Journal of Thermal Engineering, Vol. 7, No. 4, pp. 791-805, May, 2021
Yildiz Technical University Press, Istanbul, Turkey

STUDY OF THE FLOW AND HEAT TRANSFER OF A VISCOELASTIC FLUID USING HYBRID NEURAL NETWORK-PARTICLE SWARM OPTIMIZATION (HNNPSO)

Reza Mirzaei[1] Mohammad Ghalambaz[2,3,*] Aminreza Noghrehabadi[1]

ABSTRACT

Fluid flow and heat transfer of a second-order viscoelastic fluid in an axisymmetric channel with a porous wall for turbine cooling applications are studied. The nonlinear differential equations of the fluid flow and heat transfer arising from similarity solutions are computed employing a Hybrid Neural Network-Particle Swarm Optimization algorithm (HNNPSO). A trial function, satisfying the boundary conditions, as a possible solution for the governing equations is introduced. The trial functions incorporate a multi-layer perceptron neural network with adjustable parameters (the weights and biases). The Particle Swarm Optimization algorithm (PSO) is applied to find the adjustable parameters of the trial solution to satisfy the governing equations. Finally, comparisons are made between the results of the present method (HNNPSO) and the results of the fourth order Runge–Kutta method, finite difference method, and Variational Iteration Method. The results indicate that HNNPSO method conveniently produces a polynomial analytic solution with remarkable accuracy, and the accuracy of the solution improves as the number of neurons of the neural network increases.

Keywords: *Ordinary differential equations (ODEs), Neural network (NN), Boundary layer, Viscoelastic fluid, Particle swarm optimization (PSO)*

INTRODUCTION

Viscoelastic fluids are a kind of non-Newtonian fluids which have both viscous and elastic characteristics. In these materials, the viscosity decreases as the shear/strain rate remains constant. Polymer solutions and melts like nylon, gelatin solution, and most of the industrial fluids are viscoelastic in nature [1].

Analysis of the behavior of the non-Newtonian fluid flows, especially the viscoelastic flows, have attracted much attention in the recent years due to the various industrial applications in different fields such as extrusion of plastics, lubrication and hot rolling. Generally, the interest in flow and heat transfer problem of non-Newtonian fluids has grown considerably [1, 2]. The boundary layer flow and heat transfer over surfaces is an important issue with many practical engineering applications. Various aspect of flow and heat transfer over surfaces of channels have been addressed in recent literature works. For instance, the boundary layer heat transfer of viscoelastic nanofluids [3], non-Newtonian nano-fluids [4], micropolar fluids [5], and radiation effects [6] have been addressed recently.

In most cases, the governing equations of the fluid motion in the boundary layer are inherently nonlinear in nature that cannot be solved analytically, therefore, special techniques such as numerical techniques or advanced analytical methods to tackle such nonlinear equations are needed. In this way, different methods such as Homotopy Perturbation Method (HPM) [7], Parameterized Perturbation Method (PPM) [8], Homotopy Analysis Method (HAM) [9], Adomian Decomposition Method (ADM) [10] and Variational Iteration Method (VIM) [11] have been developed to solve differential equation. Although these methods are very powerful and can deal with various types of nonlinear differential equations; they have also encountered some limitations and disadvantages. For example, the Homotopy Perturbation Method (HPM) and Parameter Perturbation Method (PPM) depend on perturbation, linearization, or small parameters. The Homotopy Analysis Method (HAM) requires many harmonic terms to produce an accurate

This paper was recommended for publication in revised form by Regional Editor Tolga Taner

[1] *Department of Mechanical Engineering, Shahid Chamran University of Ahvaz, Ahvaz, Iran.*

[2] *Metamaterials for Mechanical, Biomechanical and Multiphysical Applications Research Group, Ton Duc Thang University, Ho Chi Minh City, Vietnam*

[3] *Faculty of Applied Sciences, Ton Duc Thang University, Ho Chi Minh City, Vietnam.*

**E-mail address:* *mohammad.ghalambaz@tdtu.edu.vn*

Orcid ID: 0000-0002-0252-8963, 0000-0003-0965-2358, 0000-0001-8491-2397

Manuscript Received 27 June 2019, Accepted 24 September 2019

solution, which such a solution with too many analytic terms practically seems unworthy. VIM shows some disadvantages, namely, repeated computations and computations of unneeded terms, which consumes time and effort. Hence, a simple and powerful method to solve the nonlinear equations, which reduces these limitations and difficulties, is still a challenge.

Many researchers have studied the flow and heat transfer phenomena over open surfaces and channels. For instance, the MHD flows [12, 13, 14], flow over flat surfaces [15], natural and mixed convection flow and heat transfer [16-18], flow over a cylinder [19], and entropy generation [20] have been studied in past years. The non-Newtonian fluids are important in industrial applications as the behavior of many practical fluids cannot be described using the Newtonian model of fluids. In this regard, many researchers have studied the flow and heat transfer of non-Newtonian fluids. Among them, White and Metzner [21] have evaluated the constitutive equations for viscoelastic fluids. Debruge and Han [22] studied the heat transfer in a channel with a porous wall for turbine cooling application. Following the study of Debruge and Han [22], Kurtcebe and Erim [23] used the Perturbation Method (PM), and Shakeri et al. [24] applied Variational Iteration Method (VIM) to solve the heat transfer of non-Newtonian viscoelastic fluid flow in an axisymmetric channel with a porous wall. They [23, 24] analyzed the effects of different parameters namely Reynolds number, injection Reynolds number, Prandtl number, and power-law index. The same problem was also solved by Esmaeilpour et al. [25] using the Homotopy Analysis Method (HAM) to obtain an expression for velocity and temperature fields.

The intelligence exhibited by machines or software is known as artificial intelligence. An Artificial Neural Network (ANN) is a new form of artificial intelligence used in machine learning that tries to mimic the learning process of the human brain. Artificial neural networks are powerful tools for functions approximation. This method has been successfully applied to a wide variety of applications [26]. Particle swarm optimization (PSO) is a swarm-based artificial intelligence technique that tries to simulate the social behavior of biological populations [27]. This heuristic method can be used to find approximate solutions to numeric maximization and minimization problems.

In recent years, some new artificial methods have been introduced to solve differential equations arising from engineering problems [28-34]. Noghrehabadi et al. [30] studied the buckling shape of a multi-wall carbon nanotube cantilever. The buckling of the nanotube follows a fourth-order non-linear boundary value differential equation. They utilized the power series with adjustable variables as a trial function for the governing equation, satisfying the boundary conditions. Afterward, Artificial Bee Colony (ABC) algorithm was employed to adjust the power series variables to satisfy the governing differential equation in the domain of the solution. Using the power series as a trial function and ABC as the training method, Noghrehabadi et al. [30] have successfully reported a semi-analytic solution in the form of a power series for the buckling shape of the nanotube.

