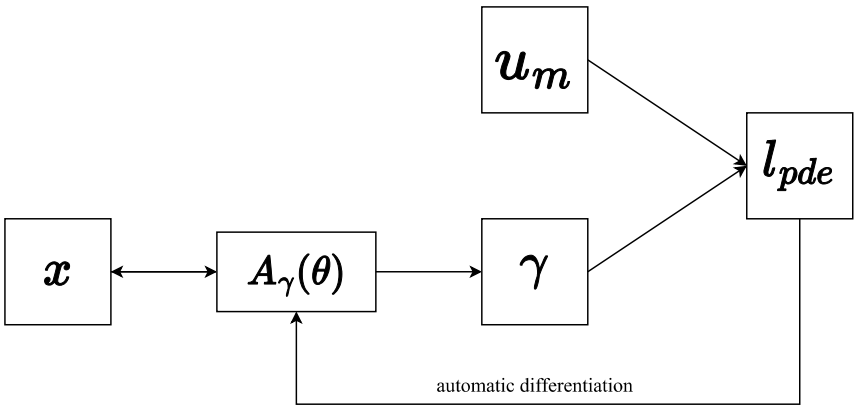


Figure 5: Simplified traditional physics-informed method

In order to overcome the limitations presented in the previous method the Physics-informed method coupled with a numerical solver is developed starting from the work done in this paper [37]. The original method was modified so it could deal with Lamb waves. Such a method is selected due to the reduction in complexity of the optimization process compared to a more traditional PINN approach that uses a neural network to solve both forward and inverse problems as shown in Subsection 2.1. This method implements a finite difference solver (Forward Solver) that computes the wave field $u$ using material distribution $\boldsymbol{\gamma}$ predicted from the neural network, in this way the partial differential equation is always satisfied and therefore the loss function is the mean of the square difference between predicted displacement in the sensors points $\boldsymbol{u}(\boldsymbol{x_m})$ and the actual displacement in those points $\boldsymbol{u_m}$:

$$l_m = \frac{1}{n_m}\sum_{i=0}^{n_m}[u(x_m) - u_m]^2 \tag{33}$$

In equation (33) $n_m$ represents the number of sensors installed on the domain. Minimizing this relation the solution predicted by the neural network tends to the unknown material distribution, the network makes a prediction of material distribution $\boldsymbol{\gamma}$ given spatial position $\boldsymbol{x}$, material distribution predicted is then fed with spatial position $\boldsymbol{x}$ and time $\boldsymbol{t}$ in the forward solver to obtain partial differential equation solution. The last step involves confronting the displacement field computed with material distribution predicted from the net $\boldsymbol{u}$ and the one measured by the sensors $\boldsymbol{u_m}$, to do this the loss function has equation (33), then by means of backpropagation algorithm weights and biases of the net are tuned to reduce as much as possible the loss function. In this way the material distribution predicted converges towards the actual material properties distribution, nevertheless local minimum in the loss function can prevent the optimization process from reaching the actual solution.

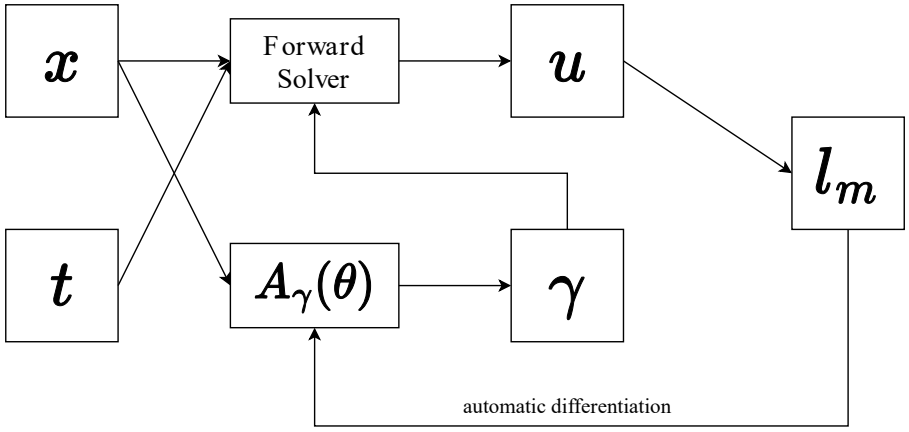


Figure 6: Physics-informed method coupled with finite difference solver

# 3. Case studies

## 3.1. Solver validation: comparison with Abaqus

The first model to be tested and validated is the one-dimensional finite difference solver that is capable of solving the one-dimensional wave equation with certain initial and boundary conditions as mentioned in Subsection 2.2. The domain is one dimensional and thus wave propagates across this dimension $x$ and thus to excite the wave, force is applied on the middle of the domain with direction $x$. The forcing term applied is a sine wave modulated by the Hanning window described by Equation (34), in this particular case the frequency is set to $f = 200kHz$ and the number of cycles $n_{cycles} = 4$.

$$f(t) = \begin{cases} \sin(2\pi f t)\sin(\frac{\pi f t}{n_{cycles}}), & \text{for} \quad 0 \leq t \leq \frac{n_{cycles}}{f} \\ 0, & \text{for} \quad \frac{n_{cycles}}{f} < t \end{cases} \tag{34}$$

The forcing term signal is represented in Figure 7, the one employed to validate the solvers presented.

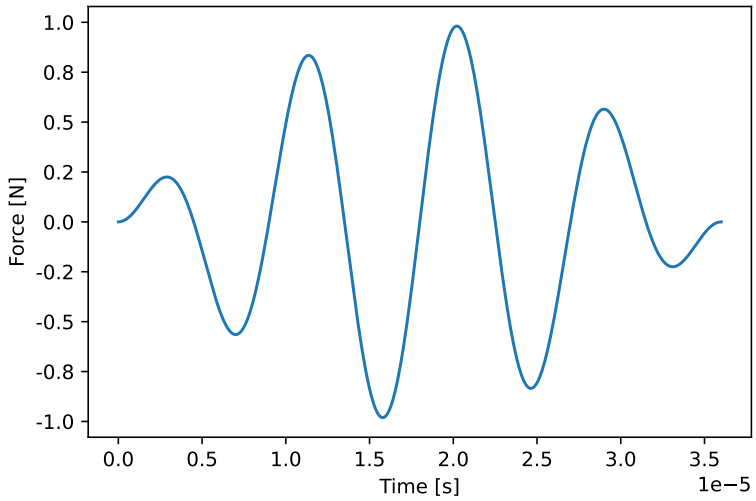


Figure 7: Forcing term

Validation is carried out by comparing the results of the finite difference solver developed and the Abaqus finite element method solution in different positions of the domain with variable material properties. Geometry and other parameters are reported in Table 1.

**String parameters**

| | |
|---|---|
| **Elastic modulus 1** $[GPa]$ | $E_1 = 70$ |
| **Elastic modulus 2** $[GPa]$ | $E_2 = 35$ |
| **Density** $[Kg/m^3]$ | $\rho = 2700$ |
| **Length x** $[cm]$ | $L_x = 30$ |
| **Position of material change** $[cm]$ | $L_x = 7.5$ |
| **Simulation time** $[s]$ | $3.6 \cdot 10^{-5}$ |
| **dt** $[s]$ | $2 \cdot 10^{-8}$ |

Table 1: Property and parameters of the model

Abaqus model is a wire geometry with explicit beam elements to simulate correctly the wave propagation, the section assigned to the beam is a square with side $l = 1\ \ cm$. The two edges of the beam are both constrained with no displacement allowed. Sensors and material distribution are shown in Figure 8, the black part of the domain represents a material with elastic modulus $E_1$ and the yellow part is the one with $E_2$.

