C. R. Mecanique 346 (2018) 39–47

Contents lists available at ScienceDirect

Comptes Rendus Mecanique

www.sciencedirect.com

Neural network solution to an inverse problem associated with the eigenvalues of the Stokes operator

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

[a] Instituto de Matemáticas, Pontificia Universidad Católica de Valparaíso, Blanco Viel 596, CerroBarón, Valparaíso, Chile
[b] Facultad de Ingeniería, Ciencia y Tecnología, Universidad Bernardo O'Higgins, Avenida Viel 1497, Santiago, Chile

ARTICLE INFO

Article history:
Received 20 October 2017
Accepted 14 November 2017
Available online 1 December 2017

Keywords:
Artificial neural network
Radial basis function
Viscosity and density coefficients
Inverse problems
Eigenvalues of the Stokes operator
Finite element method

ABSTRACT

A numerical method, based on the design of two artificial neural networks, is presented in order to approximate the viscosity and density features of fluids from the eigenvalues of the Stokes operator. The finite element method is used to solve the direct problem by training a first artificial neural network. A nonlinear map of eigenvalues of the Stokes operator as a function of the viscosity and density of the fluid under study is then obtained. This relationship is later inverted and refined by training a second artificial neural network, solving the aforementioned inverse problem. Numerical examples are presented in order to show the effectiveness and the limitations of this methodology.

1. Introduction

The Stokes and Navier–Stokes problems appear in many areas of engineering, physics, computational sciences, and applied mathematics; justifying a wide variety of studies. In the case of viscous fluid with a reduced velocity on simple regions, the Stokes model is a valid choice for studying these phenomena. Furthermore, when we solve the incompressible Navier–Stokes equations by Newton's method, the solution to Stokes or Stokes-like problems is required in the associated nonlinear iterations, being often the most computationally expensive part of the numerical procedures. As examples of works that study the numerical solutions to the Navier–Stokes equation, we can mention Chorin [1] and Tomasset [2].

The practical interest in Stokes problems regarding the eigenvalues and eigenfunctions has a long history in structural dynamics where the eigenvalues of a linear structural system correspond to the squares of the natural frequencies (see Hughes [3]). Also, it can be used to analyze diffusive problems, where the eigenvalues of the differential system correspond to the dissipation rates of distinct eigenfunctions (see Bazilevs et al. [4]). Another problem that can be analyzed through the Stokes equation is the study of a plate buckling problem, when it is subject to clamped boundary conditions admitting an equivalent formulation in terms of a Stokes problem (see [4], [5], [6] and [7]). In general, the calculation of eigenvalues (and eigenfunctions), associated with the Stokes operator in a bounded domain, is a fundamental area of study in fluid mechanics. For example, this knowledge can provide some analysis on turbulent instantaneous flow field (see [8]). On the other hand, the features of a fluid, such as the viscosity and the density, can be used as important manipulated variables, in order to control the dynamics of the fluid under study.

* Corresponding author.
E-mail addresses: sebastian.ossandon@pucv.cl (S. Ossandón), mauricio.barrientos@pucv.cl (M. Barrientos), creyesm70@gmail.com (C. Reyes).

https://doi.org/10.1016/j.crme.2017.11.006
1631-0721/

The use of neural networks to solve partial differential equations is not new, for example in Baymani et al. ([9] and [10]) the use of neural networks, in order to solve the Stokes and Navier–Stokes direct problems, is discussed, showing through the results that the neural network has a higher accuracy than classical methods. In Girosi et al. [11], the regularization problem for the neural networks is discussed and analyzed in detail. More recently, in Ossandón and Reyes [12] and Ossandón et al. [13], the authors solve, respectively, inverse eigenvalue problems for the linear elasticity operator and for the anisotropic Laplace operator. Also, in Ossandón et al. [14], the authors solve an inverse problem, using a neural network approach, in order to calculate the potential coefficients associated with the Hamiltonian operator in quantum mechanics.

In this work, we are interested in solving an inverse eigenvalue problem for the Stokes equation using an Artificial Neural Network (ANN) methodology. In other words, we are interested into obtaining the viscosity (ν) and density (ρ) for the fluid under study, based on the design of two ANNs (direct and inverse ANNs), as a function of eigenvalues of the Stokes operator. The proposed ANNs are multilayered Radial-Basis Function (RBF) networks. The RBF ANN is chosen due to the nature of the problem that is analyzed and the features exhibited by the neural network. As discussed in Schilling et al. [15], a RBF ANN can approximate a function f using nonlinear functions, which provides an optimal fit to the training data. The design of a RBF ANN in its most basic form consists in three layers: input, hidden, and output. Through a backpropagation algorithm, the parameters (weights) of the network are optimized in order to fit the input–output data. Let us mention that our final purpose is to evaluate the effectiveness (speed and accuracy) of the ANN methodology in comparison with other techniques, for a known operator whose eigenvalues can be obtained through more classical numerical methods.

The article is organized as follows: in Section 2, the direct and inverse problems associated with the calculation of the eigenvalues of the Stokes operator are presented. In Section 3, the methodology employed, using a FEM technique, to obtain a solution for the direct problem is described. On the other hand, Section 4 shows the methodology employed, using a RBF ANN, to obtain a solution to the related inverse problem. Numerical results are given in Section 5. Finally, in Section 6, the conclusions of this work are presented.

2. The direct and inverse problems

2.1. The direct problem

In this subsection, the mathematical formulation of the time-harmonic direct problem associated with the computation of the eigenvalues of the Stokes operator is presented.