In another study, Noghrehabadi et al. [31] successfully utilized the Cuckoo search optimization algorithm as the training method to adjust the coefficients of a trial function for the buckling of a microactuator. Lagaris et al. [32] and Tawfiq and Hussein [33] have utilized the feedforward neural networks to obtain a solution for differential equations. Malek and Beidokhti [34] examined the Feed Forward Neural Networks (FFNN) to obtain a solution for high order differential equations. They utilized the Nelder-Mead optimization method to adjust the parameters of a trial solution, which was constructed using the feedforward neural networks.

The approaches utilized in [28-34] can be considered as hybrid methods. An advantage of the hybrid method is that it can provide an analytical approximation to a wide class of nonlinear equations without the computational difficulties of HAM. Also, there is no need for any perturbation, linearization, or small parameter as required in perturbation methods. The hybrid method is simple and powerful to solve different differential equations, while the results have excellent agreement with numerical methods. In the previous studies [28-34], the hybrid method was applied for a single high order differential equation. Although there are various practical systems of differential equations, there has been no attempt to develop the hybrid method to the case of the system of high order differential equations. Recently, Noghrehabadi et al. [28] utilized the hybrid ANN and PSO optimization method to solve the differential equation of nanofluids over a flat plate.

As mentioned, the literature review shows that the problem of heat transfer of a non-Newtonian fluid in a channel has been studied by previous researchers [22-25] using Perturbation Method (PM) [22], Variational Iteration Method (VIM) [24] and Homotopy Analysis Method (HAM) [25] due to its important practical applications and inherent non-linearity. The governing equations of heat transfer of a non-Newtonian fluid in a channel result in a system of high order non-linear differential equations, one high order boundary value equation for heat and one for flow.

The utilized method in the literature requires as mall value [22], or too many terms [23- 25] to provide an accurate analytical solution. The present study tends to obtain an accurate analytical solution with only a few terms for the model of heat transfer introduced in [22-25]. Here, the hybrid method is employed to deal with a system of high-order boundary value differential equations (a fourth-order differential equation and a second-order differential equation). For this purpose, two feed-forward neural networks are introduced as the trial functions, and the Particle Swarm Optimization (PSO) method as the training method is employed to obtain a semi-analytical solution for the heat transfer and fluid flow of a non-Newtonian fluid in a channel.

MATHEMATICAL FORMULATION

Consider the flow of a non- Newtonian viscoelastic fluid on a turbine disc to cool the surfaces of the disc. Fig. 1 shows the physical model of the problem and the selected coordinate system. The r-axis is adopted parallel to the surface of the disk, and the z-axis is normal to the surface of the disk. The porous disc of the channel is placed at z=+*L*. The wall that coincides with the r-axis is heated externally; a non-Newtonian fluid is injected uniformly from the other perforated wall.

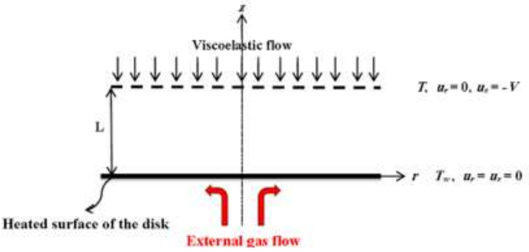

Figure 1. Physical model, along with the coordinate system

The flow and heat transfer equations and the similarity solution are proposed by Kurtcebe and Erim [23] and followed by Shakeri et al. [24] and Esmaeilpour et al. [25] for the above problem description. For a steady, axisymmetric and two-dimensional non-Newtonian fluid flow and neglecting dissipation effect, the governing equations for flow and heat can be represented as [23]:

$$f^{iv}(\eta) + 2Ref(\eta)f'''(\eta) + K_1Re\left(4f''(\eta)f'''(\eta) + 2f'f^{iv}(\eta)\right) = 0 \tag{1}$$

$$q_n'(\eta) - Re\,Pr\big(nf'(\eta)q_n(\eta) - 2f'q_n'(\eta)\big) = 0 \tag{2}$$

Where *f* and q_n are the fluid and heat variables, respectively and η is the domain variable. Here, K_1 is the viscoelastic parameter, *Re* is the injection Reynolds number, *Pr* is the Prandtl number, which are defined based on the first viscosity characteristic (φ_0). *n* is the order of heat equation which can be considered as a constant integer positive number. The corresponding boundary conditions for flow and heat are:

$$f(0) = 0, f'(0) = 0, f(1) = 0, f'(1) = 0 \tag{3}$$

$$q_n(0) = 0, q_n(1) = 0 \tag{4}$$

As seen, the momentum equation is a fourth order and highly non-linear equation, and the heat equation is a second order coupled to the momentum equation. To find an analytical solution, the hybrid neural network and swarm optimization method are introduced and utilized to solve these equations.

Artificial Neural Networks (ANNS)

An Artificial Neural Network (ANN) is a new mathematical model used in machine learning that tries to simulate the structure of biological neural networks and the learning process of the human brain. The basic element of every artificial neural network is an artificial neuron, which is a simple mathematical function [35].

Using the science of computer programming, one can design a data structure that acts as a neuron. Then, the desired network can be introduced and trained by establishing a network of interconnected artificial neurons and utilizing a learning algorithm. ANNs are known as universal approximations. These networks can efficiently estimate and approximate functions [36].

In cases that include too many input data, one neuron alone is not enough to solve the problem. Accordingly, there should be a layer of neurons. In a multi-layer neural network, there is an input layer, which receives the input data, some hidden layers, which receive the information from the previous layer. Ultimately, there is an output layer to deliver the outcomes. A wide variety of learning algorithms have been developed. A supervised network needs a teacher to tell the network what the desired output should be. In contrast, an unsupervised network adapts purely in response to its inputs. Such networks can learn to pick out a structure in their input.

In a feed-forward neural network, the neurons are in successive layers and send their output to the forwarded layers. Thus, there is no feedback, and the output layer does not affect the other layer. The topology of the neural network is an important factor. Multi-layer perceptron which first was introduced by Rosenblatt [37], is a simple feed-forward neural network that involves one input layer, one or more hidden layers, and one output layer. Fig.2 shows a typical three-layered perceptron.

