C. R. Mecanique 344 (2016) 113–118

Contents lists available at ScienceDirect

Comptes Rendus Mecanique

www.sciencedirect.com

On the neural network calculation of the Lamé coefficients through eigenvalues of the elasticity operator

Sebastián Ossandón [a,*], Camilo Reyes [b]

[a] Instituto de Matemáticas, Pontificia Universidad Católica de Valparaíso, Blanco Viel 596, CerroBarón, Valparaíso, Chile
[b] Departamento de Ciencias Fisicas, Universidad Andres Bello, Avenida Republica 220, Santiago, Chile

ARTICLE INFO

Article history:
Received 23 June 2015
Accepted 22 October 2015
Available online 30 December 2015

Keywords:
Artificial neural network
Radial basis function
Lamé coefficients
Inverse problems
Eigenvalues of the elasticity operator
Finite-element method

ABSTRACT

A new numerical method is presented with the purpose to calculate the Lamé coefficients, associated with an elastic material, through eigenvalues of the elasticity operator. The finite element method is used to solve repeatedly, using different Lamé coefficients values, the direct problem by training a direct radial basis neural network. A map of eigenvalues, as a function of the Lamé constants, is then obtained. This relationship is later inverted and refined by training an inverse radial basis neural network, allowing calculation of mentioned coefficients. A numerical example is presented to prove the effectiveness of this novel method.

1. Introduction

In many practical problems of mechanical engineering, knowledge of material properties is one of the key elements in the design of safety systems or quality control. However, the measurement of material properties or quality control through non-destructive methods is a difficult task, since many materials used are anisotropic, composite or multilayer ones. In addition, many of these materials, especially the ones designed for safety reasons, must operate under, sometimes, extreme mechanical stress. A method for measuring and evaluating these properties is then necessary.

The inverse problems that arise in the context of elasticity are usually motivated by the need to have information concerning the properties and parameters of the materials under study. For example, we can mention:

- mathematical and computational methods for the reconstruction of cavities, cracks or inclusions (see [1,2]);
- ultrasonic waves for non-destructive testing of structures (see [3]);
- identification of model parameters such as Lamé coefficients, elastic moduli, mass density or wave velocity (see [4]);
- reconstruction of residual stresses (see [5]);
- model updating when local parameters are not known with sufficient accuracy, and therefore need to be corrected (usually with experimental information) on the dynamic response of the structure [6].

This article is devoted to the identification of model parameters. Specifically the paper proposes a method based on artificial neural networks to calculate the Lamé coefficients through eigenvalues of the elasticity operator. The applicability of this technique to solve real inverse problems depends on the measure, in practice, of the eigenvalues (or resonances)

* Corresponding author.
E-mail address: sebastian.ossandon@pucv.cl (S. Ossandón).

http://dx.doi.org/10.1016/j.crme.2015.10.004

associated with the elastic solid under study. Experimentally, it is possible to obtain both eigenvalues and model parameters using devices that use piezoelectric transducers. The operation of these devices is based on resonance methods such as resonant ultrasound spectroscopy (RUS) (see [7]). In a resonance experiment, we apply a periodical excitation (typically a sinusoidal excitation) to some point on the material, measure its response at some other point, and repeat the process for many frequencies. In typical RUS measurements, our purpose is to measure all of the resonances below some upper limit, because with a complete set of resonances we can assure the extraction of all the available information, which significantly simplifies the calculation process.

The Artificial Neural Network (ANN) proposed is a multilayered Radial-Basis Function (RBF) network (see Girosi et al. [8]). As discussed in Schilling et al. [9], a RBF ANN can approximate a function f using nonlinear functions that provides the best fit to the training data. Our aim is to evaluate the speed and accuracy of our neural network methodology in comparison with a method based on FEM, for a known operator whose eigenvalues can be obtained through more classical numerical methods. In other words, our purpose is to note that all the computation process using neural networks, including the training process, the validation process and the simulation process, needs less computational time than the FEM technique, with a good calculated error performance.