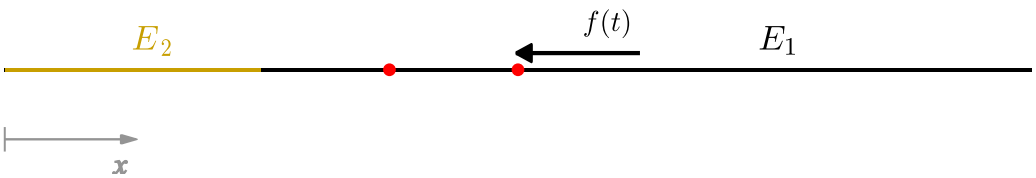


Figure 8: Domain configuration, each colour represents a different material, red dots are the positions of the sensors

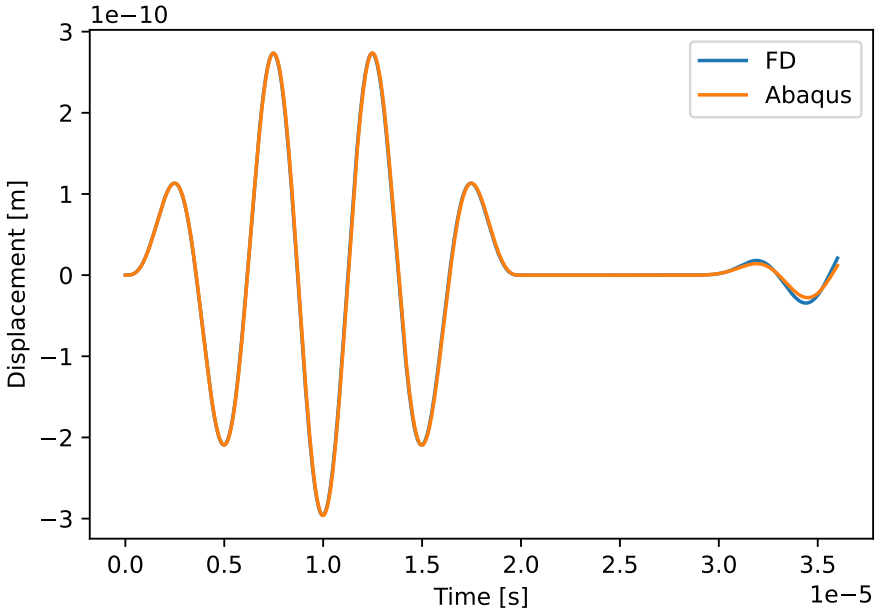


Figure 9: Sensor displacement measurement, positioned where the force is applied

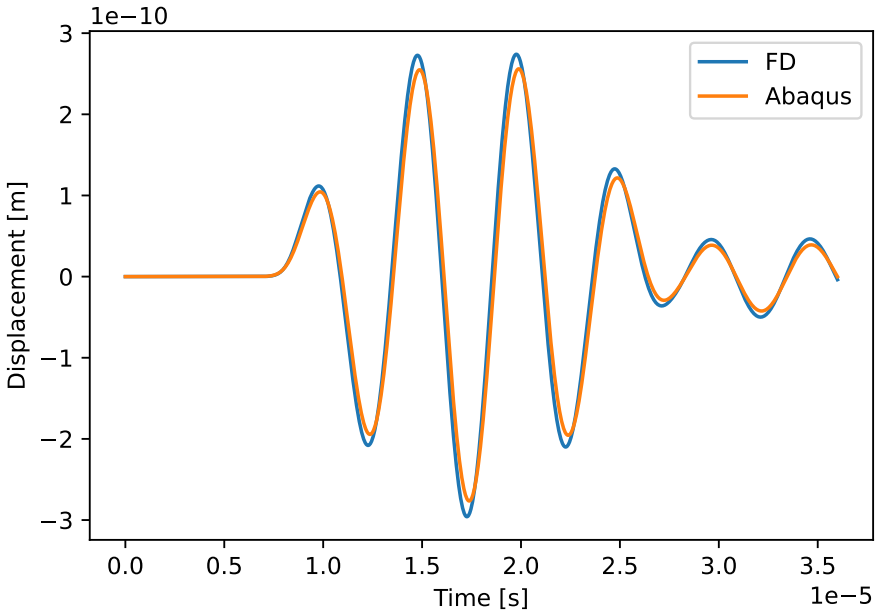


Figure 10: Sensor displacement measurement, positioned at 3.75 $[cm]$ from the source position

The measured displacement in the position of the sensors shows almost the same results, wave propagation speed is coherent with a chosen material, waveform remains the same in both the simulations, and a similar discussion is for amplitude attenuation that shows similar behaviour. The wave reflected from the change in wave speed is the second wave shown in both Figure 9 and 10, once has been reflected comes back and thus can be measured from the sensors.

The finite difference solver in three dimensions capable of simulating Lamb wave propagation in a plate is tested and compared to the Abaqus finite element software with the aim of validating the results obtained by the tool developed. Lamb wave type excited is symmetrical mode S0 due to limitation introduced by discretization of surface free boundary condition that is not accurate enough to show dispersion behaviour in particular for anti symmetrical mode A0, more information on dispersion phenomenon is reported in Appendix B. The forcing frequency utilized ranged from 200 kHz to 400 kHz, a value that appears to be in the region of lowest dispersion for S0 mode while this region for A0 mode has a much higher frequency. Therefore symmetrical mode S0 prove to be the right choice in our scenario to simulate realistic Lamb wave propagation. Geometry and parameters

selected for this comparison are reported in Table 2.

**Plate parameters**

| | |
|---|---|
| **Elastic modulus 1 [$GPa$]** | $E_1 = 70$ |
| **Elastic modulus 2 [$GPa$]** | $E_2 = 35$ |
| **Poisson ratio** | $\nu = 0.33$ |
| **Density [$Kg/m^3$]** | $\rho = 2700$ |
| **Length x [$cm$]** | $L_x = 30$ |
| **Length y [$cm$]** | $L_y = 30$ |
| **Thickness [$cm$]** | $L_z = 0.1$ |
| **Position of material change [$cm$]** | $L_x = 7.5$ |
| **Simulation time [$s$]** | $3.6 \cdot 10^{-5}$ |
| **dt [$s$]** | $2 \cdot 10^{-8}$ |

Table 2: Property and parameters of the model

Shell continuum with explicit step is selected for the Abaqus model in order to maintain the model as simple as possible and computationally fast to solve, the number of elements differs due to the necessity in finite difference solver of having more than one node along thickness to apply boundary conditions schemes, while shell elements in Abaqus can have only one element in that direction. The forcing term is the same as the one selected in 1D validation and therefore follows equation (34). This force is applied symmetrically in order to excite the S0 Lamb wave as shown in Figure 11 and both are positioned in the middle of the plate.

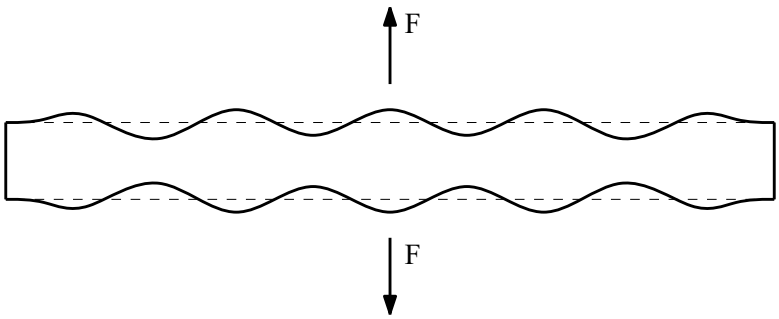


Figure 11: Symmetrical mode S0 excitation scheme

Different sensors are placed to compare the wave obtained by solving the partial differential equations with those two different approaches, in Figure 13 represents the displacement as a function of time in the position where the force is applied on the surface of the plate. While in Figure 14 the sensor is placed at a distance of $x = 3.75$ $[cm]$ from the source of the signal, the first wave is the one emitted by the source while the two peaks following is the wave reflected by the change in material properties. Figure 12 shows the top view of the plate with sensor positions and material distribution highlighted.

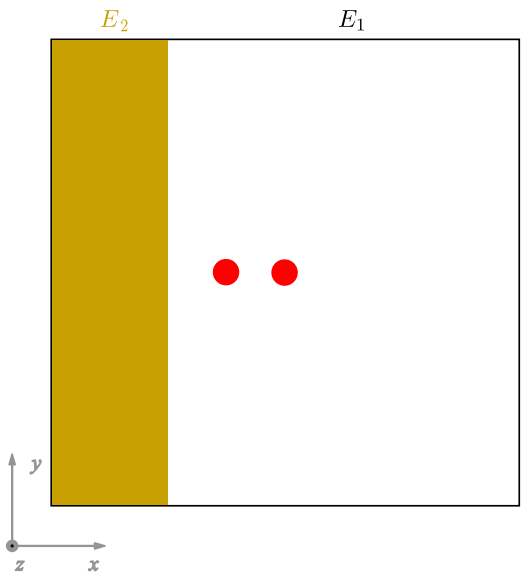


Figure 12: Plate domain configuration, each color represent a different material, red dots are sensors positions