The purpose is to solve the following eigenvalue problem: find $\lambda \in \mathbb{R}$ and the non-null valued functions $(\mathbf{u}, p)$ that are solutions to

$$\begin{cases} -\dfrac{\nu}{\rho}\Delta\mathbf{u} + \dfrac{1}{\rho}\nabla p = \lambda\,\mathbf{u} & \text{in } \Omega \\ \nabla\cdot\mathbf{u} = \mathbf{0} & \text{in } \Omega \\ \mathbf{u} = \mathbf{0} & \text{on } \Gamma \end{cases} \tag{1}$$

Let us notice (see [16]) that the only non-null solutions to equations (1) are a countable eigenvalue sequence $\{\lambda_j\}_{j\geqslant 1}$, given by

$$0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_k \leq \ldots \text{such that } \lim_{k\to\infty} \lambda_k = \infty$$

and the associated eigenfunctions

$$(\mathbf{u}_1, p_1),\ (\mathbf{u}_2, p_2), \ldots, (\mathbf{u}_k, p_k), \ldots$$

Let us define the following function $\mathcal{F}_{\Omega,\Gamma,N}$ associated with Eq. (1):

$$\begin{aligned} &\mathcal{F}_{\Omega,\Gamma,N} : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}^N \\ &\boldsymbol{\lambda}^d := (\lambda_1, \lambda_2, \cdots, \lambda_N)^{\mathbf{T}} = \mathcal{F}_{\Omega,\Gamma,N}(\nu, \rho) \end{aligned} \tag{2}$$

In other words, given the values of $\nu, \rho \in \mathbb{R}_+$, $\mathcal{F}_{\Omega,\Gamma,N}$ ($N \in \mathbb{N}$), for each domain Ω with its regular boundary Γ, solves the direct problem associated with Eq. (1), calculating the first N eigenvalues of the Stokes operator.

2.2. The inverse problem

Let us consider the following inverse problem associated with (1):
find the values of $(\nu, \rho) \in \mathbb{R}_+ \times \mathbb{R}_+$ such that the following holds:

$$\begin{cases} -\dfrac{\nu}{\rho}\Delta\mathbf{u}_{\mathbf{n}}^{\mathbf{d}} + \dfrac{1}{\rho}\nabla p_n^d = \lambda_n^d\,\mathbf{u}_{\mathbf{n}}^{\mathbf{d}} & \text{in } \Omega \\ \nabla\cdot\mathbf{u}_{\mathbf{n}}^{\mathbf{d}} = \mathbf{0} & \text{in } \Omega \\ \mathbf{u}_{\mathbf{n}}^{\mathbf{d}} = \mathbf{0} & \text{on } \Gamma \end{cases} \tag{3}$$

where the desired sequence $\left\{\lambda_n^d, \mathbf{u}_\mathbf{n}^\mathbf{d}, p_n^d\right\}_{n=1}^N$, with $n \in \mathbb{N}$ and $n \leq N < +\infty$, is given.

Thus, now it is possible to define the function $\mathcal{F}^{-1}_{\Omega,\Gamma,N}$, which is the inverse function of $\mathcal{F}_{\Omega,\Gamma,N}$, in order to solve the inverse problem associated with Eq. (3):

$$\begin{aligned} &\mathcal{F}^{-1}_{\Omega,\Gamma,N} : \mathbb{R}^N \to \mathbb{R}_+ \times \mathbb{R}_+ \\ &(\nu, \rho) = \mathcal{F}_{\Omega,\Gamma,N}(\boldsymbol{\lambda}^d) \end{aligned} \tag{4}$$

3. Solution of the direct problem

3.1. Variational formulation

Hereafter we will use the standard notation (see [17] and [18]) for the Sobolev spaces $H^r(\Omega)$ (standard interpolation spaces for real numbers r) and their associated inner products $(\cdot,\cdot)_r$, norms $\|\cdot\|_r$ and seminorms $|\cdot|_r$ for $r \geq 0$. The Sobolev space $H^0(\Omega)$ coincides with $L^2(\Omega)$, in which case the norm and inner product are denoted by $\|\cdot\|$ and $(\cdot,\cdot)$, respectively. In addition, the subspace of $L^2(\Omega)$, denoted by $L^2_0(\Omega)$, consists of the functions on $L^2(\Omega)$ having mean value zero. We also use the vector valued functions $[H^r(\Omega)]^2$ just as in [17] and [19].

Let us consider the eigenvalue problem for the Stokes system with homogeneous boundary conditions given in the above equations (1). The corresponding weak formulation is given by: *find* $(\lambda, \mathbf{u}, p) \in (\mathbb{R}, \mathbf{V}, W)$ *such that*

$$\begin{cases} a(\mathbf{u},\mathbf{v}) - b(\mathbf{v}, p) = \lambda\,(\mathbf{u},\mathbf{v}) & \forall \mathbf{v} \in \mathbf{V} \\ b(\mathbf{u}, q) = 0 & \forall q \in W \end{cases} \tag{5}$$

where $\mathbf{V} = [H^1(\Omega)]^2$, $W = L^2_0(\Omega)$ and

$$a(\mathbf{u},\mathbf{v}) := \int_\Omega \frac{\nu}{\rho}\, \nabla\mathbf{u} \cdot \nabla\mathbf{v}\, \mathrm{d}x$$

$$b(\mathbf{v}, p) := \int_\Omega \frac{1}{\rho}\, \mathbf{div}\,\mathbf{v}\, p\, \mathrm{d}x$$