2. Eigenvalue problem in elasticity

Let $\Omega \subset \mathbb{R}^k$ be a nonempty, open, connected and bounded domain, with a Lipschitz-continuous boundary $\Gamma := \partial\Omega$. The unit normal vector into the exterior of Ω is denoted by $\boldsymbol{n} = (n_1, n_2, \ldots, n_k)^T \in \mathbb{R}^k$ and $\boldsymbol{x} = (x_1, x_2, \ldots, x_k)^T \in \mathbb{R}^k$.

2.1. Constitutive relations

Let us denote the fourth-rank elasticity tensor by $\boldsymbol{C}_{ijrl}$, with indices $\{i, j, r, l\}$ running from 1 to k. From thermodynamical energy considerations, the coefficients above have the following symmetry properties $\boldsymbol{C}_{ijrl} = \boldsymbol{C}_{rlij} = \boldsymbol{C}_{jirl}$. Furthermore, the elasticity tensor is positive definite. With this, the constitutive physic law relating the mechanical displacements $\boldsymbol{u}$ and the associated stresses $\boldsymbol{\sigma}_{ij}$ (known as the generalized Hooke's law) is given by

$$\boldsymbol{\sigma}_{ij}(\boldsymbol{u}) = \boldsymbol{\sigma}_{ji}(\boldsymbol{u}) = \sum_{r,l} \boldsymbol{C}_{ijrl}\boldsymbol{S}_{rl}(\boldsymbol{u}) \tag{1}$$

where we have introduced the strain tensor $\boldsymbol{S}_{ij}(\boldsymbol{u}) = \frac{1}{2}\left(\partial_j u_i + \partial_i u_j\right)$.

Let us notice that for an isotropic medium, the most general form for a fourth-rank elasticity tensor, independent of any rotation, is $\boldsymbol{C}_{ijrl} = \lambda\delta_{ij}\delta_{rl} + \mu(\delta_{jr}\delta_{il} + \delta_{il}\delta_{ir})$, where the constants λ and μ are known as Lamé constants, and $\delta_{\cdot,\cdot}$ is the Kronecker delta function.

2.2. Eigenvalue problem

Let $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}) = (u_1(\boldsymbol{x}), u_2(\boldsymbol{x}), \ldots, u_k(\boldsymbol{x})) \in \mathbb{C}^k$ be the mechanical displacements field and $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{u}) = \{\sigma_{ij}(\boldsymbol{u}(\boldsymbol{x}))\}_{1\le i,j\le k} \in \mathbb{C}^k \times \mathbb{C}^k$ the stress tensor associated. Let us consider on the boundary Γ the following Dirichlet condition $\boldsymbol{u} = \boldsymbol{0}$.

Thus, the elasticity eigenvalue problem (in the static case) can be written as:

$$-\nabla\cdot\boldsymbol{\sigma} = \gamma\boldsymbol{u} \qquad \text{for} \quad \boldsymbol{x}\in\Omega \tag{2a}$$

$$\boldsymbol{u} = \boldsymbol{0} \qquad \text{for} \quad \boldsymbol{x}\in\Gamma \tag{2b}$$

In an isotropic (and homogeneous) medium, we have:

$$\nabla\cdot\boldsymbol{\sigma} = \mu\Delta\boldsymbol{u} + (\lambda+\mu)\nabla\nabla\cdot\boldsymbol{u} \tag{3}$$

Using (2) and (3), we have the following eigenvalue system equation:

$$-\mu\Delta\boldsymbol{u} - (\lambda+\mu)\nabla\nabla\cdot\boldsymbol{u} = \gamma\boldsymbol{u} \qquad \text{for} \quad \boldsymbol{x}\in\Omega \tag{4a}$$

$$\boldsymbol{u} = \boldsymbol{0} \qquad \text{for} \quad \boldsymbol{x}\in\Gamma \tag{4b}$$

3. The direct and inverse problems

3.1. The direct problem

From now $k = 2$, thus $\Omega \subset \mathbb{R}^2$. Our purpose is to solve the following eigenvalue problem: find $\gamma \in \mathbb{R}$ and the non-null valued functions $\boldsymbol{u}$ that are solutions to

$$\begin{cases} -\mu\Delta\boldsymbol{u} - (\lambda+\mu)\nabla\nabla\cdot\boldsymbol{u} = \gamma\,\boldsymbol{u} & \text{in} \quad \Omega \\ \boldsymbol{u} = \boldsymbol{0} \quad \text{on} \quad \Gamma \end{cases} \tag{5}$$

Let us notice (see [10]) that the only non-null solutions to equations (5) are a countable pair sequence $\{(\gamma_j, \boldsymbol{u}_j\}_{j\geqslant 1}$ of eigenvalues and eigenfunctions.