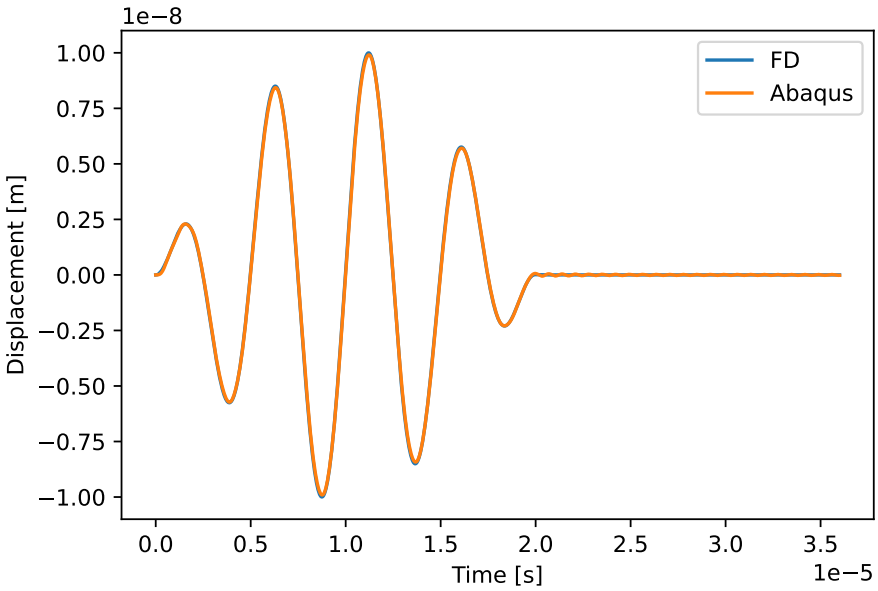


Figure 13: Sensor displacement measurement, positioned where the force is applied

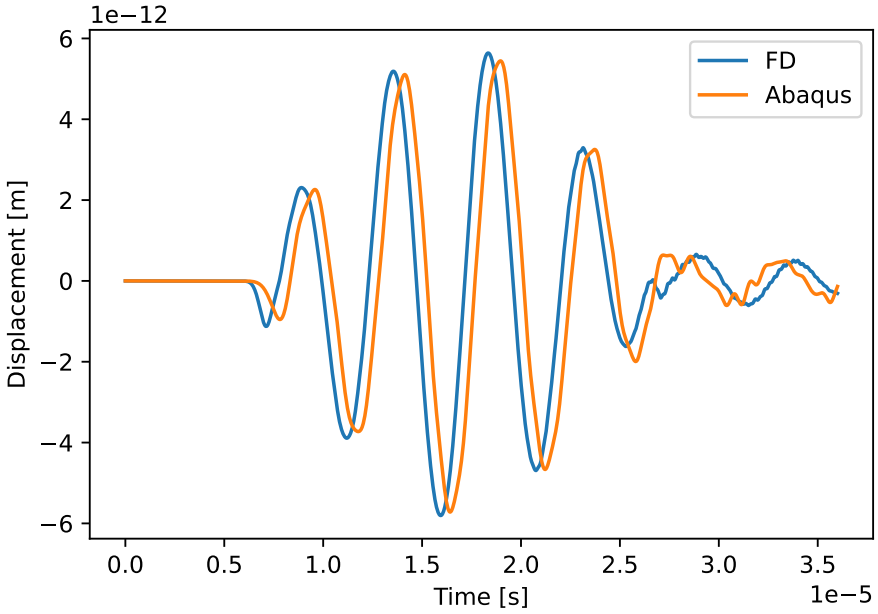


Figure 14: Sensor displacement measurement, positioned at 3.75 $[cm]$ from the source position

Though the results are not completely similar, the waveform behaves in the same way in both simulations, the propagation speed is very similar, and the only notable difference is the phenomenon of amplitude attenuation that shows some differences accountable to slight differences in element number and type, that are the essential differences between the two methods. The finite difference solver presented has some limitations related to the type of Lamb waves simulated, while the solver works numerically fine in simulating A0 mode, the wave propagating does not show the same dispersion as the one simulated with Abaqus. This is due to the surface-free boundary condition schemes that are not accurate enough to simulate the correct dispersion behaviour as indicated by [40].

## 3.2. 1D

The first model developed and analyzed is the one-dimensional model with two variants studied in order to assess the capabilities and performance of the methods presented in this paper. Simulating a wave in one-dimensional domain is comparably simple to propagating Lamb waves in a plate and thus makes the one-dimensional case a perfect environment to compare the two methods.
In order to discuss the results obtained is necessary to specify the geometry and the important parameters of the two models, the domain can be considered a beam clamped at both ends with the property listed in Table 3, where a wave excited in the middle of the domain propagates along $x$ direction, the displacement produced by the wave is in the same direction of the propagation direction i.e. $x$. Sensors are equally spaced along the beam length and record the displacement $u$ in time, note that the sensor readings are computed using the finite difference solvers developed once validated to simplify the testing phase.

**String parameters**

| | |
|---|---|
| **Wave speed**$[m/s]$ | $c = 5000$ |
| **Wave speed damage**$[m/s]$ | $c_d = 2500$ |
| **Density** $[Kg/m^3]$ | $\rho = 2700$ |
| **Length** $[cm]$ | $L_x = 10$ |
| **Number of Sensors** | 9 |
| **Damage position**$[cm]$ | $6.5 - 7.5$ |

Table 3: Property and parameters of the model

The domain and sensors' position is shown in detail in Figure 15, note that sensors at the tips of the beam are not useful considering the boundary condition selected being null displacement, for this reason, no displacement would be measured and thus no contribution in the loss function.

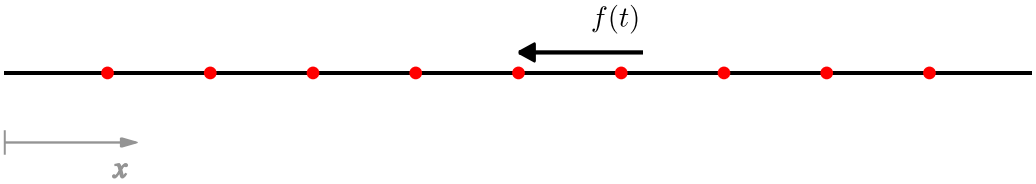


Figure 15: Domain configuration, the black line is the beam, while the red dots represent the sensors

The neural network used to predict material distribution has one input which is the spatial coordinate $x$ and one output, the material distribution $\gamma$. In Figure 16 is visible the structure of the neural network employed in this section consists of 1 neuron at the input layer, 4 hidden layers with, respectively, 5,10,10,5 neurons and the output layer with 1 output neuron. The activation function of the neurons is a sigmoid except for the output neuron which has no activation function.

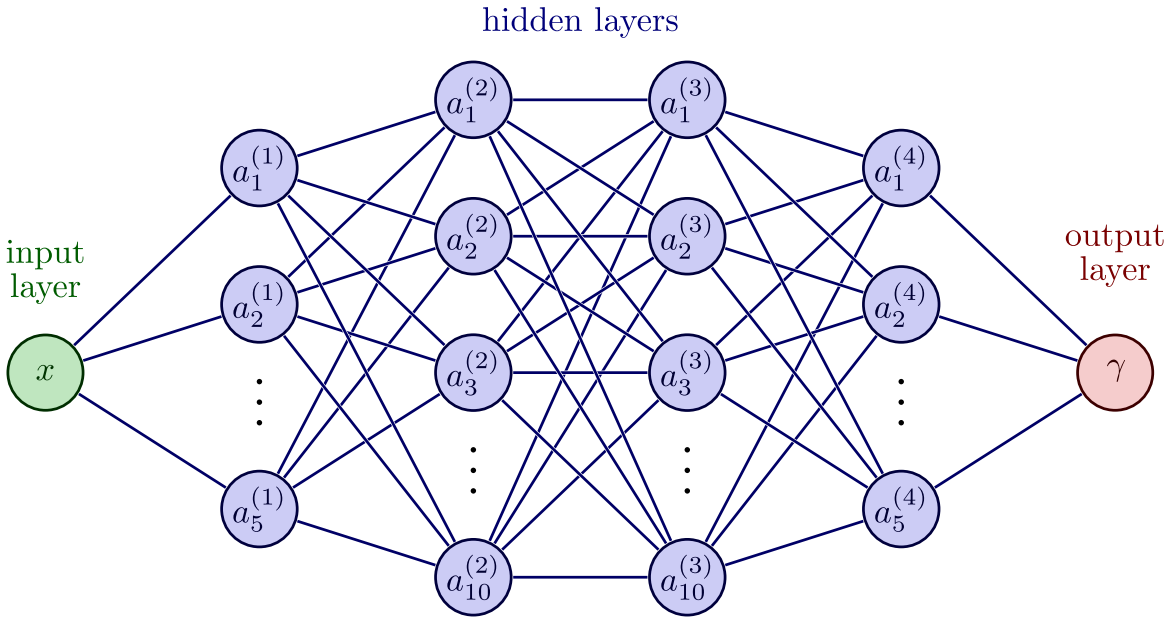


Figure 16: Neural network structure for 1D and 2D

The simplified physics-informed method is applied following the procedure shown previously, the major difficulty related to this method is the error introduced by computing displacement $u$ derivative in the loss function equation (31) using finite difference schemes. Problems arise when the number of measuring points is reduced, the solution becomes unstable and the algorithm is no longer capable of identifying the damage. This issue can be mitigated by reducing the frequency of the forcing term, for more information about this behaviour see Appendix A. Therefore reducing the number of sensors down to 9 needs a reduction of the frequency with respect to the one used typically in Lamb wave studies, in order to maintain the derivative computed with finite difference accurate enough to allow the algorithm to reach a satisfying solution. Forcing term is a sine wave modulated with Hanning window of frequency $f = 25 \ \ kHz$ with 3 cycles, using this configuration condition (37) is satisfied.
The results obtained are reported in Figure 17, note that this method is not capable of predicting material distribution outside the point where the sensor is placed. The training process takes 200,000 epochs in around 500 seconds, with a learning rate $l_r = 0.005$ with step decrease down to $l_r = 3.9 \cdot 10^{-5}$ value where the loss stabilizes.