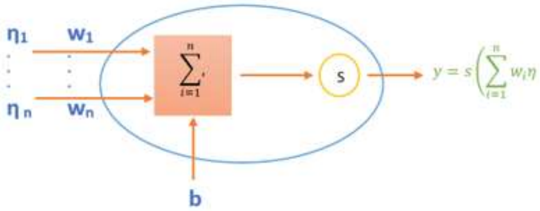

Figure 2. An artificial neural network (perceptron)

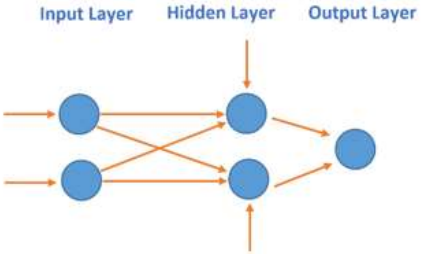

Figure 3. An artificial a multi-layered perceptron

The ability to function approximation is the most important application of a multi-layered perceptron. Based on the Kolmogorov theorem, a three-layered perceptron with *n(2n+1)* nodes can approximate a continuous function of n variables [38, 39]. Hence, it can be concluded that the accuracy of the function approximation is under the significant influence of the number of neurons in the hidden layer.

Considering an input of x, the network output can be evaluated by

$$N(\eta, p) = \sum_{i=1}^{H} v_i S(w_i \eta + b_i) \tag{5}$$

Where w_i denotes a weight parameter from the input layer to ith neuron in the hidden layer, b_i denotes a bias parameter for ith neuron in the hidden layer, and v_i denotes a weight parameter from ith neuron in the hidden layer to output layer [35]. Here, p is a vector containing w_i, v_i and b_i.

The multi-layered feed-forward neural networks are trainable. In this paper, a feed-forward neural network is used to form a non-linear function from the input data. Here, the ANN learns the function behavior from the theory of differential equations. During the learning process the parameters of ANN as adjusted in a way to satisfy the differential equations in some collocation point samples. An ANN with minimum deviation from the behavior of the ODE is the final solution and has learned the ODE adequately. Therefore, it can be concluded that an ANN that has learned the ODE is an analytic function that approximates the solution to the corresponding ODE.

Particle Swarm Optimization (PSO)

The Particle Swarm Optimization (PSO) is a population-based optimization technique proposed by Eberhart and Kennedy [27], inspired by the social behavior of biological populations. Like other swarm-based techniques, PSO starts with a population of random solutions and searches for the optimum solution by updating generations. In PSO, each particle has a velocity and a position that can move. Each particle adjusts its movement in the search space according to two pieces of information, which are the best position the particle has reached so far, which is called the personal best (pbest), and the best position obtained by any particle in the neighbors of the particle, which is called the local best (lbest). When a particle takes the whole swarm as its neighbor, the value is called the global best (gbest). In the PSO, each particle i has a position Z_i and a velocity v_i that is updated at each iteration:

$$\vec{v_\iota} = \omega\vec{v_\iota} + c_1 r_{1i}\left(\vec{\rho_\iota} - \vec{Z_\iota}\right) + c_2 r_{2i}\left(\vec{\rho_g} - \vec{Z_\iota}\right) \tag{6}$$

Where ω is the inertia weight, $\vec{\rho_\iota}$ is the personal best found by particle, $\vec{\rho_g}$ is the global best, r_1 and r_2 are random values between [0,1], and c_1 and c_2 are positive constant parameters called acceleration coefficients which determine the effect of the cognitive and social behavior of the particles in the velocity vector. The position of each particle is the sum of the present position and the distance that the particles have traveled. Hence, the new positions of the particles can be evaluated as:

$$\vec{Z_\iota} = \vec{Z_\iota} + \vec{v_\iota} \tag{7}$$

Some important factors in PSO are inertia weight (ω), particle size and maximum iteration number (t). Generally, the PSO algorithm can be arranged in five key steps:

- Initialize random velocity and position vectors for all particles. Then, maximum velocity and maximum particle movement amplitude are specified.
- Evaluate the fitness of each particle.
- If the obtained solution is better than the previous local best solution, replace it. If the existing solution is better than the previous global best solution, then the existing solution is selected as the new gbest.
- Change positions and velocities by computing the new position employing Eqs. (6) and (7).
- Repeat steps 2 to 4 until the gbest reaches a satisfactory optimized solution.

Hybrid Method For A System Of Differential Equations

In the hybrid method, mentioned in [30-34], a trial function with adjustable coefficients is introduced. The trial function consists of the sum of two parts. The first part is meant to satisfy the boundary conditions and the second part, which is an FFNN here, is meant to satisfy the differential equation in the domain of the solution. This technique has been referred to as the collocation neural network in the study of Tawfiq and Hussein [33]. In this section, we will propose the formulation of the hybrid method for a system of boundary value differential equations consisting of two non-linear differential equations. Now, assume the general form of a system of differential equations as

$$z(y^{\varepsilon}, y^{\varepsilon-1}, \dots, y^{1}, y, \eta) = 0 \tag{8}$$

Subject to

$$b(y^{\varepsilon}, y^{\varepsilon-1}, \dots, y^{1}, y, \eta) = 0 \tag{9}$$

Where $z=z_1, z_2 \dots z_j$ are the governing differential equation, and $y=y_1, y_2, \dots y_j$ are the differential equation variables which are a function of the independent variable of η (the solution in domain D that should be computed). Here, $b=b_1, b_2 \dots b_L$ are the boundary equations. The indexes of j and L denote the number of differential equations and boundary conditions respectively. Consider a trial function, $Y_T(\eta, a, p)$, with adjustable parameters of a and p, that exactly satisfies the boundary conditions introduced by b in Eq. (9). It should be noted that $Y_T = Y_{T1}, Y_{T2} \dots Y_{Tj}$ and a and p are vectors of adjustable coefficients for each trial function of $Y_{T1,\ YT2\ \dots\ YTj}$. Considering $Y_T(\eta, a, p)$ as a trial solution with adjustable parameters of a and p, the problem is transformed into a discretized form as:

$$Min \sum_{\eta_i \epsilon D} \left(z\left({Y_T}^{\varepsilon}(\eta_i, a, p), {Y_T}^{\varepsilon-1}(\eta_i, a, p), \dots, {Y_T}^{1}(\eta_i, a, p), Y_T(\eta_i, a, p), \eta_i \right) \right) \tag{10}$$