Let us define the following function $\mathcal{R}_{\overline{\Omega},N}$ associated with equation (5):

$$\mathcal{R}_{\overline{\Omega},N} : \mathbb{R}^2 \to \mathbb{R}^N, \quad \overrightarrow{\boldsymbol{\gamma}}^N := (\gamma_1, \gamma_2, \cdots, \gamma_N)^{\mathbf{T}} = \mathcal{R}_{\overline{\Omega},N}(\mu, \lambda) \tag{6}$$

Let us notice that, given the values of the Lamé coefficients $\mu > 0$ and $\lambda > 0$, $\mathcal{R}_{\overline{\Omega},N}$ $(N \in \mathbb{N})$, for each domain Ω with its regular boundary Γ, solves the direct problem associated with equation (5), calculating the first N eigenvalues of the elasticity operator.

3.2. The inverse problem

Let us consider the following inverse problem associated with (5):

Find $(\mu, \lambda) \in \mathbb{R}^2$ such that the following holds

$$\begin{cases} -\mu \Delta \boldsymbol{u}_n - (\lambda + \mu)\nabla\nabla \boldsymbol{u}_n = \gamma_n \boldsymbol{u}_n & \text{in} \quad \Omega \\ \boldsymbol{u}_n = \mathbf{0} \quad \text{on} \quad \Gamma \end{cases} \tag{7}$$

where the desired sequence $\left\{\gamma_n, \boldsymbol{u}_n\right\}_n$, with $n \in \mathbb{N}$ and $n \leq N < +\infty$, is given.

Thus, now it is possible to define the function $\mathcal{R}^{-1}_{\overline{\Omega},N}$, which is the inverse function of $\mathcal{R}_{\overline{\Omega},N}$, in order to solve the inverse problem associated with equation (7):

$$\mathcal{R}^{-1}_{\overline{\Omega},N} : \mathbb{R}^N \to \mathbb{R}^2, \quad (\mu, \lambda) = \mathcal{R}^{-1}_{\overline{\Omega},N}(\overrightarrow{\boldsymbol{\gamma}}^N) \tag{8}$$

4. Solution to the direct problem

4.1. Variational formulation

Let us define the functional space

$$\mathbf{V} = \mathbf{U} = \left\{ \boldsymbol{v} = (v_1, v_2) \in [H^1(\Omega)]^2; \quad v_i = 0 \quad \text{on} \quad \Gamma, \quad 1 \leqslant i \leqslant 2 \right\} \tag{9}$$

equipped with the norm $\|\boldsymbol{v}\|^2_{1,\Omega} = (\sum_{i=1}^{2} \|v_i\|^2_{1,\Omega})^{1/2}$.

Let us notice that the Sobolev space $[H^0(\Omega)]^2$ coincides with $[L^2(\Omega)]^2$, in which case the norm and inner product are denoted by $\|\cdot\|_{0,\Omega}$ and $(\cdot,\cdot)_{0,\Omega}$, respectively.