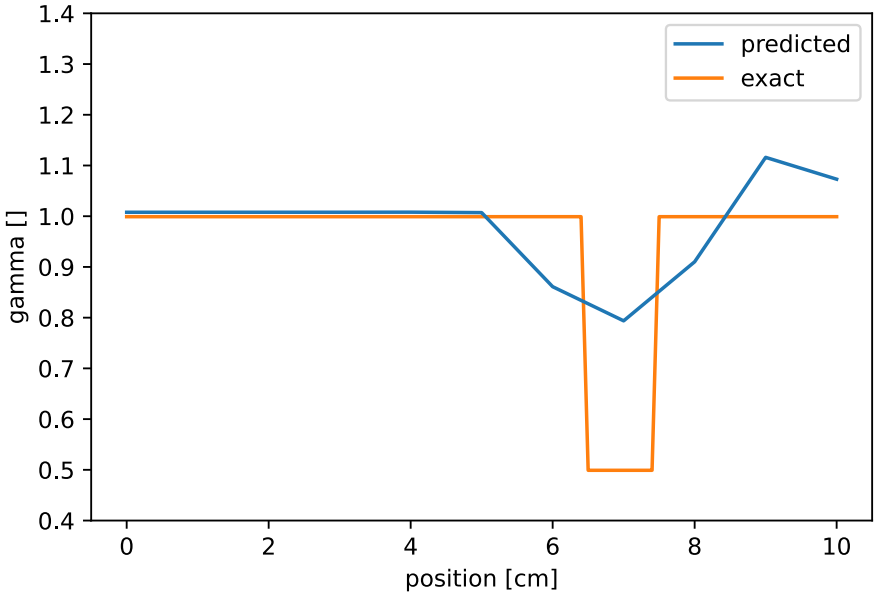


Figure 17: Simplified physics-informed method material distribution prediction after training

Physics-informed coupled with finite difference solver method has far less limitation than the other method hence allows to use of a forcing term with the desired frequency, more suitable for a Lamb waves study, that is sine wave modulated with Hanning window of frequency $f = 300 \ \ kHz$ and 3 cycles. The results obtained by the training process are represented in the following Figure 18. Epochs needed to train the neural network are around 10,000 epochs completed in 2000 seconds. The learning rate selected is the same as the other method $l_r = 0.005$ with step decay down to $l_r = 3.2 \cdot 10^{-4}$ to reach a stable loss value.

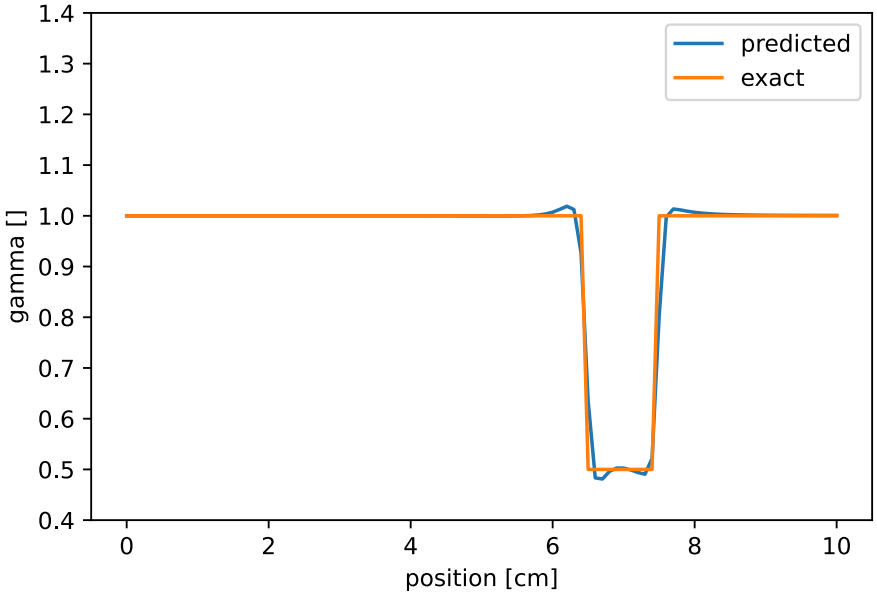


Figure 18: Physics-informed coupled with finite difference solver method material distribution prediction after training

Comparing the solution obtained by the different methods is evident that the solution provided by Physics-informed coupled with the finite difference method is much more accurate allowing to use wider range of frequencies and a lower amount of sensors maintaining a good level of accuracy in the prediction of material distribution. This advantage comes with an increased computational cost, in more complex scenarios this can become a great limitation for this method. In fact, the time needed to perform a fine prediction by the last method presented is 4 times greater than the time occupied by the other method.

### 3.3. 2D

The natural step following mono-dimensional analysis is introducing the second dimension to consider more complex waves such as Lamb waves, following the hypothesis introduced with the two-dimensional finite difference solver presented in Section 2.2 is possible to prove the capability of this method in predicting material distribution and hence damages in presence of Lamb waves. The method applied is the Physics-informed coupled with solver method preferred to the traditional one due to difficulties in treating waves with high frequency typical of Lamb waves. Another important aspect that rendered impossible the use of such a method is the addition of the equation for displacement $w$ in $z$ direction, to compute the residual of this equation to build the loss function required displacement in that direction, this leads to the necessity of measuring $w$ through the thickness of the plate that is not feasible in a real scenario. The increased computational cost of this new solver leads to the necessity of reducing the domain to keep the algorithm fast enough, in Table 4 are reported the property and the parameters of the model. Sensors are evenly spaced and measure displacement at the surface of the plate. The geometry selected is, as anticipated in Subsection 2.1, a plate section obtained by enforcing plane strain condition that renders the three-dimensional plate problem a two-dimensional problem, simplifying greatly the formulation.

**Plate parameters**

| | |
|---|---|
| **Elastic modulus** $[GPa]$ | $E = 70$ |
| **Poisson ratio** | $\nu = 0.33$ |
| **Density** $[Kg/m^3]$ | $\rho = 2700$ |
| **Length** $[cm]$ | $L_x = 5$ |
| **Thickness** $[cm]$ | $L_z = 0.1$ |
| **Number of Sensors** | 4 |
| **Damage position** $[cm]$ | $3 - 3.75$ |

Table 4: Property and parameters of the model

To better visualize the position of the sensor and the domain itself comes in handy illustration in Figure 19, sensors are placed on one surface of the plate, is also possible to observe the particular force configuration that is specifically designed to excite Lamb wave symmetrical mode S0, to do this in the middle of the plate at the two free surfaces the same force is placed but with opposite direction. The force is a sine wave modulated with a Hanning window of frequency $f = 400\ \ kHz$ and 3 cycles.

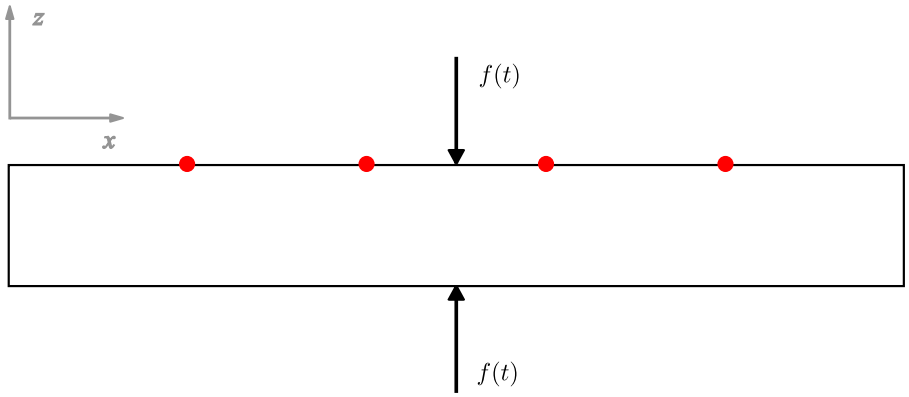


Figure 19: Domain configuration, the rectangle is the plate section while red dots are the sensors

The neural network employed is the same as the one in Subsection 3.2, material distribution is predicted only in the $x$ direction since the objective of the work is to detect damaged position and hence is assumed that the material is constant along thickness direction $z$. Material distribution obtained by running the algorithm is shown in Figure 20 compared to the expected one. Increased complexity of the solver greatly increases the time needed to train the network, maintaining the same number of epochs around 10,000 the time needed becomes 20,000 s.