By introducing the bilinear form

$$B((\mathbf{u}, p), (\mathbf{v}, q)) := a(\mathbf{u},\mathbf{v}) - b(\mathbf{v}, p) + b(\mathbf{u}, q),$$

the problem (5) can be written in a variational form as follows: *find* $(\lambda, \mathbf{u}, p) \in (\mathbb{R}, \mathbf{V}, W)$ *such that*

$$B((\mathbf{u}, p), (\mathbf{v}, q)) = \lambda\,(\mathbf{u},\mathbf{v}) \qquad \forall (\mathbf{v}, p) \in \mathbf{V} \times W \tag{6}$$

which has a unique solution given by the properties described above. Moreover, the following Rayleigh quotient expression holds for each eigenvalue λ:

$$\lambda = \frac{a(\mathbf{u},\mathbf{u})}{(\mathbf{u},\mathbf{u})} \tag{7}$$

3.2. Discretization

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 $\overline{\Omega} = \bigcup\{T : T \in \mathcal{T}_h\}$. Associated with the mesh $\mathcal{T}_h$, we select finite elements spaces $\mathbf{V}_h \subset \mathbf{V}$ and $W_h \subset W$ of piecewise polynomials of degree k. Let us assume that the polynomial space $\mathbb{P}_k$, with $k \geq 1$, is used for the construction of $\mathbf{V}_h$, and that $\mathbb{P}_{k-1}$ is used for the construction of W_h. The two finite element spaces $\mathbf{V}_h$ and W_h are assumed to satisfy the following approximation:

$$\begin{cases} \inf\limits_{\mathbf{v}_h\in\mathbf{V}_h} \left(\|\mathbf{u}-\mathbf{v}\|_0 + h\,\|\mathbf{u}-\mathbf{v}\|_1\right) \leq C\,h^{m+1}\,\|\mathbf{u}\|_{m+1}\,, & 0 \leq m \leq k \\ \inf\limits_{q_h\in W_h} \|p-q\|_0 \leq C\,h^m\,\|p\|_m\,, & 0 \leq m \leq k \end{cases} \tag{8}$$

for any $\mathbf{u} \in [H^{m+1}(\Omega)]^2$ and $p \in H^m(\Omega)$. Since the finite element spaces are subspaces of $[H^1_0(\Omega)]^2$, the functions in $\mathbf{V}_h$ are continuous and $k \geq 1$.

Now, Let us consider the discrete Stokes eigenvalue problem: *find* $(\lambda_h, \mathbf{u}_h, p_h) \in (\mathbb{R}, \mathbf{V}_h, W_h)$ *such that*

$$\begin{cases} a(\mathbf{u}_h, \mathbf{v}_h) - b(\mathbf{v}_h, p_h) = \lambda_h (\mathbf{u}_h, \mathbf{v}_h) & \forall \mathbf{v}_h \in \mathbf{V}_h \\ b(\mathbf{u}_h, q_h) = 0 & \forall q_h \in W_h \end{cases} \tag{9}$$

If the pair of finite element spaces $\mathbf{V}_h$ and W_h satisfies the Babuska–Brezzi condition

$$\inf_{0 \neq q_h \in W_h} \sup_{0 \neq \mathbf{v}_h \in \mathbf{V}_h} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_1 \, \|q_h\|_0} \geq C > 0 \tag{10}$$

the eigenvalue approximation of λ_h and the corresponding eigenfunction approximation $(\mathbf{u}_h, p_h)$ are bounded as follows (see [17], [5], [19]):

$$|\lambda - \lambda_h| \leq C \left(\inf_{\mathbf{v}_h \in \mathbf{V}_h} \|\mathbf{u} - \mathbf{v}_h\|_1 + \inf_{q_h \in W_h} \|p - q_h\|_0 \right)^2 \tag{11}$$

$$\|\mathbf{u} - \mathbf{u}_h\|_0 + h \, \|\mathbf{u} - \mathbf{u}_h\|_1 \leq C \, h \left(\inf_{\mathbf{v}_h \in \mathbf{V}_h} \|\mathbf{u} - \mathbf{v}_h\|_1 + \inf_{q_h \in W_h} \|p - q_h\|_0 \right) \tag{12}$$

$$\|p - p_h\|_0 \leq C \left(\inf_{\mathbf{v}_h \in \mathbf{V}_h} \|\mathbf{u} - \mathbf{v}_h\|_1 + \inf_{q_h \in W_h} \|p - q_h\|_0 \right) \tag{13}$$

Finally, as explained previously, from (9) we know that the Rayleigh quotient for each eigenvalue λ_h is given by:

$$\lambda_h = \frac{a(\mathbf{u}_h, \mathbf{u}_h)}{(\mathbf{u}_h, \mathbf{u}_h)} = \mathcal{F}_{\Omega,\Gamma,N}(\nu, \rho) \tag{14}$$

Also, from [16] we know that the Stokes eigenvalue problem (9) has a finite sequence of eigenvalues $\{(\lambda_h)_j\}_{j=1}^{N}$, given by:

$$0 < (\lambda_h)_1 \leq (\lambda_h)_2 \leq \cdots \leq (\lambda_h)_k \leq \cdots \leq (\lambda_h)_N$$

and the associated eigenfunctions

$$(\mathbf{u}_h, p_h)_1, \ (\mathbf{u}_h, p_h)_2, \ldots, (\mathbf{u}_h, p_h)_k, \ldots, (\mathbf{u}_h, p_h)_N$$

4. Solution of the inverse problem using ANN

In this section, our purpose is to give an approximation of the inverse function $\mathcal{F}^{-1}_{\Omega,\Gamma,N}$ using a RBF ANN.