Since that the associated variational form of equations (4a) and (4b) introduces a “bad boundary condition”, we use the corresponding variational form of equations (2a) and (2b):

$$a_{\mu,\lambda}(\boldsymbol{u}, \boldsymbol{v}) := \int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}) : \boldsymbol{S}(\boldsymbol{v})\,\mathrm{d}x = \int_\Omega \left\{ \lambda \nabla \cdot \boldsymbol{u} \nabla \cdot \boldsymbol{v} + 2\mu \boldsymbol{S}(\boldsymbol{u}) : \boldsymbol{S}(\boldsymbol{v}) \right\} \mathrm{d}x = \gamma \int_\Omega \boldsymbol{u} \cdot \boldsymbol{v}\,\mathrm{d}x \tag{10}$$

Thus the eigenvalue problem for the elasticity system with homogeneous boundary conditions (weak formulation) is given by: *find* $(\gamma, \boldsymbol{u}) \in (\mathbb{R}, \mathbf{U})$ *such that*

$$a_{\mu,\lambda}(\boldsymbol{u}, \boldsymbol{v}) = \gamma\,(\boldsymbol{u}, \boldsymbol{v})_{0,\Omega} \qquad \forall \boldsymbol{v} \in \mathbf{V} \tag{11}$$

4.2. Discretization

The wellposedness of the discrete weak form of (11) can be guaranteed by the fact that the corresponding approximation spaces satisfy the Babuska–Brezzi condition (see [10–15]). Let $\{\mathcal{T}_h\}_{h>0}$ be a regular family of triangulations of Ω, made up of triangles T of diameter h_T, such that $h := \sup_{T \in \mathcal{T}_h} h_T$ and $\overline{\Omega} = \bigcup \{T : T \in \mathcal{T}_h\}$. Let us select, associated with the mesh $\mathcal{T}_h$, the finite element space $\mathbf{V}_h \subset \mathbf{V}$ of piecewise polynomials $\mathbb{P}_k$ of degree k, with $k \geq 1$.

Let $(\gamma_h, \boldsymbol{u}_h) \in (\mathbf{V}_h, \mathbb{R})$ be the eigenpair solution to the discrete weak form of (11). It is well known that the Rayleigh quotient for each eigenvalue γ_h is given by:

$$\gamma_h = \frac{a_{\mu,\lambda}(\boldsymbol{u}_h, \boldsymbol{u}_h)}{(\boldsymbol{u}_h, \boldsymbol{u}_h)_{0,\Omega}} = \mathcal{R}_{\overline{\Omega},N}(\mu, \lambda) \tag{12}$$

5. Solution to the inverse problem

Let us consider a direct RBF ANN (see Schilling et al. [9]) $\widehat{\mathcal{R}}^{\boldsymbol{\theta}_1}_{\overline{\Omega},N} : \mathbb{R}^2 \to \mathbb{R}^N$ as an approximation of the function $\mathcal{R}_{\overline{\Omega},N}$, with one hidden layer containing s_1 neurons and one output layer containing N neurons. Let us notice that the activation function associated with each neuron is characterized by $y = \exp\{-x^2\}$.

The function $\widehat{\mathcal{R}}^{\boldsymbol{\theta}_1}_{\overline{\Omega},N}$ has the following form:

$$\widehat{\overrightarrow{\boldsymbol{\gamma}}}^N = \widehat{\mathcal{R}}^{\boldsymbol{\theta}_1}_{\overline{\Omega},N}(\mu,\lambda) = \mathcal{L}^1_W \cdot \exp(-\boldsymbol{y}_1(\mu,\lambda) \cdot * \boldsymbol{y}_1(\mu,\lambda)) + \boldsymbol{b}^1_2 \tag{13}$$

where $\widehat{\overrightarrow{\boldsymbol{\gamma}}}^N := (\widehat{\gamma}_1, \widehat{\gamma}_2, \cdots, \widehat{\gamma}_N)^{\mathbf{T}}$ is the output vector and $\boldsymbol{y}_1(\mu,\lambda) = (\mathcal{I}^1_W \cdot (\mu,\lambda)^T) \cdot * \boldsymbol{b}^1_1$. Furthermore, $\boldsymbol{\theta}_1$ is a vector containing all the weights associated with the neural network, which must be determined in the training of the network. In other words, $\boldsymbol{\theta}_1$ contains all coefficients associated with the design parameters $\mathcal{L}^1_W$ $(N \times s_1)$, $\mathcal{I}^1_W$ $(s_1 \times 2)$, $\boldsymbol{b}^1_1$ $(s_1 \times 1)$ and $\boldsymbol{b}^1_2$ $(N \times 1)$. It is important to remark that "$\cdot$" is the classic matrix vector product, and "$\cdot *$" is the component to component vectorial product.