In our proposed method the trial solution of Y_T employs a polynomial ($A(\eta, a)$), containing the adjustable coefficients of a to satisfy the boundary conditions and a function containing a feed-forward neural network $(G(\eta, N(\eta, p))$ as:

$$Y_T(\eta, a, p) = A(\eta, a) + G\big(\eta, N(\eta, a)\big) \tag{11}$$

$A(\eta, a) = A_1(\eta, a_1), A_2(\eta, a_2) \dots A_j(\eta, a_j)$ and $G(\eta, N(\eta, p)) = G_1(\eta, N_1(\eta, p_1)), G_2(\eta, N_2(\eta, p_2)) \dots G_j(\eta, N_j(\eta, p_j))$ where each of $a_1, a_2 \dots a_j$ and $p_1, p_2 \dots p_j$ are vectors containing adjustable parameters. Each $N_i(\eta, p)$, such as $N_1(\eta, p_1)$ is a single layer feed-forward neural network the same as the one that was introduced in Eq. (5), and pi contains the weights (w and v) and biases of the neural network of N_i. The function of G is constructed so as not to contribute to the boundary conditions of b since $A(\eta, a)$ would satisfy them. After introducing proper trial functions for $Y_{T1}, Y_{T2} \dots Y_{Tj}$ the minimization equation of Eq. (10) can be formed as the sum of the square of the residuals of each trial function as:

$$\begin{aligned} Min \sum_{\eta_i \epsilon D} \Bigg\{ & \left(\left(z_1\left({Y_T}^{\varepsilon}(\eta_i, a, p), {Y_T}^{\varepsilon-1}(\eta_i, a, p), \dots, {Y_T}^{1}(\eta_i, a, p), Y_T(\eta_i, a, p), \eta_i \right) \right) \right)^2 \\ & + \left(\left(z_2\left({Y_T}^{\varepsilon}(\eta_i, a, p), {Y_T}^{\varepsilon-1}(\eta_i, a, p), \dots, {Y_T}^{1}(\eta_i, a, p), Y_T(\eta_i, a, p), \eta_i \right) \right) \right)^2 + \cdots \\ & + \left(\left(z_j\left({Y_T}^{\varepsilon}(\eta_i, a, p), {Y_T}^{\varepsilon-1}(\eta_i, a, p), \dots, {Y_T}^{1}(\eta_i, a, p), Y_T(\eta_i, a, p), \eta_i \right) \right) \right)^2 \Bigg\} \end{aligned} \tag{12}$$

Now, a minimization technique such as PSO, which was introduced in the previous section, is required to adjust the coefficients of a and p.

At this end, it is worth mentioning that the present formulation extends the ideas proposed in the previous studies of [28-34] to the case of a system of differential equations. In addition, in the studies of [28-34] the function of $A(\eta)$ which satisfies the boundary conditions is solely a function of the independent variable of η. However, in the present study, the function A is considered as a function free of the independent variable of η and the adjustable coefficient of a. In this way, the trial function would have a higher degree of freedom next to the boundary conditions to satisfy the governing equation. Hence, using few degrees of freedom in the form of adjustable coefficients of a would significantly increase the accuracy of the trial solution with only a few adjustable coefficients.

In the next section, the present method will be employed to obtain a solution for the governing equations of Eq. (1) and (2) subject to the boundary conditions of Eqs. (3) and (4).

Problem Formulation

Consider the governing equations of fluid flow and heat transfer of a second-order viscoelastic fluid in an axisymmetric channel with a porous wall which is expressed by Eqs. (1) and (2) and is subject to the boundary conditions (3) and (4). Let’s assume as an approximation solution (trial function) to Eq. (1) and as an approximation solution (trial function) to Eq. (2) as

$$Y_{T1}(\eta, a, p) = a_{1,1}\eta^4 - 2(a_{1,1}+1)\eta^3 + (a_{1,1}+3)\eta^2 + \eta^2(\eta-1)^2 N_1(\eta, p_1) \tag{13}$$

$$Y_{T1}(\eta, a, p) = -\eta + 1 + \eta(\eta-1)N_2(\eta, p_2) \tag{14}$$

In Eqs. (13) and (14), the terms $a_{1,1}\eta^4 - 2(a_{1,1}+1)\eta^3 + (a_{1,1}+3)\eta^2$ and $= -\eta + 1$ denote $A_1(\eta,\ a_1)$ and $A_2(\eta,\ a_2)$, respectively. It should be noted that for Y_{T1} only one adjustable coefficient of a11 is utilized, and for Y_{T2} no adjustable parameter of a is utilized, i.e., a_{21}=0. The terms of $\eta^2(\eta-1)^2N_1(\eta, p_1)$ and $\eta(\eta-1)N_2(\eta, p_2)$ shows $G_1(\eta,\ N_1(\eta,\ p_1))$ and $G_2(\eta,\ N_2(\eta,\ p_2))$ parts of the trial solution of Y_T respectively. $N_1(\eta,\ p_1)$ and $N_2(\eta,\ p_2)$ are perceptron neural networks (simple FFNN) that involves adjustable parameters (p) and could involve one or more neurons. In the present study, perceptron neural networks with three, four, and five neurons will be utilized. Attention to Eqs. (13) and (14) shows that Eq. (13) satisfies all given boundary conditions in (3) and Eq. (14) satisfies all given boundary conditions in Eq. (4). Now, substituting the trial functions of Y_{T1} and Y_{T2} in the discretized governing equations of Eqs. (1) and (2) and forming the sum of the square of the residuals similar to Eq. (12) results in:

$$\begin{aligned} in \sum_{\eta_i \epsilon D} \Bigg\{ \Big(& {Y_{T1}}^{iv}(\eta_i, a, p) + 2ReY_{T1}(\eta_i, a, p){Y_{T1}}'''(\eta_i, a, p) \\ & - K_1Re\left(4{Y_{T1}}''(\eta_i, a, p){Y_{T1}}'''(\eta_i, a, p) + 2{Y_{T1}}'(\eta_i, a, p){Y_{T1}}^{iv}(\eta_i, a, p)\right)\Big)^2 \\ & + \Big({Y_{T2}}''(\eta_i, a, p) \\ & - Re\ Pr\big({Y_{T1}}'(\eta_i, a, p)Y_{T2}(\eta_i, a, p) \\ & - 2Y_{T1}(\eta_i, a, p){Y_{T2}}'(\eta_i, a, p)(\eta_i, a, p)\big)\Big)^2\Bigg\} \end{aligned} \tag{15}$$

Where $\forall \eta_i \epsilon D, i = 1,2,3,\dots,m$ are collocation points in domain D. As mentioned Y_{T1} and Y_{T2} are the trial functions with adjustable parameters (ANNs). Y_{T1} and Y_{T2} will approximate the solution of differential equations after their parameters were adjusted. Here, Eq. (15) builds a framework to adjust the parameters of the trial functions. Eq. (15) contains the trial function and its gradients respect to η. In the present study, we utilized a sigmoid transfer function as (1/(1+exp(-η)) for the transfer function of the neurons for convenience. PSO is utilized to determine the optimal adjustable parameters of a_{11}, p_1, and p_2 in the trial functions of Y_{T1} and Y_{T2} to minimize Eq. (15). Thus, Eq. (15) can be rewritten as

$$Min \sum_{\eta_i \epsilon D} \big(E_1(a, p) + E_2(a, p)\big) \tag{16}$$

Where

$$E_1(a,p) = \sum_{\eta_i \epsilon D} \Big({Y_{T1}}^{iv}(\eta_i,a,p) + 2ReY_{T1}(\eta_i,a,p){Y_{T1}}'''(\eta_i,a,p) \\ - K_1 Re \left(4{Y_{T1}}''(\eta_i,a,p){Y_{T1}}'''(\eta_i,a,p) + 2{Y_{T1}}'(\eta_i,a,p){Y_{T1}}^{iv}(\eta_i,a,p) \right) \Big)^2 \tag{17}$$