4.1. RBF ANN approximation

In order to obtain an approximation of the direct function $\mathcal{F}_{\Omega,\Gamma,N}$, we will consider a first RBF ANN (see Schilling et al [15]) $\widehat{\mathcal{F}}^{\boldsymbol{\theta}_1}_{\Omega,\Gamma,N} : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}^N$, with one hidden layer containing s_1 neurons and one output layer containing N neurons. Let us notice that the transfer (or activation) function associated with each neuron has the following form: $y = \exp\{-x^2\}$.

The function $\widehat{\mathcal{F}}^{\boldsymbol{\theta}_1}_{\Omega,\Gamma,N}$ is characterized by

$$\widehat{\boldsymbol{\lambda}}^d = \widehat{\mathcal{F}}^{\boldsymbol{\theta}_1}_{\Omega,\Gamma,N}(\nu, \rho) = \mathcal{L}^1_W \cdot \exp(-\boldsymbol{y}_1(\nu, \rho) \cdot * \boldsymbol{y}_1(\nu, \rho)) + \boldsymbol{b}^1_2 \tag{15}$$

where $\widehat{\boldsymbol{\lambda}}^d := (\widehat{\lambda}_1, \widehat{\lambda}_2, \cdots, \widehat{\lambda}_N)^{\mathbf{T}}$ is the output vector and $\boldsymbol{y}_1(\nu, \rho) = (\mathcal{I}^1_W \cdot (\nu, \rho)^T) \cdot * \boldsymbol{b}^1_1$. Furthermore, $\boldsymbol{\theta}_1$ is a parameter vector containing all the parametric weights of the network that must be determined from the network training. 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 $\boldsymbol{b}^1_2$ $(N \times 1)$.

In our case "$\cdot$" is the classic matrix vector product and "$\cdot *$" is the component vectorial product.

Having decided on the structure (or topology) of the network is necessary to train the network. Let us consider a training set containing $N^{(1)}_t$ input–output vectors $\{(\nu^{(i)}, \rho^{(i)})^T, (\boldsymbol{\lambda}^d)^{(i)}\}_{i=1}^{N^{(1)}_t}$, where $(\boldsymbol{\lambda}^d)^{(i)} = \mathcal{F}_{\Omega,\Gamma,N}(\{\nu^{(i)}, \rho^{(i)}\}_{i=1}^{N^{(1)}_t})$, and let us define

$$J_{N^{(1)}_t}(\boldsymbol{\theta}_1) = \frac{1}{N^{(1)}_t} \sum_{i=1}^{N^{(1)}_t} ((\boldsymbol{\lambda}^d)^{(i)} - \widehat{\mathcal{F}}^{\boldsymbol{\theta}_1}_{\Omega,\Gamma,N}(\nu^{(i)}, \rho^{(i)}))^2 \tag{16}$$

Then, an optimal estimation for $\boldsymbol{\theta}_1$ is given by

$$\widehat{\boldsymbol{\theta}_1} = \inf_{\boldsymbol{\theta}_1} J_{N^{(1)}_t}(\boldsymbol{\theta}_1) \tag{17}$$

Starting with an initial parameter vector $\boldsymbol{\theta}_1^0$, the training algorithm iteratively decreases the mean square error updating $\boldsymbol{\theta}_1$, with each iteration, as follows

$$\boldsymbol{\theta}_1^{i+1} = \boldsymbol{\theta}_1^i - \epsilon \mathbf{L} \cdot \frac{\partial J_{N_t^{(1)}}(\boldsymbol{\theta}_1^i)}{\partial \boldsymbol{\theta}_1^i} \tag{18}$$

where ϵ controls the length of the update increment and $\mathbf{L}$ is a matrix that defines, depending on the choice, the algorithm to be used (Levenberg–Marquardt, Gauss–Newton, steepest-descent or backpropagation).

Once determined, the optimal value for $\boldsymbol{\theta}_1$, i.e. the determined $\widehat{\boldsymbol{\theta}_1}$, it is possible to consider a second RBF ANN $\widehat{\mathcal{F}}^{\boldsymbol{\theta}_2}_{\Omega,\Gamma,N} : \mathbb{R}^N \to \mathbb{R}_+ \times \mathbb{R}_+$, trained with simulated data obtained from the first network, to calculate the inverse of Eq. (15), in order to obtain an approximation for $\mathcal{F}^{-1}_{\Omega,\Gamma,N}$, as follows:

$$\begin{cases} (\widehat{\nu}, \widehat{\rho})^T = \widehat{\mathcal{F}}^{\boldsymbol{\theta}_2}_{\Omega,\Gamma,N}(\widehat{\boldsymbol{\lambda}}^d) = \mathcal{L}^2_W \cdot \exp(-\boldsymbol{y}_2(\widehat{\boldsymbol{\lambda}}^d) \cdot * \boldsymbol{y}_2(\widehat{\boldsymbol{\lambda}}^d)) + \boldsymbol{b}_2^2 \\ \boldsymbol{y}_2(\widehat{\boldsymbol{\lambda}}^d) = (\mathcal{I}^2_W \cdot \widehat{\boldsymbol{\lambda}}^d) \cdot * \boldsymbol{b}_1^2 \end{cases} \tag{19}$$

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 $\mathcal{L}^2_W$ $(2 \times s_2)$, $\mathcal{I}^2_W$ $(s_2 \times N)$, $\boldsymbol{b}_1^2$ $(s_2 \times 1)$ and $\boldsymbol{b}_2^2$ (2×1). Let us notice that s_2 is the number of neurons in the hidden layer.