Let us consider a training set with $N^{(1)}_t$ input–output vectors $\left\{(\mu^{(i)},\lambda^{(i)}), (\overrightarrow{\boldsymbol{\gamma}}^N)^{(i)}\right\}_{i=1}^{N^{(1)}_t}$, where $(\overrightarrow{\boldsymbol{\gamma}}^N)^{(i)} = \mathcal{R}_{\overline{\Omega},N}(\mu^{(i)},\lambda^{(i)})$ and let us define the following optimization problem:

$$\widehat{\boldsymbol{\theta}_1} = \inf_{\boldsymbol{\theta}_1} J_{N^{(1)}_t}(\boldsymbol{\theta}_1) = \inf_{\boldsymbol{\theta}_1}\left\{ \frac{1}{N^{(1)}_t} \sum_{i=1}^{N^{(1)}_t} \left((\overrightarrow{\boldsymbol{\gamma}}^N)^{(i)} - \widehat{\mathcal{R}}^{\boldsymbol{\theta}_1}_{\overline{\Omega},N}(\mu^{(i)},\lambda^{(i)}) \right)^2 \right\} \tag{14}$$

The problem (14) can be solved iteratively using the backpropagation algorithm.

Once determined, the optimal value for $\boldsymbol{\theta}_1$, i.e. the determined $\widehat{\boldsymbol{\theta}_1}$, it is possible to consider a inverse RBF ANN $\widehat{\mathcal{R}}^{\boldsymbol{\theta}_2}_{\overline{\Omega},N} : \mathbb{R}^N \to \mathbb{R}^2$, trained with simulated data obtained from the direct network, to calculate the inverse of equation (13), in order to obtain an approximation for $\mathcal{R}^{-1}_{\overline{\Omega},N}$, as follows:

$$\begin{cases} (\widehat{\mu},\widehat{\lambda}) = \widehat{\mathcal{R}}^{\boldsymbol{\theta}_2}_{\overline{\Omega},N}(\widehat{\overrightarrow{\boldsymbol{\gamma}}}^N) = \mathcal{L}^2_W \cdot \exp(-\boldsymbol{y}_2(\widehat{\overrightarrow{\boldsymbol{\gamma}}}^N) \cdot * \boldsymbol{y}_2(\widehat{\overrightarrow{\boldsymbol{\gamma}}}^N)) + \boldsymbol{b}^2_2 \\ \boldsymbol{y}_2(\widehat{\overrightarrow{\boldsymbol{\gamma}}}^N) = (\mathcal{I}^2_W \cdot \widehat{\overrightarrow{\boldsymbol{\gamma}}}^N) \cdot * \boldsymbol{b}^2_1 \end{cases} \tag{15}$$

where $\boldsymbol{\theta}_2$ is a parameter vector containing everything that is going to be determined from the network training and associated with the design parameters $\mathcal{L}^2_W$ $(2 \times s_2)$, $\mathcal{I}^2_W$ $(s_2 \times N)$, $\boldsymbol{b}^2_1$ $(s_2 \times 1)$ and $\boldsymbol{b}^2_2$ (2×1). Let us notice that s_2 is the number of neurons in the hidden layer.

The problem of training this inverse network also can be solved using the backpropagation algorithm with $N^{(2)}_t$ input–output vectors.

6. Numerical example

Let us consider a square domain $\Omega = (]0.0, 1.0[\times]0.0, 1.0[) \subset \mathbb{R}^2$ and the following coefficients used for training the direct ANN: $\mu^i = \dfrac{E}{2(1+\nu^i)}$ and $\lambda^i = \dfrac{E\nu^i}{(1+\nu^i)(1-2\nu^i)}$, where the constant $E = 21.5$ is the Young's modulus and $\nu^i = 0.1 + 0.01(i-1)$, with $1 \le i \le N^{(1)}_t = 21$, is Poisson's ratio.