And

$$E_2(a,p) = \sum_{\eta_i \epsilon D} \Big({Y_{T2}}''(\eta_i,a,p) \\ - Re\,Pr\big({Y_{T1}}'(\eta_i,a,p)Y_{T2}(\eta_i,a,p) \\ - 2Y_{T1}(\eta_i,a,p){Y_{T2}}'(\eta_i,a,p)(\eta_i,a,p)\big)\Big)^2 \tag{18}$$

RESULTS AND DISCUSSION

The hybrid method and PSO algorithm were coded in MATLAB 2009 [40]. Then, the consequent results of the presented method are compared with the results of the fourth order Runge–Kutta method [24] and the VIM method [24] in literature. Besides, the governing equations (Eqs. (1) and (2) subject to the boundary conditions in Eqs. (3) and (4)) were solved using a finite difference method [41].

In order to solve the equations using the present method, the coefficients of the trial functions for H=5 (five neurons) with sigmoid units in the hidden layer and m=6 equally spaced points inside the interval [0, 1] was determined by minimizing Eq. (15) using PSO. In all of this section, the results are reported for a neural network with five neurons; otherwise, the number of the utilized neurons will be stated. The following parameters of PSO were used for this problem: Inertia weight = 0.9; Acceleration factors = 2.5; Maximum Iteration=150 and Population = 100. Table 1 displays the results of the optimization of Eqs. (15) using PSO. The optimal adjustable parameters in trial function $Y_{T1}(\eta, a, p)$ and $Y_{T2}(\eta, a, p)$ are summarized in Table 1 for a case with K_1=0.01, Re=0.5 and Pr=1 and n=0. The optimal adjustable parameter in the first part of the trial function $Y_{T1}(\eta, a, p)$, i.e., a_{11}, was found to be a_{11}=-0.2308 while the obtained errors are and. According to the results of Table 1, $Y_{T1}(\eta, a, p)$ and $Y_{T2}(\eta, a, p)$ can be written as:

$$Y_{T1}(\eta,a,p) = -0.2308\eta^4 - 1.5384\eta^3 + 2.7692\eta^2 \\ + \eta^2(\eta-1)^2 \Big(\frac{1.0375}{1+e^{-0.8006\eta \quad .0975}} + \frac{-0.6621}{1+e^{-1.2355\eta-1.4787}} \\ + \frac{-0.4258}{1+e^{1.7691\eta-0.3692}} + \frac{0.5842}{1+e^{0.7106\eta-1.1208}} + \frac{-0.3809}{1+e^{0.1903\eta+2.9867}} \Big) \tag{19}$$

$$Y_{T1}(\eta,a,p) = -\eta + 1 \\ + \eta(\eta-1) \Big(\frac{0.9601}{1+e^{-0.6640\eta \quad .1741}} + \frac{0.2593}{1+e^{-0.6552\eta+0.2774}} \\ + \frac{0.5473}{1+e^{-0.3878\eta \quad .2180}} + \frac{-1.5850}{1+e^{-1.9623\eta+0.1756}} + \frac{-0.7294}{1+e^{2.1112\eta \quad .4522}} \Big) \tag{20}$$

It should be noted that Y_{T1} represents the solution for variable f and Y_{T2} represents the solution for variable q_n in the domain of D. The results of the present are summarized in Table 2. This table shows the results computed using Y_{T1} and Y_{T2} consisting of three, four and five neurons. It is obvious that the accuracy of the solution improves as the number of neurons of the neural network increases. Next, the results for $f(\eta)$ and $q_n(\eta)$ as a function of η have been shown in Table 3. A comparison between the hybrid method with five neurons and the numerical solutions (fourth order Runge-Kutta and the finite difference method) is reported in this table. In Table 3, %Error is introduced as %Error= $\left|\frac{f(\eta)_{NM}-f(\eta)_{HNNPSO}}{f(\eta)_{NM}}\right|$where NM refers to the fourth order Runge-Kutta method.

Table 1. The evaluated optimal values of the adjustable parameters in trial function of YT1(η, a, p) and YT2(η, a, p) when K_1=0.01, *Re*=0.5 and *Pr*=1 and *n*=0

Index(i)	$Y_{T1}(\eta, a, p)$			$Y_{T2}(\eta, a, p)$		
	w_i	b_i	v_i	w_i	b_i	v_i
1	0.8006	1.0975	1.0375	0.6640	-0.1741	0.9601
2	1.2355	1.4787	-0.6621	0.6552	-0.2744	0.2593
3	-1.7691	0.3692	-0.4258	0.3878	0.2180	0.5473
4	-0.7106	1.1208	0.5842	1.9623	-0.1756	-1.5850
5	-0.1903	-2.9867	-0.3809	-2.1112	-0.4522	-0.7294

Table 2. A comparison between the results obtained by the hybrid method using three, four and five neurons when K_1=0.01, *Re*=0.1, *Pr*=1 and *n*=0

	Y_{T1} **or** $f(\eta)$			Y_{T2} **or** $q_n(\eta)$		
	3 neurons	**4 neurons**	**5 neurons**	**3 neurons**	**4 neurons**	**5 neurons**
0	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000
0.1	0.029559	0.029561	0.029561	0.884689	0.884733	0.884718
0.2	0.109302	0.109307	0.109309	0.769865	0.769859	0.769828
0.3	0.225720	0.225775	0.225794	0.656363	0.656256	0.656218
0.4	0.365639	0.365624	0.365621	0.545325	0.545191	0.545161
0.5	0.515643	0.515639	0.515400	0.438203	0.438162	0.438097
0.6	0.663187	0.663164	0.663156	0.336491	0.336404	0.336407
0.7	0.796160	0.796132	0.796123	0.241363	0.241283	0.241223
0.8	0.903397	0.903316	0.903307	0.153483	0.153330	0.153295
0.9	0.974389	0.974385	0.974385	0.073053	0.072952	0.072931
1	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000

Results show that the hybrid method gives an analytical solution with high accuracy by utilizing only a few neurons. A comparison between the results of the present HNNPSO and the results reported in [24] is performed in Figs. 3 and 4 and Table 3. The finite difference method used in Table 3 was implemented using bvp4c functions of MATLAB.