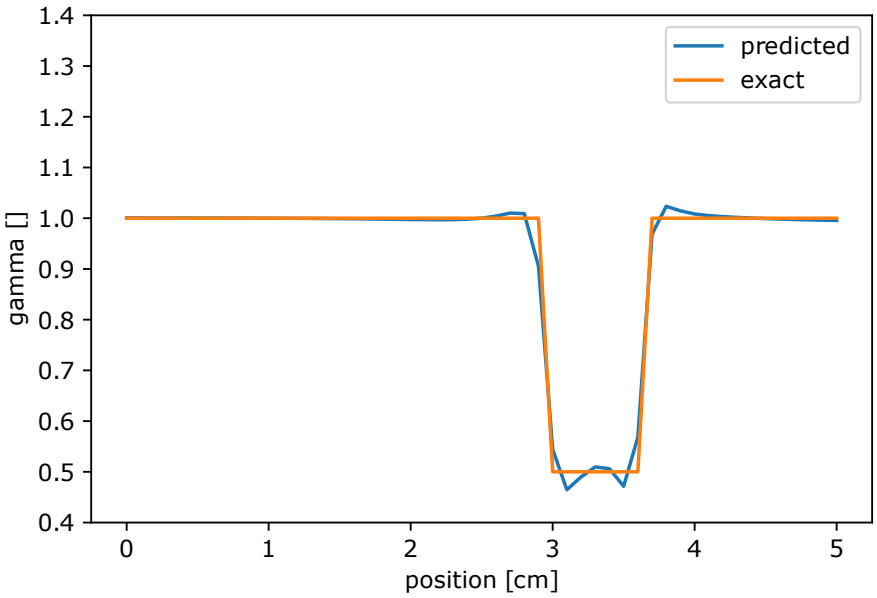


Figure 20: Neural network material distribution prediction after training

In this particular case the network prediction is able to detect accurately the discontinuity created by the presence of the damage and furthermore quantify correctly the entity of the material property discontinuity both in value, position and length.

## 3.4. 3D

In order to prove this framework on a case as close as possible to reality, the algorithm has to run on a full plate with a damaged region present to evaluate its performance. This is possible by implementing the finite difference solver presented in Subsection 2.2 with all three spatial dimensions $x, y, z$ to simulate Lamb wave propagating through a three-dimensional domain. The type of Lamb wave selected is pure symmetrical mode S0 excited with force configuration as shown in Figure 11 applied on the middle of the plate. Property and parameters of the model are listed in Table 5. Material distribution for the case study is represented in Figure 21 alongside with sensors' position in Figure 22, which are equally distributed on the plate surface, the middle sensor is excluded due to the presence of the forcing term that generated considerably bigger displacement then the other sensors and then can impair the process of optimization. Sensors positioned on the edge of the plates are not useful considering that the plate is constrained and thus no displacement is allowed on that surface rendering sensors useless. Note that similarly to the two-dimensional algorithm, material distribution is constant along thickness direction $z$.

**Plate parameters**

| | |
|---|---|
| **Elastic modulus** $[GPa]$ | $E = 70$ |
| **Poisson ratio** | $\nu = 0.33$ |
| **Density** $[Kg/m^3]$ | $\rho = 2700$ |
| **Length x** $[cm]$ | $L_x = 5$ |
| **Length y** $[cm]$ | $L_y = 5$ |
| **Thickness z** $[cm]$ | $L_z = 0.1$ |
| **Number of Sensors** | 24 |

Table 5: Property and parameters of the model

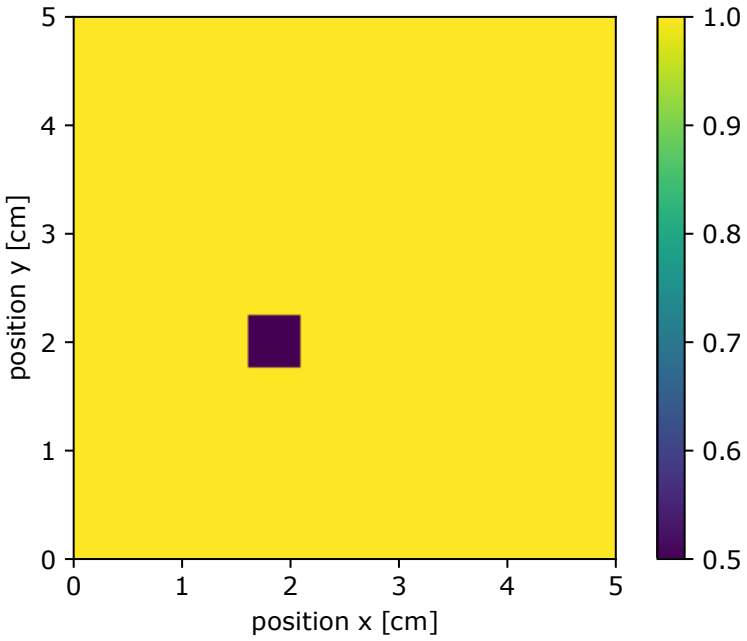


Figure 21: Material distribution of the plate

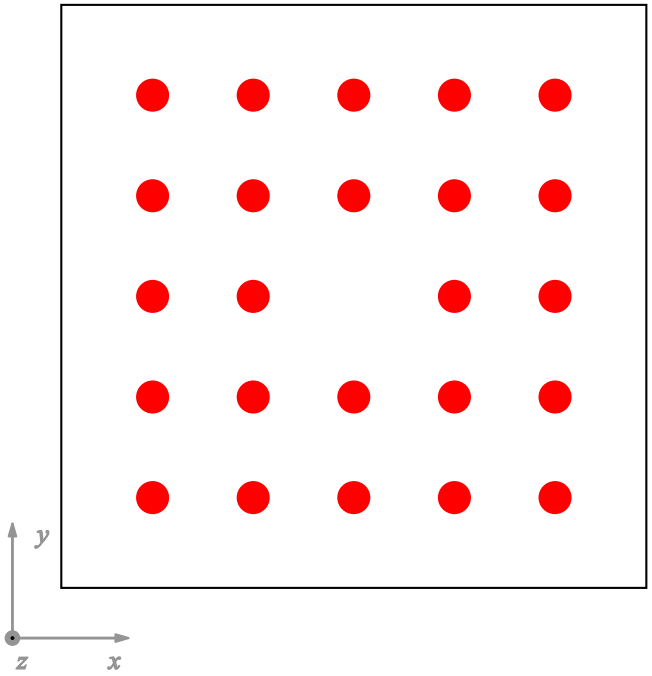


Figure 22: Sensors distribution on the plate represented by the red dots

In this scenario, the neural network remained the same FFNN used since the beginning of the paper however the input is no more one spatial coordinate $\boldsymbol{x}$ but two $\boldsymbol{x}$ and $\boldsymbol{y}$ represented in the Illustration 23, this is due to the geometry of the problem that is a plate, while the output is the material distribution parameters $\boldsymbol{\gamma}(\boldsymbol{x}, \boldsymbol{y})$, that do not depend from $\boldsymbol{z}$ coordinate that is constant in that direction. The architecture is then two neurons at the input layer, the hidden layers have the following number of neurons 5,10,10,5 and finally, the output layer has one neuron without any activation function.