To train this inverse network, let us consider $N_t^{(2)}$ input–output vectors $\{(\widehat{\boldsymbol{\lambda}}^d)^{(i)}, (\nu^{(i)}, \rho^{(i)})\}_{i=1}^{N_t^{(2)}}$, where $(\widehat{\boldsymbol{\lambda}}^d)^{(i)} = \widehat{\mathcal{F}}^{\boldsymbol{\theta}_1}_{\Omega,\Gamma,N}(\nu^{(i)}, \rho^{(i)})$ (using the first RBF ANN), and let us define

$$J_{N_t^{(2)}}(\boldsymbol{\theta}_2) = \frac{1}{N_t^{(2)}} \sum_{i=1}^{N_t^{(2)}} ((\nu^{(i)}, \rho^{(i)})^T - \widehat{\mathcal{F}}^{\boldsymbol{\theta}_2}_{\Omega,\Gamma,N}((\widehat{\boldsymbol{\lambda}}^d)^{(i)}))^2 \tag{20}$$

Then, an optimal estimation for $\boldsymbol{\theta}_2$ is given by

$$\widehat{\boldsymbol{\theta}_2} = \inf_{\boldsymbol{\theta}_2} J_{N_t^{(2)}}(\boldsymbol{\theta}_2) \tag{21}$$

Let us remark that problem (21) can be solved iteratively using the same method used to obtain $\widehat{\boldsymbol{\theta}_1}$.

5. Numerical examples

In this section, numerical examples are presented in order to show the effectiveness and relevance of the proposed numerical method.

5.1. Example 1

In order to test the validity of our numerical procedure, we give in this subsection an approximation of the first eigenvalue of the Stokes operator. Let us notice that the exact eigenvalues of Stokes operator are not known; however, we take, as an adequate approximation of the first eigenvalue, the accurate reference value given by Wieners [7]: $\lambda_1 = 52.3446911$. In the Wieners case, $\Omega =]0.0, 1.0[\times]0.0, 1.0[$ is a square domain in $\mathbb{R}^2$, and the viscosity and density coefficients are, respectively, $\nu = 1.0$ and $\rho = 1.0$.

The training data set for the first (direct) RBF ANN is generated as follows: $\nu^{(i)} = 0.45 + 0.1(i-1)$ and $\rho^{(i)} = 1.0$ with $1 \le i \le N_t^{(1)} = 21$. Once the first network is trained, it is used to simulate a larger amount of data $N_t^{(2)}$, obtaining a training data set for the second (inverse) network: $\nu^{(i)} = 0.45 + 0.01\,(i-1)$ and $\rho^{(i)} = 1.0$ with $1 \le i \le N_t^{(1)} = 201$. Since the generation of the training data sets is done using FEM with $\mathbb{P}_2 - \mathbb{P}_1$ elements, the first (and thus the second) RBF ANN can not have a better convergence error than FEM with $\mathbb{P}_2 - \mathbb{P}_1$ elements. However, as we shall see in the following examples, once trained the corresponding RBF ANN, it can perform calculations very quickly, improving very well the computational time of the classical finite elements.

Tables 1 and 2 show, for different sizes $M \times M$ of the associated mesh, respectively, the application of the first trained network to the coefficients $\nu = 1.0$ and $\rho = 1.0$, and the application of the second trained network to the first eigenvalue documented in Wieners [7].

Fig. 1 shows numerical comparisons between our technique and other methodologies presented in the literature (see [6], [20] and [21]). Specifically, we show the relative errors, for different sizes $M \times M$ of the associated mesh, in order to obtain the first eigenvalue for the Stokes operator, considering $\nu = 1.0$ and $\rho = 1.0$. We compare with:

- $\mathbb{P}_2 - \mathbb{P}_1$ finite elements: FEM P2-P1.
- B–R mixed finite elements: FEM MIXED (see [21]).

Table 1
Application of the first (direct) RBF ANN in $\Omega \subset \mathbb{R}^2$.

Direct RBF ANN	ν	ρ	$M \times M$	Calculated eigenvalue $\widehat{\lambda_1}$
$\widehat{\mathcal{F}}^{\boldsymbol{\theta}_1}_{\Omega,\Gamma,N}$	1.0	1.0	4×4	53.3665
			8×8	52.4269
			16×16	52.3505
			32×32	52.3451

Table 2
Application of the second (inverse) RBF ANN in $\Omega \subset \mathbb{R}^2$.

Inverse RBF ANN	Wieners documented first eigenvalue	$M \times M$	Calculated coefficients $(\widehat{\nu}, \widehat{\rho})$
$\widehat{\mathcal{F}}^{\boldsymbol{\theta}_2}_{\Omega,\Gamma,N}$	52.3446911	4×4	(0.9805, 1.0)
		8×8	(0.9981, 1.0)
		16×16	(0.9995, 1.0)
		32×32	(0.9996, 1.0)

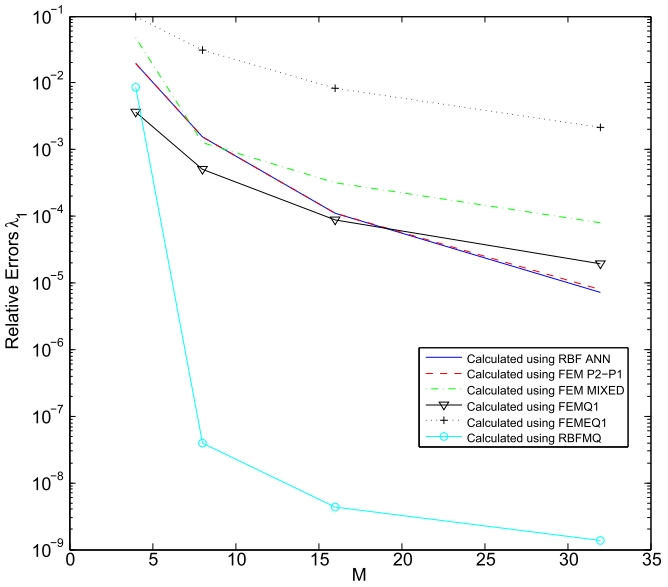

Fig. 1. Numerical Comparisons between our technique and other methodologies presented in the literature.