Table 3. Comparison between results of the fourth-order Runge–Kutta method (NM) [24], VIM [24], finite difference method and the hybrid method, when: *Re*=0.5, K_1=0.01, *Pr*=1 and *n*=0

η	$f(\eta)$					$q_n(\eta)$				
	NM [24]	**VIM [24]**	**Finite difference**	**HNNPSO (5 neurons)**	**%Error**	**NM [24]**	**VIM [24]**	**Finite difference**	**HNNPSO (5 neurons)**	**%Error**
0	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000	1.000000	0.000000
0.1	0.029627	0.029627	0.029561	0.029561	0.002228	0.884720	0.884133	0.884719	0.884718	0.000002
0.2	0.109442	0.109442	0.109311	0.109309	0.001215	0.769829	0.769044	0.769826	0.769828	0.000001
0.3	0.225897	0.225896	0.225795	0.225794	0.000455	0.656223	0.655468	0.656216	0.656218	0.000008
0.4	0.365597	0.365594	0.365624	0.365621	0.000065	0.545172	0.544558	0.545162	0.545161	0.000020
0.5	0.515454	0.515447	0.515644	0.515400	0.000104	0.438113	0.437672	0.438099	0.438097	0.000034
0.6	0.662849	0.662836	0.663077	0.663156	0.000463	0.336423	0.336142	0.336409	0.336407	0.000047
0.7	0.795795	0.795778	0.796081	0.796123	0.000412	0.241237	0.241082	0.241225	0.241223	0.000058
0.8	0.903067	0.903056	0.903297	0.903307	0.000265	0.153305	0.153239	0.153298	0.153295	0.000065
0.9	0.974297	0.974290	0.974382	0.974385	0.000090	0.072938	0.072921	0.072933	0.072931	0.000096
1	1.000000	0.999962	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Comparing the present method with other solutions, the results for *f(η)* and *f '(η)* are shown in Figs. (4) and (5), respectively. The results for $q_n(\eta)$ are also plotted in Fig. (6). Figures show that the results of the present HNNPSO are in good agreement with the numerical solution (fourth order Runge-Kutta method [24]) and the VIM method [24].

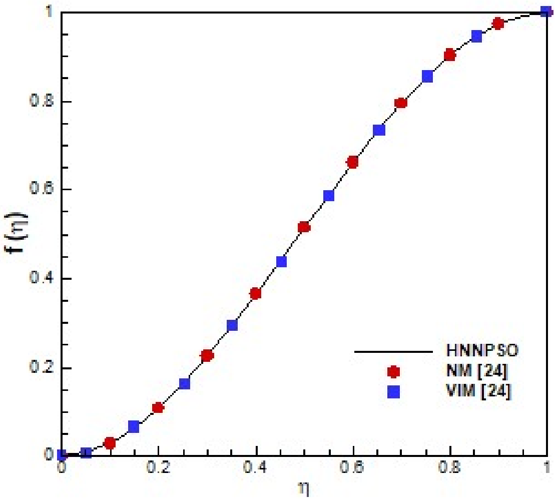

Figure 4. Comparison between the solutions via numerical results, VIM and HNNPSO for *f* (*η*) when: *Re* =0.5, K_1=0.01

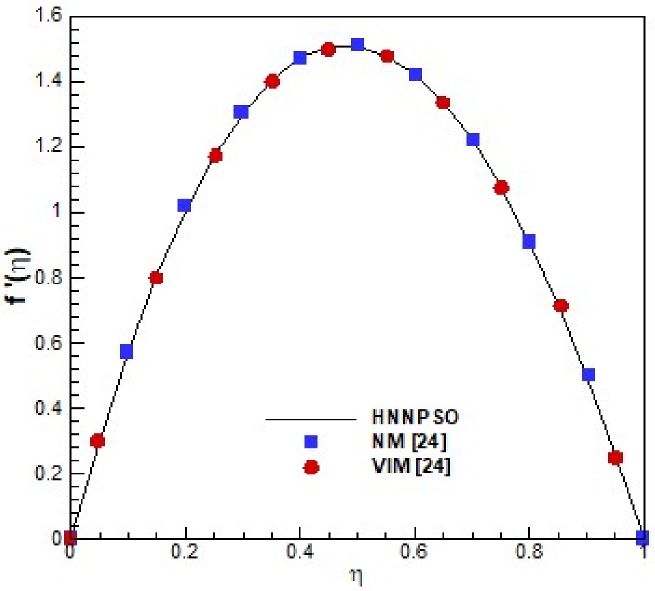

Figure 5. Comparison between the solutions via numerical results, VIM and HNNPSO for $f'(\eta)$ when: Re=0.5, K_1=0.01

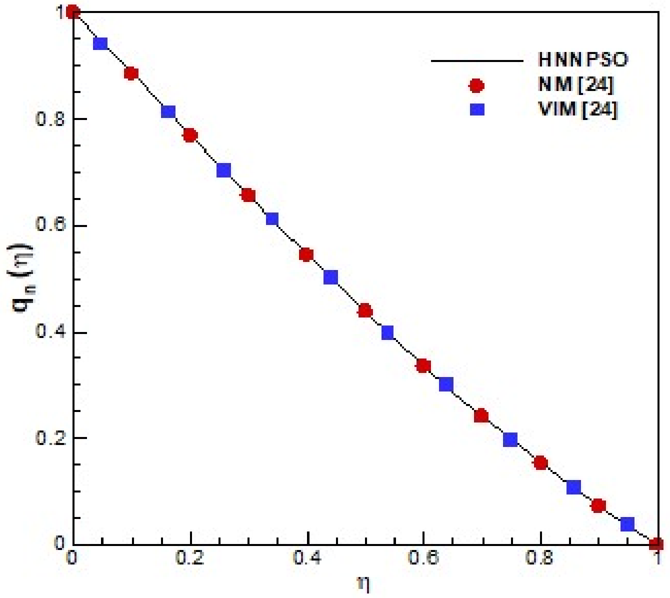

Figure 6. Comparison between the solutions via numerical results, VIM and HNNPSO for $q_n(\eta)$ when: Re=0.5, K_1=0.01, Pr=1 and n=0

Figs. (7) and (8) show the effect of various parameters namely Reynolds number (Re), dimension parameter (n), Prandtl number (Pr) and K_1 parameter on the flow and heat transfer. The effect of different Re values on the velocity profile is shown in Fig. (7), while Fig. (8) shows the effect of various parameters on the temperature profile. Again, excellent agreement is seen between the results of the proposed method and the results of the numerical solution. Figs. (7) and (8) show that increasing Reynolds number leads to an increase in the curvature of the temperature profile and decrease of $q_n(\eta)$ values. Also, from Fig. (8) can be concluded that for a constant value of η, temperature increases if the power-law index decreases.