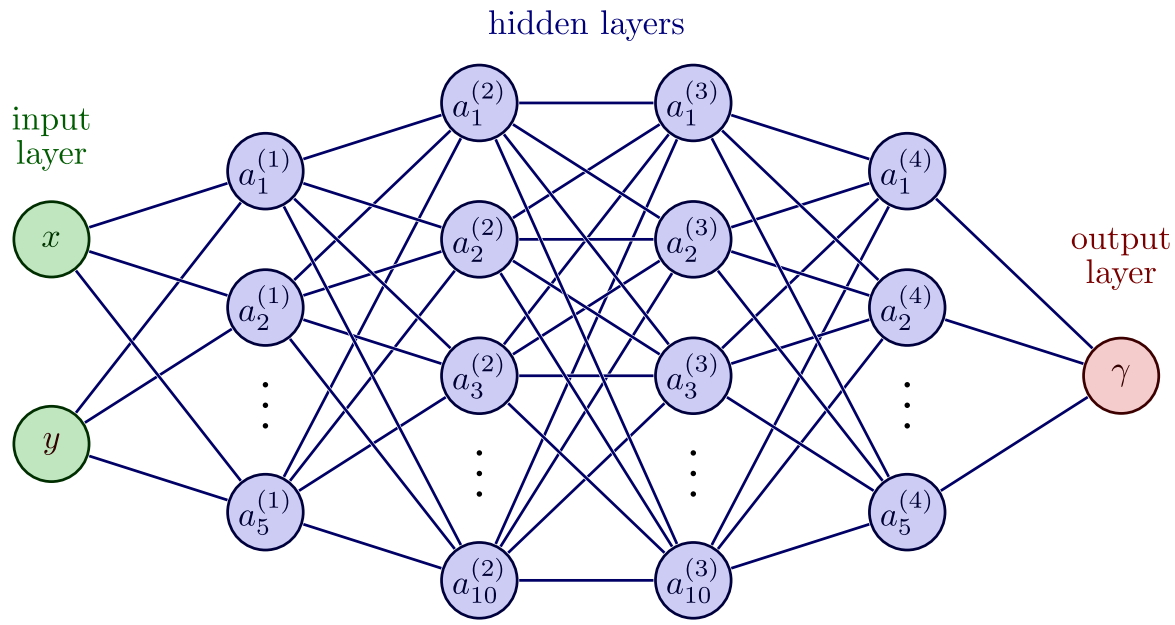


Figure 23: Neural network scheme

The addition of one dimension greatly increases the complexity of the optimization process with respect to the previous cases due to the increased size of the domain. As a consequence training process needs around 25000 epochs, and computational cost is very high, nevertheless, there are possible solutions that can improve drastically the performance of the algorithm that are discussed in Section 4.

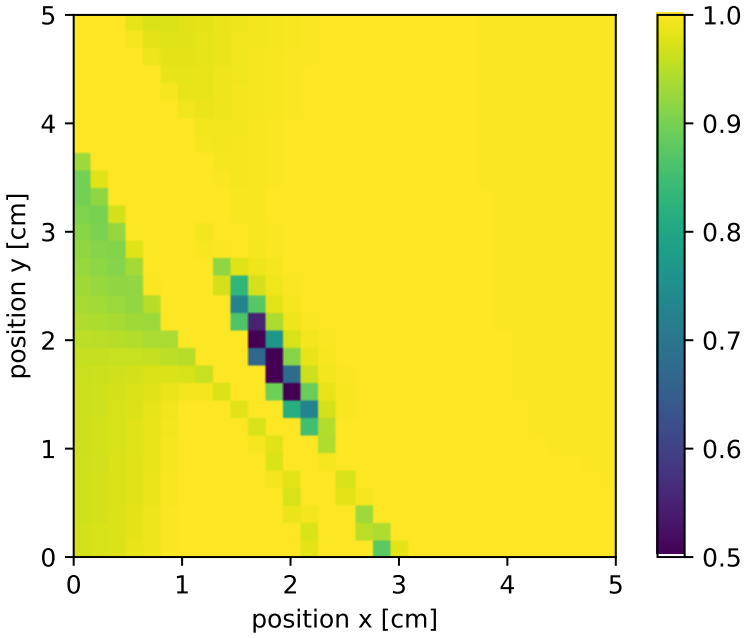


Figure 24: Neural network material distribution prediction after training

Once performed the training of the net illustrated in Figure 23, using the Physics-informed method coupled with finite difference solver, the approximation of material properties $\boldsymbol{\gamma}(\boldsymbol{x}, \boldsymbol{y})$ across the domain is presented in Figure 24. The application of this method on the plate shows less precise prediction than the previous cases but is still capable of indicating correctly the region of the damage which is a great achievement considering the only information given to the model is the displacement measured at the sensors positions.

# 4. Conclusions

In this work, two physics-informed data-driven methods that process Lamb waves to localise damage in thin-walled structures have been presented. These types of algorithms were considered to explore the possibility of performing damage diagnosis without relying on black box-like neural networks and without performing any feature extraction processes.
The two data-driven methods only differed by the way the known physical laws were implemented. The traditional physics-informed method included the physics of the problem, i.e., the underlying partial differential equations, directly in the loss function. Instead, the physics-informed method coupled with the finite difference solver was composed of a normal neural network predicting the material distribution, and the physics of the problem was satisfied by an in-house finite difference solver specifically developed to solve 1D, 2D and 3D elastodynamic simulations involving Lamb waves. Three case studies have been considered. The following main conclusions can be drawn from this work:

- 1D case: Simplified physics-informed method and Physics-informed coupled with finite difference solver method are tested with the aim to detect damage position in a string domain, both methods are capable of detecting damages on the domain. However, the first method is not able to quantify correctly the material property in the damaged region as well as the position due to the low resolution. Another drawback is the numerical stability problem developed with the reduction of the number of sensors needing a reduction in the frequency of the wave, the second method overcomes all these limitations however increasing the computational.
- 2D case: only the most promising method is tested, that is the Physics-informed coupled with finite difference solver method, the domain selected is a plate under plane strain in one dimension and thus problem becomes two dimensional, that is a plate section. The method tested is capable of detecting damage position and evaluating the wave speed change in the damaged area.
- 3D case: the method used in the previous case is then tested on a plate, the algorithm after the training phase is capable of detecting damage position, however, the solution is not as accurate as the previous cases. Considering the lower accuracy the method is still promising, taking into account that the only data used where the displacement was measured in sensor positions, no DIs or other processing has been done to the signals.
- The computational cost of the Physics-informed coupled with finite difference solver is very high due to the presence of the finite difference solver embedded in the algorithm. There are different solution well known in the literature that can drastically improve performance and thus render this approach less time-consuming: implementing CNN, which allows physically bound solutions and prove to be efficient in [37]. Use of gradient clippings and loss function weighting, valuable techniques to reduce convergence time. Utilizing GPU implementation of the finite difference solvers, to run the whole algorithm on a graphics card that is faster for this type of computation compared to the CPU.
- Finite difference formulation for wave equation is complex and developing the correct stress-free boundary conditions has proven to be a limitation in possible Lamb wave mode simulated. Dispersion phenomena were consequently not represented correctly in a certain configuration of the forcing term actually limiting the range of frequencies available.

As a future extension, testing the method's damage detection ability using sensor measurements from a real plate with installed piezoelectric sensors could further assess its capabilities in a real-world scenario.

# References


[1] Omar AlShorman, Muhammad Irfan, Nordin Saad, D. Zhen, Noman Haider, Adam Glowacz, and Ahmad AlShorman. A review of artificial intelligence methods for condition monitoring and fault diagnosis of rolling element bearings for induction motor. *Shock and Vibration*, 2020:1–20, 11 2020.

[2] Omar AlShorman, Fahad Alkahatni, Mahmoud Masadeh, Muhammad Irfan, Adam Glowacz, Faisal Althobiani, Jaroslaw Kozik, and Witold Glowacz. Sounds and acoustic emission-based early fault diagnosis of induction motor: A review study. *Advances in Mechanical Engineering*, 13:168781402199691, 02 2021.

[3] Adam Glowacz. Thermographic fault diagnosis of shaft of bldc motor. *Sensors*, 22:8537, 11 2022.

[4] Onur Avci, Osama Abdeljaber, Serkan Kiranyaz, Mohammed Hussein, Moncef Gabbouj, and Daniel J. Inman. A review of vibration-based damage detection in civil structures: From traditional methods to machine learning and deep learning applications. *Mechanical Systems and Signal Processing*, 147:107077, 01 2021.

[5] Yu Wang and Lei Qiu. A piezoelectric sensor network with shared signal transmission wires for structural health monitoring of aircraft smart skin. *Mechanical Systems and Signal Processing*, 141:106730, 07 2020.

[6] Cong Du, Susom Dutta, Pradeep Kurup, Tzuyang Yu, and Xingwei Wang. A review of railway infrastructure monitoring using fiber optic sensors. *Sensors and Actuators A: Physical*, 303:111728, 03 2020.

[7] Zhongqing Su and Lin Ye. *Identification of Damage Using Lamb Waves*, volume 48. Springer, 01 2009.

[8] Mira Mitra and S Gopalakrishnan. Guided wave based structural health monitoring: A review. *Smart Materials and Structures*, 25:053001, 03 2016.