Once trained the network $\widehat{\mathcal{R}}^{\boldsymbol{\theta}_1}_{\overline{\Omega},N}$, using the FEM technique with $\mathbb{P}_2$ elements in Ω (see Section 4), and calculated the associated vector $\widehat{\boldsymbol{\theta}_1}$, we use this direct network to simulate a more larger amount of data $N^{(2)}_t$, obtaining a set of training data for the inverse network $\widehat{\mathcal{R}}^{\boldsymbol{\theta}_2}_{\overline{\Omega},N}$. In this case, $1 \le i \le N^{(2)}_t = 201$, $\nu^i = 0.1 + 0.001(i-1)$. This last training gives us the value of $\widehat{\boldsymbol{\theta}_2}$. The algorithm used to train both networks is the backpropagation algorithm. Fig. 1 shows a comparison of the evolution of the Lamé coefficients, when $\nu^i = 0.1 + 0.0001(i-1)$ with $1 \le i \le 2001$, as a function of the first $N = 3$ eigenvalues: 1) calculated using the inverse RBF ANN directly applied to the set of eigenvalues and 2) calculated from the inverse functional relationship between Lamé coefficients and eigenvalues obtained with the FEM technique. As seen in this figure, the Lamé coefficients calculated from the neural network method approach quite well the calculated coefficients using the inverse functional relationship.

Table 1 summarizes the computational performance using the mean squared error (MSE), the computational time, in seconds, using RBF ANN (CT ANN) directly applied to the set of eigenvalues, and also the computational time, in seconds, using the inverse functional relationship between Lamé coefficients and eigenvalues obtained with the FEM technique (CT FEM), required for simulations. The computer used to obtain the above results have a 2.16-GHz Intel Core Duo processor with 1 GB 667 MHz DDR2 SDRAM.

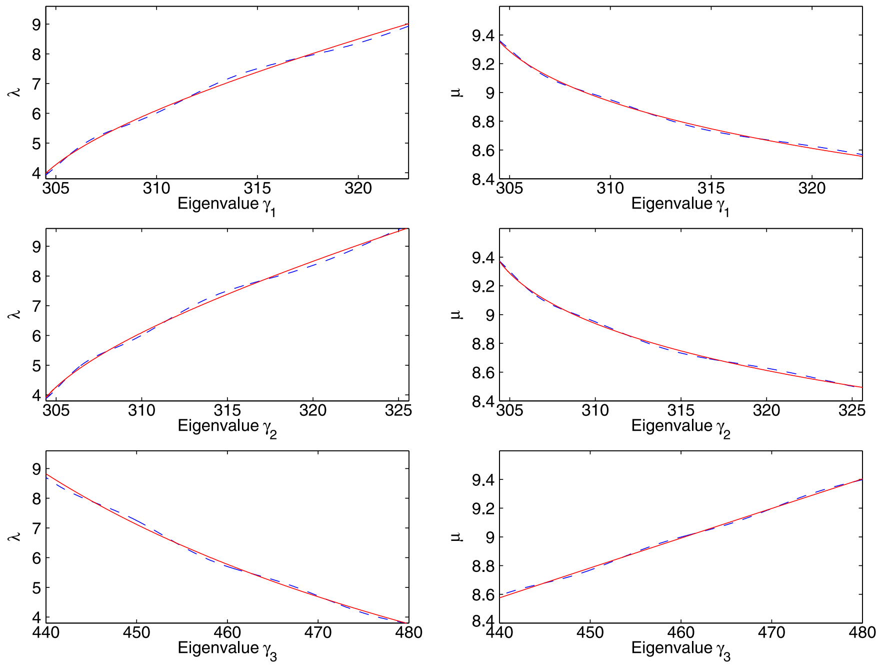

Fig. 1. Lamé coefficients as a function of the first $N = 3$ eigenvalues: 1) calculated using inverse RBF ANN (in dashed line), 2) calculated using the inverse functional relationship (in solid line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Summary for computational performance and the computational time for the numerical example using $N_s = 2001$ simulation data.

N_s	MSE λ	MSE μ	CT ANN	CT FEM
2001	4.3863e–05	1.8435e–06	14.1525	573.39

Let us notice that CT ANN is obtained taking into account the computational time required to calculate the training data, through FEM, needed by the first network in each example. We observe from the above table the excellent computational time obtained by using the RBF ANN compared with the computational time obtained by using the FEM procedure, remarking the also good computational performance that is measured using the MSE.

