World Academy of Science, Engineering and Technology
International Journal of Mechanical and Mechatronics Engineering
Vol:5, No:1, 2011

A New Solution for Natural Convection of Darcian Fluid about a Vertical Full Cone Embedded in Porous Media Prescribed Wall Temperature by using a Hybrid Neural Network-Particle Swarm Optimization Method

M.A.Behrang, M. Ghalambaz, E. Assareh, A.R. Noghrehabadi

***Abstract*—**Fluid flow and heat transfer of vertical full cone embedded in porous media is studied in this paper. Nonlinear differential equation arising from similarity solution of inverted cone (subjected to wall temperature boundary conditions) embedded in porous medium is solved using a hybrid neural network- particle swarm optimization method.

To aim this purpose, a trial solution of the differential equation is defined as sum of two parts. The first part satisfies the initial/ boundary conditions and does contain an adjustable parameter and the second part which is constructed so as not to affect the initial/boundary conditions and involves adjustable parameters (the weights and biases) for a multi-layer perceptron neural network. Particle swarm optimization (PSO) is applied to find adjustable parameters of trial solution (in first and second part). The obtained solution in comparison with the numerical ones represents a remarkable accuracy.

***Keywords*—**Porous Media, Ordinary Differential Equations (ODE), Particle Swarm Optimization (PSO), Neural Network (NN).

I. Introduction

MOST scientific problems such as boundary layer similarity solution for forced, natural and mix convection in porous medium, and other fluid mechanic and heat transfer problems are inherently of nonlinearity. Except a limited number of these problems, most of them do not have analytical solution. Therefore, these nonlinear equations should be solved using other methods see [1-12]. Recently, artificial intelligence techniques are used in order to solve various nonlinear differential equations [13-18].

The prediction of natural convection heat transfer characteristics from heated bodies embedded in a porous media has a number of thermal engineering applications [1].

M.A. Behrang is with Mechanical Engineering Department of Islamic Azad University, Dezful Branch, Iran (corresponding author to provide phone: +989169478017; e-mail: Mohammadali.behrang@gmail.com).

M. Ghalambaz is a PhD candidate of fluid mechanics of Shahid Chamran University of Ahvaz (e-mail: M.Ghalambaz@yahoo.com).

E. Assareh is with Mechanical Engineering Department of Islamic Azad University, Dezful Branch, Iran (e-mail: Ehsanolah.assareh@gmail.com).

A.R. Noghrehabadi is with Mechanical Engineering Department of Shahid Chamran University of Ahvaz.

Many problems have been investigated about external natural convection in a porous medium adjacent to heated bodies in the form of flat plate [2, 3], cylinder [2, 4, 5], sphere [2, 5, 6] and cone [1, 2, 7-12].

This study presents a reliable method to solve the nonlinear ordinary differential equations of natural convection of Darcian fluid about a vertical full cone embedded in porous media. A remarkable accuracy for the presented method is achieved when the obtained results are compared with numerical solution.

The paper is organized as follows: Governing equation of fluid flow and heat transfer of full cone embedded in porous medium is presented in Sections 2. A brief review of Artificial Neural Networks (ANN) and Particle Swarm Optimization (PSO) are brought in Sections 3 and 4, respectively. Mathematical formulation to solve mentioned equations in section 2 is discussed in section 5. Numerical results are discussed in Section 5. Finally, conclusions and directions for future research are presented in section 6.

II. Governing Equation

Consider an inverted cone with semiangle γ and take axes in the manner indicated in Fig.1(a). The boundary layer develops over the heated frustum $x = x_0$. In terms of the streamfunction ψ defined by:

$$u = \frac{1}{r}\frac{\partial \psi}{\partial y}, \qquad v = -\frac{1}{r}\frac{\partial \psi}{\partial x} \tag{1}$$

The boundary layer equations are:

$$\frac{1}{r}\frac{\partial^2 \psi}{\partial y^2} = \frac{g\beta K}{\nu}\frac{\partial T}{\partial y}$$
$$\frac{1}{r}\left(\frac{\partial \psi}{\partial y}\frac{\partial T}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial T}{\partial y}\right) = \alpha_m \frac{\partial^2 T}{\partial y^2} \tag{2}$$