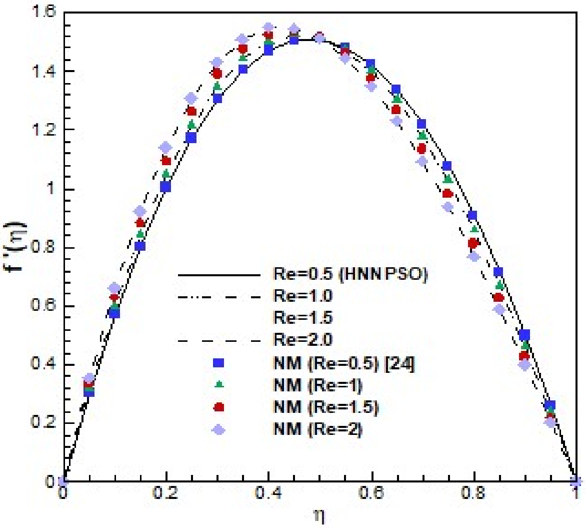

Figure 7. Comparison effect of various Reynolds number (*Re*) on *f* (*η*), when K_1=0.01

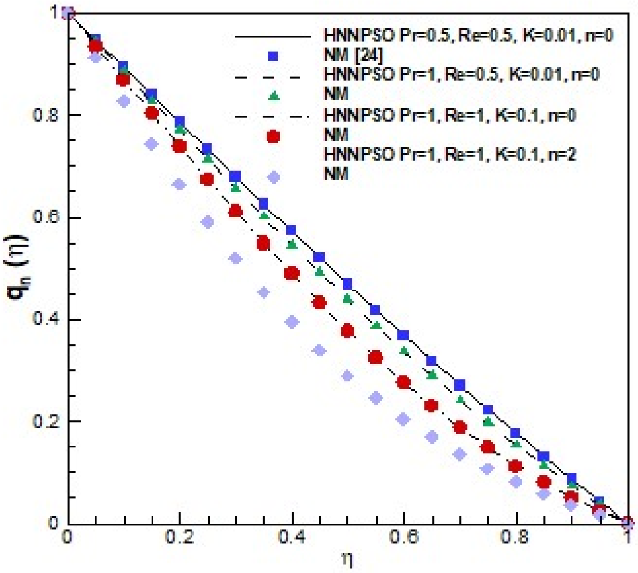

Figure 8. Comparison effect of various parameters on $q_n(\eta)$

CONCLUSION

The main purpose of this study was to apply Hybrid Neural Network-Particle Swarm Optimization (HNNPSO) to nonlinear differential equation arising from the similarity solution of the flow and heat transfer of a second-order viscoelastic fluid in an axisymmetric channel with a porous wall. PSO was applied to minimize the error function and find the optimized values of the adjustable parameters of the trial function. The obtained solution is a closed-form solution and represents a remarkable accuracy in comparison with the numerical methods and the VIM method, and the accuracy improves as the number of neurons of the neural network increases. Using three neurons can provide acceptable results for most of graphical and engineering purpose while using five neurons (five non-linear sigmoid terms) can provide a relative error lower than 0.01%. Therefore, the present method can be utilized to obtain highly compact and accurate close-form analytical relations for boundary layer problems, arising in various engineering problems.

Moreover, in the present study a uniform collocation point is utilized. However, a non-uniform sample points can be utilized to increase the local accuracy of the close form solution, which can be subject of future studies.

ACKNOWLEDGMENTS

The second and third authors acknowledge Shahid Chamran University of Ahvaz for the support of the present work.

NOMENCLATURE

b	Boundary equations
b_i	Bias parameter for ith neuron in the hidden layer
c_1, c_2	Positive constant parameters called acceleration coefficients.
K_1	Viscoelastic parameter
p	A vector containing w_i, v_i and b_i
Pr	Prandtl number
r_1, r_2	Random values between [0,1]
Re	Reynolds number
T	Temperature, ºC
v_i	Weight parameter from ith neuron in the hidden layer to output layer
w_i	Weight parameter from the input layer to ith neuron in the hidden layer
y	Differential equation variables
Y_T	Trial function
z	Governing differential equation

Greek symbols

ω	Inertia weight
$\overrightarrow{\rho_i}$	Personal best found by particle
$\overrightarrow{\rho_g}$	Global best
η	Independent variable

Subscripts

n	Refers to order of heat equation

REFERENCES

[1] Barnes HA. A Handbook of Elementary Rheology, University of Wales Institute of non-Newtonian fluid mechanics, 2000.

[2] Chhabra RP, Richardson JF. Non-Newtonian Flow in the Process Industries, Butterworth-Heinemann, Oxford, 1999.

[3] Almakki M, Mondal H, Sibanda P. Entropy Generation in MHD Flow of Viscoelastic Nanofluids with Homogeneous-Heterogeneous Reaction, Partial Slip and Nonlinear Thermal Radiation. Journal of Thermal Engineering, 2020: 6(3): pp.327-45.

[4] Kezzar M, Nafir M, Tabet I, Khanetout A. A New Analytical Investigation of Natural Convection of Nanofluids Flow Non-Newtonian between Two Vertical Flat Plates by the Generalized Decomposition Method (GDM). Journal of Thermal Engineering, 2018: 4(6): 2496-508.

[5] Umavathi JC, Chamkha AJ, Marudappa S. Free Convection Flow of an Electrically-Conducting Micropolar Fluid between Parallel Porous Vertical Plates Using Differential Transform. Journal of Applied and Computational Mechanics, 2018: 4(4): 286-98.

[6] Mondal H. Unsteady MHD Micropolar Fluid in a Stretching Sheet over an Inclined Plate with the Effect of Non-Linear Thermal Radiation and Soret-Dufour Journal of Thermal Engineering, 2019: 5(6): 205-13.

[7] He JH. Homotopy perturbation method for bifurcation of nonlinear problems, Product Information International Journal of Nonlinear Sciences and Numerical Simulation 2005;6:207–18.

[8] He J. Some new approaches to Duffing equation with strongly and high order nonlinearity (II) parametrized perturbation technique, Communications in Nonlinear Science and Numerical Simulation 1999; 4: 81-3.

[9] Liao S. Proposed Homotopy Analysis Techniques for the Solution of Nonlinear Problems, Ph.D. dissertation, Shanghai Jiao Tong University, 1992.

[10] Adomian G. Solving Frontier Problems of Physics: The Decomposition Method, Fundamental Theories of Physics, Kluwer Academic Publishers, 1994.

[11] He J. A new approach to nonlinear partial differential equations, Communications in Nonlinear Science and Numerical Simulation 1997; 2: 230-5.