- $\mathbb{Q}_1^{\text{rot}}$ nonconforming finite elements: FEMQ1 (see [20]).
- $\mathbb{EQ}_1^{\text{rot}}$ nonconforming finite elements: FEMEQ1 (see [20]).
- Meshfree method based on multiquadric radial basis functions: RBFMQ (see [6]).

Clearly the results presented above show the validity of our approach, and also show its limitations in relation to the convergence error. We note that, as we have mentioned above, the neural networks designed can not have a better convergence error than FEM with $\mathbb{P}_2 - \mathbb{P}_1$ elements; however, the error can be improved by using, for training both networks, some methods with better convergence order. For example, the aforementioned mesh-free method based on multiquadric radial basis functions (RBFMQ) seems (see Fig. 1) to be an excellent candidate that could be used, in future investigations, in order to train the first (and the second) RBF ANN.

Finally, let us remark that the algorithm used to train both networks, in this example, is the backpropagation algorithm.

5.2. Example 2

Let us consider, also, in this example, a square domain $\Omega =]0.0, 1.0[\times]0.0, 1.0[) \subset \mathbb{R}^2$ and the following viscosity and density coefficients used for training the first RBF ANN: $\nu^{(i)} = 1002\frac{t(i)}{1-2t(i)}$ and $\rho^{(i)} = (1 + t(i))$, where $t(i) = 0.1 + 0.1(i-1)$ with $1 \leq i \leq N_t^{(1)} = 21$.

Once trained the network $\widehat{\mathcal{F}}^{\boldsymbol{\theta}_1}_{\Omega,\Gamma,N}$, using again the FEM technique with $\mathbb{P}_2 - \mathbb{P}_1$ elements in Ω (see Section 3), and calculated the associated vector $\widehat{\boldsymbol{\theta}_1}$, the direct network is used to simulate a larger amount of data $N_t^{(2)}$, obtaining a set of training data for the inverse network $\widehat{\mathcal{F}}^{\boldsymbol{\theta}_2}_{\Omega,\Gamma,N}$. In this case, $\nu^{(i)} = 1002\frac{t(i)}{1-2t(i)}$ and $\rho^{(i)} = (1 + t(i))$, where $t(i) = 0.1 +$

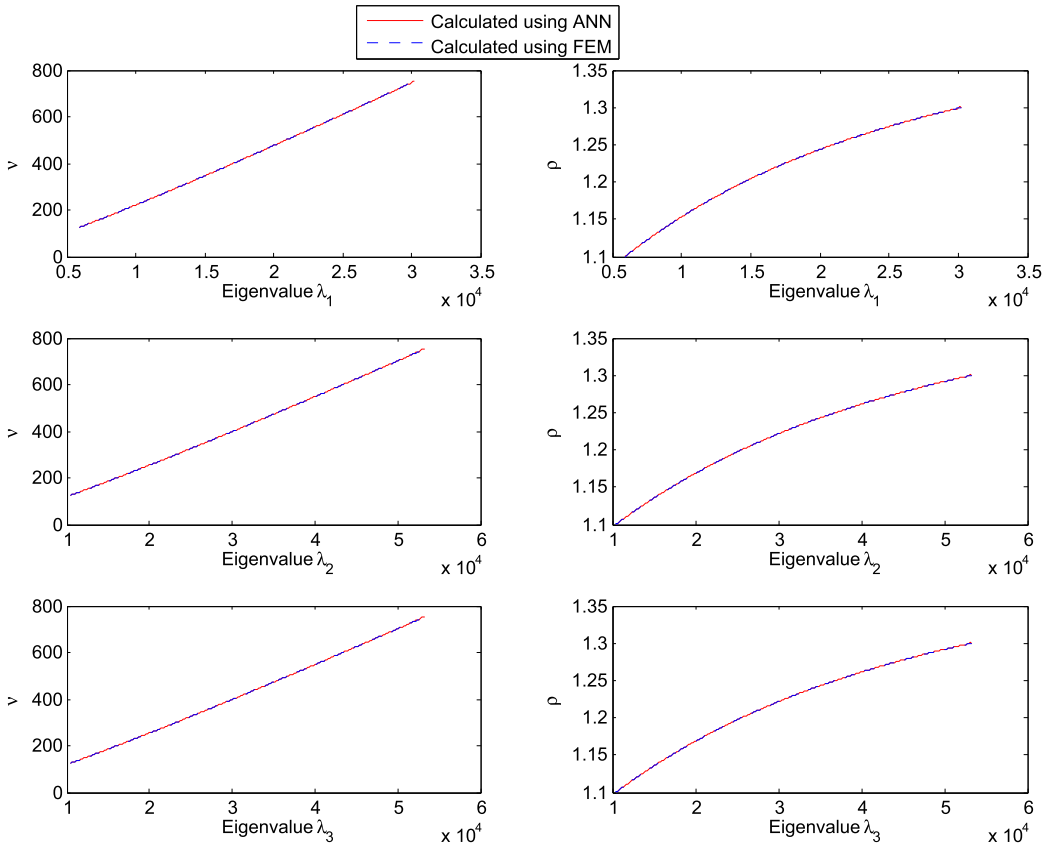

Fig. 2. Viscosity and density coefficients as a function of the first eigenvalues corresponding to $N = 3$: (1) calculated using second (inverse) RBF ANN (in solid red line), (2) calculated with FEM using the inverse functional relationship (in dashed blue line).