Finally, let us remark that a more complex geometry of the domain will be necessary to train with more data $N_t^{(1)}$ the direct RBF ANN with the purpose of improving the MSE.

7. Conclusion

In this paper, an efficient numerical method, based on an artificial neural network is presented, in order to calculate approximately the Lamé coefficients associated with an elastic solid, using the eigenvalues of the linear elasticity operator. The results show that the calculation of the Lamé coefficients, associated with the elastic properties of the material under study, through the eigenvalues computed using a neural network method, is very efficient. In other words, the relative error is negligible, and the computation time is significantly smaller than the used FEM technique, as seen in Table 1.

A neural-network-based method has shown that it can be used as an approximate method for calculating the elastic features of solids. In summary, the main advantage of this method is that all the computation process using neural networks, including the training process, the validation process and the simulation process, needs less computational time than the FEM technique, with a good calculated error performance.

Acknowledgement

Camilo Reyes thanks the partial support from CSIRO-CHILE 10CEII-9007 International Center of Excellence (Innova-Chile CORFO); Program 3-Project 1.

References

[1] C.J.S. Alves, T. Ha Duong, Inverse scattering for elastic plane cracks, Inverse Probl. 15 (1999) 91–97.

[2] S. Andrieux, A. Ben Abda, H.D. Bui, Sur l'identification de fissures planes via le concept d'écart à la réciprocité en élasticité, C. R. Acad. Sci. Paris, Ser. II 324 (1997) 1431–1438.
[3] K.R. Leonard, E.V. Malyarenko, M.K. Hinders, Ultrasonic Lamb wave tomography, Inverse Probl. 18 (2002) 1795–1808.
[4] H. Ammari, H. Kang, G. Nakamura, K. Tanuma, Complete asymptotic expansions of solutions of the system of elastostatics in the presence of an inclusion of small diameter and detection of an inclusion, J. Elast. 67 (2002) 97–129.
[5] P. Ballard, A. Constantinescu, On the inversion of subsurface residual stresses from surface stress measurements, J. Mech. Phys. Solids 42 (1994) 1767–1788.
[6] J. Ben Abdallah, Inversion gaussienne appliquée à la correction paramétrique de modèles structuraux, Ph.D. thesis, École polytechnique, Paris, France, 1995.
[7] B.J. Zadler, Properties of elastic materials using contacting and non-contacting acoustic spectroscopy, Ph.D. thesis, Colorado School of Mines, Golden, Colorado, USA, 2005.
[8] F. Girosi, M. Jones, T. Poggio, Regularization theory and neural networks architectures, J. Neural Comput. 7 (1995) 219–269.
[9] R.J. Schilling, J.J. Carroll Jr., A.F. Al-Ajlouni, Approximation of nonlinear systems with radial basis function neural networks, IEEE Trans. Neural Netw. 12 (1) (2001) 1–15.
[10] I. Babuska, J. Osborn, Eigenvalue problems, in: P.G. Lions (Ed.), Finite Element Methods (Part 1), in: Handbook of Numerical Analysis, vol. II, North-Holland, Amsterdam, 1991, pp. 641–787.
[11] D. Boffi, Finite element approximation of eigenvalue problems, Acta Numer. 19 (2010) 1–120.
[12] D. Boffi, F. Brezzi, M. Fortin, Mixed Finite Element Methods and Applications, Springer Series in Computational Mathematics, vol. 44, Springer, Heidelberg, Germany, 2013.
[13] F. Brezzi, M. Fortin, Mixed and Hybrid Finite Element Methods, Springer Series in Computational Mathematics, vol. 15, Springer-Verlag, 1991.
[14] P.G. Ciarlet, The Finite Element Method for Elliptic Problems, North-Holland, Amsterdam, 1978.
[15] B. Mercier, J. Osborn, J. Rappaz, P.A. Raviart, Eigenvalue approximation by mixed and hybrid methods, Math. Comput. 36 (1981) 427–453.