[9] B C Lee and Wiesław J Staszewski. Lamb wave propagation modelling for damage detection: I. two-dimensional analysis. *Smart Materials and Structures*, 16:249–259, 01 2007.

[10] B C Lee and W J Staszewski. Modelling of lamb waves for damage detection in metallic structures: Part ii. wave interactions with damage. *Smart Materials and Structures*, 12:815–824, 09 2003.

[11] Naoki Mori, Shiro Biwa, and Takayuki Kusaka. Damage localization method for plates based on the time reversal of the mode-converted lamb waves. *Ultrasonics*, 91:19–29, 01 2019.

[12] Zhongqing Su, Lin Ye, and Ye Lu. Guided lamb waves for identification of damage in composite structures: A review. *Journal of Sound and Vibration*, 295:753–780, 08 2006.

[13] Ruiqi Guan, Ye Lu, Wenhui Duan, and Xiaoming Wang. Guided waves for damage identification in pipeline structures: A review. *Structural Control and Health Monitoring*, 24:e2007, 03 2017.

[14] Alexander J. Dawson, Jennifer E. Michaels, and Thomas E. Michaels. Isolation of ultrasonic scattering by wavefield baseline subtraction. *Mechanical Systems and Signal Processing*, 70-71:891–903, 03 2016.

[15] Xiaoliang Zhao, Huidong Gao, Guangfan Zhang, Bulent Ayhan, Fei Yan, Chiman Kwan, and Joseph L Rose. Active health monitoring of an aircraft wing with embedded piezoelectric sensor/actuator network: I. defect detection, localization and growth monitoring. *Smart Materials and Structures*, 16:1208–1217, 08 2007.

[16] Huidong Gao, Yongyi Shi, and Joseph L Rose. Guided wave tomography on an aircraft wing with leave in place sensors. *AIP Conference Proceedings*, 760, 01 2005.

[17] Shi Yuan, Liang and Zhao. Recent progress on distributed structural health monitoring research at nuaa. *Journal of Intelligent Material Systems and Structures*, 19:373–386, 10 2007.

[18] Irina Trendafilova and Emil Manoach. Vibration-based damage detection in plates by using time series analysis. *Mechanical Systems and Signal Processing*, 22:1092–1106, 07 2008.

[19] Zhen Wu, Xinlin Qing, Kumar Ghosh, Vistasp Karbhar, and Fu-Kuo Chang. Health monitoring of bonded composite repair in bridge rehabilitation. *Smart Materials and Structures*, 17:045014–045014, 06 2008.

[20] Xinlin P. Qing, Hian-Leng Chan, Shawn J. Beard, and Amrita Kumar. An active diagnostic system for structural health monitoring of rocket engines. *Journal of Intelligent Material Systems and Structures*, 17:619–628, 07 2006.

[21] Jennifer E Michaels and Thomas E Michaels. An integrated strategy for detection and imaging of damage using a spatially distributed array of piezoelectric sensors. *Proceedings of SPIE*, 04 2007.

[22] Seunghee Park, Chung-Bang Yun, Yongrae Roh, and Jong-Jae Lee. Pzt-based active damage detection techniques for steel bridge components. *Smart Materials and Structures*, 15:957–966, 06 2006.

[23] Zenghua Liu, Xuwen Zhong, Tuocan Dong, Cunfu He, and Bin Wu. Delamination detection in composite plates by synthesizing time-reversed lamb waves and a modified damage imaging algorithm based on rapid. *Structural control and health monitoring*, 24:e1919–e1919, 08 2016.

[24] Fei Gao, Yongsheng Shao, Jiadong Hua, Liang Zeng, and Jing Lin. Enhanced wavefield imaging method for impact damage detection in composite laminates via laser-generated lamb waves. *Measurement*, page 108639, 10 2020.

[25] Dong Wang, Lin Ye, Ye Lu, and Zhongqing Su. Probability of the presence of damage estimated from an active sensor network in a composite panel of multiple stiffeners. *Composites Science and Technology*, 69:2054–2063, 10 2009.

[26] Qin Xia, Yanyan Liu, Yu Lu, Shuhao Cao, Hanfei Zhang, and Shaoping Ma. A modified damage index probability imaging algorithm based on delay-and-sum imaging for synthesizing time-reversed lamb waves. *Journal of Vibroengineering*, 21:2140–2147, 12 2019.

[27] C. Sbarufatti, G. Manson, and K. Worden. A numerically-enhanced machine learning approach to damage diagnosis using a lamb wave sensing network. *Journal of Sound and Vibration*, 333:4499–4525, 09 2014.

[28] Ziemowit Dworakowski, Krzysztof Dragan, and Tadeusz Stepinski. Artificial neural network ensembles for fatigue damage detection in aircraft. *Journal of Intelligent Material Systems and Structures*, 28:851–861, 07 2016.

[29] Ranting Cui, Guillermo Azuara, Francesco Lanza di Scalea, and Eduardo Barrera. Damage imaging in skin-stringer composite aircraft panel by ultrasonic-guided waves using deep learning with convolutional neural network. *Structural Health Monitoring*, pages 1–16, 06 2021.

[30] Chaojie Hu, Bin Yang, Jianjun Yan, Yanxun Xiang, Shaoping Zhou, and Fu-Zhen Xuan. Damage localization in pressure vessel by guided waves based on convolution neural network approach. *Journal of Pressure Vessel Technology*, 05 2020.

[31] Vincentius Ewald, Roger M Groves, and Rinze Benedictus. Deepshm: a deep learning approach for structural health monitoring based on guided lamb wave technique. *Data Archiving and Networked Services (DANS)*, 03 2019.

[32] Cheng-Shen Chang and Yung-Chun Lee. Ultrasonic touch sensing system based on lamb waves and convolutional neural network. *Sensors*, 20:2619, 05 2020.

[33] Shengyuan Zhang, Chun Min Li, and Wenjing Ye. Damage localization in plate-like structures using time-varying feature and one-dimensional convolutional neural network. *Mechanical Systems and Signal Processing*, 147:107107, 01 2021.

[34] Luca Lomazzi, Marco Giglio, and Francesco Cadini. Towards a deep learning-based unified approach for structural damage detection, localisation and quantification. *Engineering Applications of Artificial Intelligence*, 121:106003, 05 2023.

[35] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 02 2019.

[36] Majid Rasht-Behesht, Christian Huber, Khemraj Shukla, and George Em Karniadakis. Physics-informed neural networks (pinns) for wave propagation and full waveform inversions. *Journal of Geophysical Research: Solid Earth*, 127, 04 2022.

[37] Leon Herrmann, Tim Bürchner, Felix Dietrich, and Stefan Kollmannsberger. On the use of neural networks for full waveform inversion. *Computer Methods in Applied Mechanics and Engineering*, 415:116278, 10 2023.

[38] Hans Langtangen. Finite difference methods for wave motion, 2016.

[39] Stefano Carrino, Francesco Nicassio, Gennaro Scarselli, and Raffaele Vitolo. Finite difference model of wave motion for structural health monitoring of single lap joints. *International Journal of Solids and Structures*, 161:219–227, 04 2019.

[40] R Balasubramanyam, D Quinney, R E Challis, and C P D Todd. A finite-difference simulation of ultrasonic lamb waves in metal sheets with experimental verification. *Journal of Physics D: Applied Physics*, 29:147–155, 01 1996.

# A. Stability in finite difference solvers

To asses the stability of a finite difference solver is necessary to introduce those concepts, for sake of simplicity the discussion is for one dimensional case, nevertheless can be easily extended for higher dimensions scenarios. The first concept to be introduced is the Courant number represented in equation (35):

$$C = \frac{c\Delta t}{\Delta x} \tag{35}$$

where $c$ is the wave speed, $\Delta t$ is the dimension of the grid in time and $\Delta x$ is the dimension of the grid in space. This number has a crucial importance in finite difference method for its physical meaning that is the ratio between the space travelled by the wave in one time-step and the dimension of the grid. In this way it is easy to evaluate the consequences of this number, if the wave travel more space than the spatial grid, problem of stability will rise. From this consideration a safe condition enforced on this number is $C \leq 1$. Is useful to extend this consideration to higher dimension scenarios, in the simplest case where $\Delta x = \Delta y = \Delta z$ the time-step dimension $\Delta t$ has satisfy Equation (36) in order to guarantee stability:

$$\Delta t \leq \frac{\Delta x}{\sqrt{c_L^2 + c_T^2}} \tag{36}$$

where $c_L$ is the longitudinal wave speed and $c_T$ is the shear wave speed. Another important aspect of stability is the wave frequency that can be represented on a certain grid discretization, this condition is crucial to guarantee accurate derivative in particular for Simplified physics-informed method presented in this paper, coarse grid generated by low number of sensors avoid the computation of accurate derivative for waves with too high frequency. This condition is the mathematical proof of this behavior, represented in Equation (37):

$$\lambda = \frac{c}{f} > 2\Delta x \tag{37}$$

where $\lambda$ is the wavelength, $c$ is the wave speed and $f$ is the frequency of the wave. Equation (37) represent the shortest wave length that can be represented on a grid discretization, as a consequence waves with wavelength lower then the double of the grid discretization dimension $\Delta x$ are not represented having too small dimension.