Table 3
Summary of the computational performance and of the computational time for the numerical example 2.

N_s	MSE ν	MSE ρ	CT ANN	CT FEM
2001	0.0550	3.7430e − 09	12.0052	782.27

$0.01(i-1)$ with $1 \leq i \leq N_t^{(1)} = 21$. The latter training gives us the value of $\widehat{\boldsymbol{\theta}}_2$. Let us remark that the algorithm used to train both networks is the backpropagation algorithm.

Fig. 2 shows a comparison of the viscosity and density coefficients as a function of the first eigenvalues corresponding to $N = 3$, when $\nu^{(i)} = 1002\frac{t(i)}{1-2t(i)}$ and $\rho^{(i)} = (1 + t(i))$ where $t(i) = 0.1 + 0.001(i-1)$ with $1 \leq i \leq N_t^{(1)} = 2001$: (1) calculated using inverse RBF ANN, (2) calculated with FEM using the inverse functional relationship. As seen in this figure, the coefficients calculated from the neural network method approach quite well the calculated eigenvalues using FEM (is not possible at this scale to remark the differences). Fig. 3 shows the relative error associated with this comparison. As seen in this figure, the relative error is negligible, showing a very good performance of our procedure.

Finally, Table 3 summarizes the computational performance using the mean squared error (MSE), the computational time, in seconds, using ANN (CT ANN) and FEM (CT FEM), required, respectively, for simulations of example 2. The computer used to obtain the above results have a 2.7-GHz Intel Core i5 processor with 8 GB 1600 MHz DDR3.

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 very good computational performance which is measured using the MSE. Finally, let us remark that in the case of a more complex geometry of the domain, it will be necessary to train with more data $N_t^{(1)}$ the direct RBF ANN with the purpose of improving the MSE.

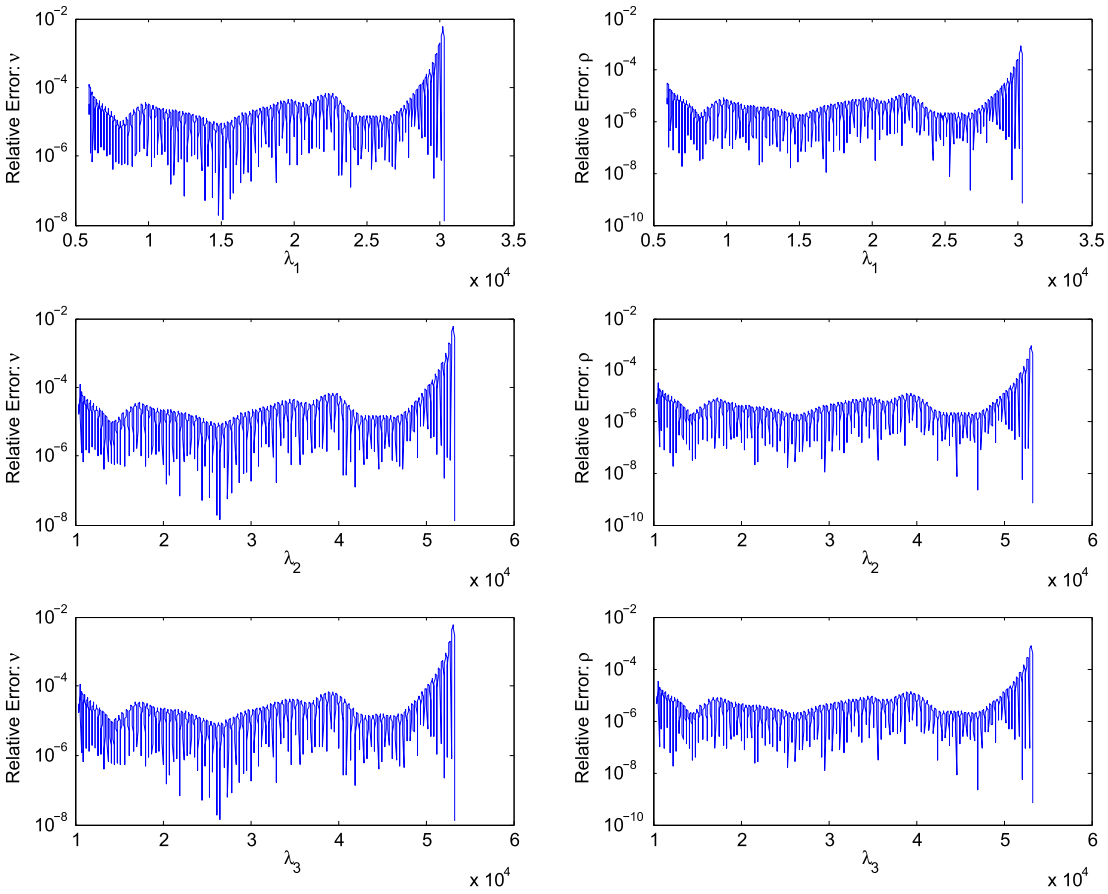

Fig. 3. Relative error of the calculated eigenvalues.