[12] Chamkha AJ, Aly AM. MHD Free Convection Flow of a Nanofluid Past a Vertical Plate in the Presence of Heat Generation or Absorption Effects, Chemical Engineering Communications 2011; 198: 425-41.

[13] Sheikholeslami M, Chamkha AJ. Influence of Lorentz Forces on Nanofluid Forced Convection Considering Marangoni Convection, Journal of Molecular Liquids 2017; 225: 750-7.

[14] Reddy PS, Chamkha AJ, Al-Mudhaf A. MHD Heat and Mass Transfer Flow of a Nanofluid Over an Inclined Vertical Porous Plate With Radiation and Heat Generation/Absorption, Advanced Powder Technology 2017; 28: 1008-17.

[15] Ghasemian A, Dinarvand S, Adamian A, Sheremet MA. Unsteady General Three-Dimensional Stagnation Point Flow of a Maxwell/Buongiorno Non-Newtonian Nanofluid, Journal of Nanofluids, 2019; 8: 1544-59.

[16] Loenko DS, Shenoy A, Sheremet MA. Natural Convection of Non-Newtonian Power-Law Fluid in a Square Cavity with a Heat-Generating Element. Energies 2019; 12.

[17] Sheremet MA, Pop I. Natural Convection Combined with Thermal Radiation in a Square Cavity Filled with a Viscoelastic Fluid. International Journal of Numerical Methods for Heat & Fluid Flow 2018; 28: 624–40.

[18] Pop I, Sheremet M. Free Convection in a Square Cavity Filled with a Casson Fluid Under the Effects of Thermal Radiation and Viscous Dissipation. International Journal of Numerical Methods for Heat & Fluid Flow 2017; 27: 2318–32.

[19] Reddy GJ, Kethireddy B, Umavathi JC, Sheremet MA. Heat Flow Visualization for Unsteady Casson Fluid past a Vertical Slender Hollow Cylinder, Thermal Science and Engineering Progress 2018; 5: 172–81.

[20] Reddy GJ, Kumar M, Umavathi JC, Sheremet MA. Transient Entropy Analysis for the Flow of a Second Grade Fluid over a Vertical Cylinder, Canadian Journal of Physics 2018; 96: 978–91.

[21] White JL, Metzner AB. Constitutive Equations for Viscoelastic Fluids with Application to Rapid External Flows, AIChE Journal 1965; 11: 324-30.

[22] Debruge LL, Han LS. Heat Transfer in a channel with a Porous Wall for Turbine Cooling Application, Journal of Heat Transfer 1972; 94: 385-90.

[23] Kurtcebe C, Erim MZ. Heat Transfer of non-Newtonian Viscoelastic Fluid in an Axisymmetric Channel with a Porous wall for Turbine Cooling Application, International Communications in Heat and Mass Transfer 29 (7) (2002) 971-982.

[24] Shakeri F, Abbasi A, Naeimaei M, Yekrangi A, Eftari M. Variational Iteration Method for the Heat Transfer of a non-Newtonian Fluid Flow in an Axisymmetric Channel with a Porous Wall, World Applied Sciences Journal 2012; 16: 26-30.

[25] Esmaeilpour M, Domairry D, Sadoughi N, Davodi AG. Homotopy Analysis Method for the Heat Transfer of a non-Newtonian Fluid Flow in an Axisymmetric Channel with a Porous Wall, Communications in Nonlinear Science and Numerical Simulation 2010; 15: 2424-2430.

[26] Picton P. Neural Networks, Palgrave Macmillan, Great Britain, 2000.

[27] J. Kennedy, R. Eberhart, Particle swarm optimization, IEEE International Conference on Neural Networks 1995; 4: 1948.

[28] Noghrehabadi A, Mirzaei R, Ghalambaz M, Chamkha A, Ghanbarzadeh A. Boundary Layer Flow Heat and Mass Transfer Study of Sakiadis Flow of Viscoelastic Nanofluids Using Hybrid Neural Network-Particle Swarm Optimization (HNNPSO), Thermal Science and Engineering Progress, 2017; 4: 150-9.

[29] Behrang MA, Assareh E, Ghalambaz M, Assari MR, Noghrehabadi AR. Forecasting Future Oil Demand in Iran Using GSA (Gravitational Search Algorithm), Energy 2011; 36: 5649-54.

[30] Noghrehabadi AR, Ghalambaz M, Ghalambaz M, Ghanbarzadeh A. A Hybrid Power Series Artificial Bee Colony Algorithm to Obtain a Solution for Buckling of Multiwall Carbon Nanotube Cantilevers near Small Layers of Graphite Sheets, Applied Computational Intelligence and Soft Computing 2012; 683483.

[31] Noghrehabadi A, Ghalambaz M, Ghalambaz M, Vosough A. A Hybrid Power Series-Cuckoo Search Optimization Algorithm to Electrostatic Deflection of Micro Fixed-Fixed Actuators, International Journal of Multidisciplinary Sciences and Engineering 2011; 2: 22-6.

[32] Lagaris IE, Likas A, Fotiadis DI. Artificial Neural Networks for Solving Ordinary and Partial Differential Equations, Neural Networks 1998; 9: 987-1000.

[33] Tawfiq LN, Hussein AA. Design Feed Forward Neural Network to Solve Singular Boundary Value Problems, ISRN Applied Mathematics 2013.

[34] Malek A, Beidokhti RS. Numerical Solution for High Order Differential Equations Using a Hybrid Neural Network—Optimization Method, Applied Mathematics and Computation 183 (1) (2006) 260-271.

[35] Cavuto DJ, An Exploration and Development of Current Artificial Neural Network Theory and Applications with Emphasis on Artificial Life, Master of Engineering dissertation, 1997.

[36] El-Bouri A, Balakrishnan S, Popplewell N. Sequencing Jobs on a Single Machine: A Neural Network Approach, European Journal of Operational Research 2000; 126: 474-90.

[37] Minsky SP. Perceptrons: An Introduction to Computational Geometry, M.I.T. Press, Cambridge, 1987.

[38] Lippmann RP. An Introduction to Computing with Neural Nets, IEEE ASSP Magazine 1987; 4: 4-22.

[39] Hornik K, Stinchcombe M, White H. Multilayer Feedforward Networks are Universal Approximators, Neural Networks 1989; 2: 359-66.

[40] MATLAB and Statistics Toolbox Release 2009a, The MathWorks, Inc., Natick, Massachusetts, United States.

[41] Shampine LF, Kierzenka J, Reichelt MW. Solving Boundary Value Problems for Ordinary Differential Equations in MATLAB with bvp4c, Available at http://www.mathworks.com/bvp_tutorial, 2003.