## B. Dispersion in Lamb waves

As with most guided waves, Lamb waves are dispersive, and their velocities are dependent on wave frequency and plate thickness. This phenomenon is referred to as dispersion. A manifestation of this phenomenon is shown in this example, a wave excited at a central frequency of 300 kHz. The S0 mode peaks at 293 kHz and the A0 mode at 332 kHz in the captured signal, deviating from the original central frequency of 300 kHz. Such a shift in the central frequency from the original excitation frequency is a key manifestation of wave dispersion, that in a plate under plane strain can be modelled with following equations:

$$\frac{\tan(qh)}{q} + \frac{4k^2 p \tan(ph)}{(k^2 - q^2)^2} = 0 \quad \text{for symmetric modes} \tag{38}$$

$$q\tan(qh) + \frac{(k^2 - q^2)^2 \tan(ph)}{4k^2 p} = 0 \quad \text{for anti-symmetric modes} \tag{39}$$

where $h$ is the half-thickness of the plate, $\omega$ is the circular frequency and $c_L$, $c_T$ and $c_P$ represent the longitudinal, transverse and phase wave velocity, respectively. Moreover, $p^2$, $q^2$ and the angular wavenumber $k$ are variables related to the circular frequency and to the wave velocities.

$$p^2 = \frac{\omega^2}{c_L^2} - k^2, \; q^2 = \frac{\omega^2}{c_T^2} - k^2, \; k = \frac{\omega}{c_P} \tag{40}$$

The equations above are the dispersion equations of Lamb waves (38) and (39), the graphic depiction of solutions of the dispersion equations is called the dispersion curves shown in Figure 25. Dispersion curves are used to describe and predict the relationship among frequency, phase/group velocity and thickness [7]. In our case study is crucial to note that there is a region with particular frequency of the signal and thickness of the plate were the wave express a reduced dispersion. This region is the one highlighted in Figure 25 with a dotted rectangle, and is the one were the finite difference solver perform better in term of wave dispersion obtained. The values for the product thickness and frequency typical of the cases presented in this paper range between 200 $Hz \cdot m$ and 400 $Hz \cdot m$, values that are well inside the region explained above.

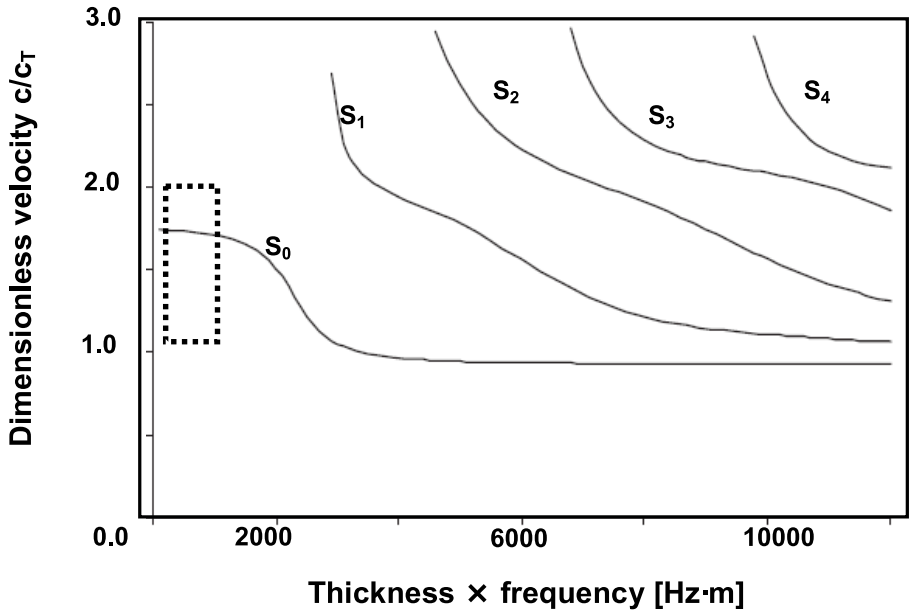


Figure 25: Dispersion curve for anti-symmetric modes

# Abstract in lingua italiana

Le Lamb waves sono ampiamente utilizzate per valutare i danni strutturali a causa della loro sensibilità ai difetti. Nonostante la facilità di eccitazione e acquisizione, spesso è necessario elaborare i segnali per ottenere indicatori univoci, noti come indici di danneggiamento. Tradizionalmente, gli indici di danneggiamento sono stati sviluppati utilizzando algoritmi tomografici per creare mappe di probabilità di danneggiamento, anche se questo approccio è soggetto a limitazioni. Recentemente, l'applicazione del machine learning è stata impiegata per migliorare l'accuratezza dei modelli che usano Lamb waves per la diagnosi dei danni. Tuttavia, molti metodi esistenti richiedono ancora l'estrazione degli indici di danneggiamento dai segnali acquisiti, potenzialmente portando a una perdita di informazioni diagnostiche e a una diminuzione dell'accuratezza. Un nuovo approccio appartenente al machine learning è recentemente emerso in popolarità per la sua flessibilità e spiegabilità: questa categoria di modelli è chiamata physics-informed neural networks. Queste reti permettono di incorporare nell'algoritmo alcune leggi fisiche conosciute, per assicurarsi che le previsioni aderiscano alla fisica del problema. Tuttavia, non si trovano molte applicazioni nel campo della diagnosi dei danni. In questo contesto, il presente lavoro si propone di presentare un framework informato dalla fisica per eseguire la diagnosi dei danni utilizzando le Lamb waves evitando l'estrazione degli indici di danneggiamento. Per valutare le prestazioni del framework proposto sono stati presi in considerazione diversi casi studio.

**Parole chiave: PINN, Machine Learning, Rete Neurale, SHM**

# Acknowledgements

I would like to express my deep gratitude to all the people who contributed to the completion of this thesis. First of all, I would like to thank my thesis advisor, Prof. Cadini, for his constant support in guiding me along a research path that was not without its difficulties. His expertise and availability made the realisation of this work possible. Special thanks go to Dr. Lomazzi and Ing. Pinello, who provided me with valuable ideas and suggestions to continue on the right path. The countless meetings and discussions were instrumental in enabling me to complete this work. I would also like to thank my parents, Massimo and Emanuela, for their constant encouragement and support throughout the academic journey, and I would like to thank my brother Riccardo and sister Carola for brightening my days in their company. I would not like to forget all my friends who have made the years of this journey fantastic. Their presence has made this journey meaningful and less demanding. Finally, I would like to express my gratitude to all the people who, directly or indirectly, contributed to this achievement. This academic journey has certainly been challenging but very formative both personally and professionally, the things I have learnt throughout these five years form the basis on which I hope to continue learning and improving as a man in the years to come.

# Ringraziamenti

Desidero esprimere la mia profonda gratitudine a tutte le persone che hanno contribuito al completamento di questa tesi. Innanzitutto, vorrei ringraziare il mio relatore, Prof. Cadini, per il suo sostegno costante nel guidarmi lungo il percorso di ricerca non privo di difficoltà. Le sue competenze e la sua disponibilità hanno reso possibile la realizzazione di questo lavoro. Un ringraziamento speciale va a Dr. Lomazzi e Ing. Pinello, che mi hanno fornito preziose idee e suggerimenti per continua sulla strada corretta. Gli innumerevoli incontri e discussioni sono stati fondamentali per permettermi il completamento di questo lavoro. Desidero anche ringraziare i miei genitori Massimo ed Emanuela, per il loro incoraggiamento costante e il supporto durante tutto il percorso accademico, vorrei ringraziare mio fratello Riccardo e mia sorella Carola per aver alleggerito le mie giornate in loro compagnia. Non vorrei dimenticare tutti i miei amici che hanno reso gli anni di questo percorso fantastici. La loro presenza ha reso questo viaggio significativo e meno impegnativo. Infine, voglio esprimere la mia gratitudine a tutti le persone che, direttamente o indirettamente, hanno contribuito al mio percorso accademico. Questo percorso accademico è stato sicuramente impegnativo ma molto formante sia a livello personale che professionale, le cose imparate lungo questi cinque anni costituiscono la base su cui mi auguro di continuare ad imparare e migliorare come uomo negli anni a venire.