6. Conclusion

In this work, an efficient numerical methodology, based on the design of two ANNs and using the eigenvalues of the Stokes operator, is developed to obtain approximately the viscosity and density coefficients associated with the fluid under study. Once trained the first network (direct RBF ANN), using FEM with $\mathbb{P}_2 - \mathbb{P}_1$ elements in Ω, a second network (inverse RBF ANN) has been trained, using as a training set the eigenvalues calculated by simulation with the first one, and thus solving the aforementioned (inverse) problem. The numerical results obtained from computational simulations show that the calculation of the coefficients ν and ρ, associated with the viscosity and density respectively, is very efficient. The relative error is negligible, and the computation time associated is significantly smaller than the FEM technique, as seen in Table 3: in fact 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, has lower computational time than the FEM technique.

The neural network approach has shown to be useful, as an approximate method, for calculating the principal features of a fluid. However, in general, RBF ANN have the disadvantage that the performance is directly related to the training data, making them unsuitable to predict the viscosity and density from eigenvalues outside the scope of the data set. Moreover, the convergence error of the first (and second) RBF ANN designed can not have a better convergence error than the technique used to train these neural networks. However, this error can be improved by using, for training both networks, some methods with better convergence order. For example, the meshfree method based on multiquadric radial basis functions (RBFMQ) seems (see Example 1 and Fig. 1) to be an excellent candidate that could be used to train the first (and second) RBF ANN(s), in order to improve the convergence error.

Finally, let us mention that, while the more complex the geometry of the domain Ω, the more training data will be needed, and therefore the longer the computational time taken by methodology will be.

Acknowledgements

Sebastián Ossandón acknowledges support from the European Union's Horizon 2020, research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 644202: Geophysical Exploration using Advanced Galerkin Methods

(GEAGAM); C. Reyes thanks a partial support from Project Innova-Chile CORFO No. 10CEII-9007: “CSIRO-CHILE International Centre of Excellence in Mining and Mineral Processing”, Program 3-Project 1.

References

[1] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.
[2] F. Thomasset, Implementation of Finite Element Methods for Navier–Stokes Equations, Springer-Verlag, Berlin, 1981.
[3] T.J.R. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Dover Publications, Mineola, NY, USA, 2000.
[4] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Comput. Methods Appl. Mech. Eng. 197 (2007) 173–201.
[5] B. Mercier, J. Osborn, J. Rappaz, P.A. Raviart, Eigenvalue approximation by mixed and hybrid methods, Math. Comput. 36 (1981) 427–453.
[6] A. Golbabai, A mesh free method based on radial basis functions for the eigenvalues of transient Stokes equations, Eng. Anal. Bound. Elem. 36 (2012) 1555–1559.
[7] t. Wieners, A numerical existence proof of nodal lines for the first eigenfunction of the plate equation, Arch. Math. 66 (1996) 420–427.
[8] E. Leriche, Are there localized eddies in the trihedral corners of the Stokes eigenmodes in cubical cavity?, Comput. Fluids 43 (2011) 98–101.
[9] M. Baymani, S. Effati, A. Kerayechian, A feed-forward neural network for solving Stokes problem, Acta Appl. Math. 116 (2011) 55–64.
[10] M. Baymani, S. Effati, H. Niazmand, A. Kerayechian, Artificial neural network method for solving the Navier–Stokes equations, Neural Comput. Appl. 26 (2015) 765–773.
[11] F. Girosi, M. Jones, T. Poggio, Regularization theory and neural networks architectures, J. Neural Comput. 7 (1995) 219–269.
[12] S. Ossandón, C. Reyes, On the neural network calculation of the Lamé coefficients through eigenvalues of the elasticity operator, C. R. Mecanique 344 (2016) 113–118.
[13] S. Ossandón, C. Reyes, C.M. Reyes, Neural network solution of an inverse problem associated with the Dirichlet eigenvalues of the anisotropic Laplace operator, Comput. Math. Appl. 72 (2016) 1153–1163.
[14] S. Ossandón, C. Reyes, P. Cumsille, C.M. Reyes, Neural network approach for the calculation of potential coefficients in quantum mechanics, Comput. Phys. Commun. 214 (2017) 31–38.
[15] 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.
[16] I. Babuska, J. Osborn, Eigenvalue problems, in: P.G. Ciarlet, J.-L. Lions (Eds.), Handbook of Numerical Analysis, vol. II: Finite Element Methods (Part 1), North-Holland, Amsterdam, 1991.
[17] D. Boffi, F. Brezzi, M. Fortin, Mixed Finite Element Methods and Applications, Springer Series in Computational Mathematics, vol. 44, Springer, Heidelberg, 2013.
[18] F. Brezzi, M. Fortin, Mixed and Hybrid Finite Element Methods, Springer Series in Computational Mathematics, vol. 15, Springer-Verlag, 1991.
[19] V. Girault, P.A. Raviart, Finite Element Methods for Navier–Stokes Equations, 2nd edition, Springer Verlag, 1986.
[20] S. Jia, H. Xie, X. Yin, Approximation and eigenvalue extrapolation of Stokes eigenvalue problem by nonconforming finite element methods, Appl. Math. Czech. 54 (2009) 1–15.
[21] X. Yin, H. Xie, S. Jia, Asymptotic expansions and extrapolations of eigenvalues for the Stokes problem by mixed finite element methods, J. Comput. Appl. Math. 215 (2008) 127–141.