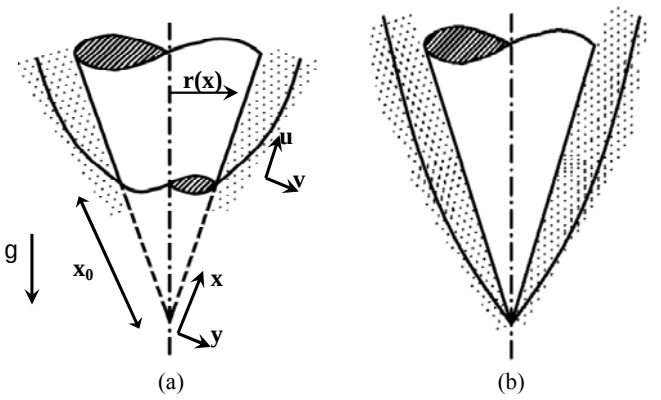

Fig. 1: (a) Coordinate system for the boundary layer on a heated frustum of a cone, (b) full cone, $x_0 = 0$

For a thin boundary layer we have approximately r = x sin (γ). We suppose the temperature is power function of distance from the vertex of the inverted cone. Accordingly, the boundary conditions are:

$$\begin{aligned} &u = 0, T = T_\infty && y \to \infty \\ &u = 0 && y = 0, x_0 \le x < \infty \\ &T = T_w = T_\infty + A(x - x_0)^\lambda && y = 0, x_0 \le x < \infty \end{aligned} \quad (3)$$

For the case of a full cone ($x_0 = 0$, Fig. 1(b)) a similarity solution exists.

In the case of prescribed wall temperature, we let:

$$\begin{aligned} &\psi = \alpha_m r Ra_x^{1/2} f(\eta), \\ &T - T_\infty = (T_w - T_\infty)\theta(\eta), \\ &\eta = \frac{y}{x} Ra_x^{1/2}, \end{aligned} \quad (4)$$

where

$$Ra_x = \frac{g\beta K \cos(\gamma)(T_w - T_\infty)x}{\upsilon \alpha_m} \quad (5)$$

The dimensionless momentum and energy equations are, Ref. [7]:

$$f' = \theta$$

$$\theta'' + \left(\frac{\lambda + 3}{2}\right) f\theta' - \lambda f'\theta = 0 \quad (6)$$

subjected to boundary conditions as:

$$f(0) = 0, \theta(0) = 1, \theta(\infty) = 0 \quad (7)$$

Finally from Eq. (2.6) and (2.7) we have:

$$\begin{cases} \text{ODE.} & f''' + \left(\frac{\lambda + 3}{2}\right) f f'' - \lambda (f')^2 = 0, \quad D : \eta \in [0, \infty] \\ \text{B.C.} & f(0) = 0, f'(0) = 1, f'(\infty) = 0, \end{cases} \quad (8)$$

This study uses a hybrid artificial neural network-particle swarm optimization (HNNPSO) method to solve Eq.8. A brief review of Artificial Neural Networks (ANN) and Particle Swarm Optimization (PSO) are brought in Sections 3 and 4, respectively.

III. Governing Equation

Neural networks are computational models of the biological brain. Like the brain, a neural network comprises a large number of interconnected neurons. Each neuron is capable of performing only simple computation [19].Any how, the architecture of an artificial neuron is simpler than a biological neuron. ANNs are constructed in layer connects to one or more hidden layers where the factual processing is performance through weighted connections. Each neuron in the hidden layer joins to all neurons in the output layer. The results of the processing are acquired from the output layer. Learning in ANNs is achieved through particular training algorithms which are expanded in accordance with the learning laws, assumed to simulate the learning mechanisms of biological system [20]. However, as an assembly of neurons, a neural network can learn to perform complex tasks including pattern recognition, system identification, trend prediction, function approximation, and process control [19].

Multi-layer Perceptron (MLPs) are perhaps the most common type of feedforward networks [21]. Their application in function approximation is well known [13]. Fig.2 shows an MLP which has three layers: an input layer, an output layer and a hidden layer.

Neurons in input layer only act as buffers for distributing the input signals x_i to neurons in the hidden layer. Each neurons j (Fig. 3) in the hidden layer sums up its input signals x_i after weighting them with the strengths of the respective connections w_{ji} from the input layer and adding the bias b_i to them, and computes its output n_j as a function g of the sum, viz.

$$n_j = g\left(\sum w_{ji} x_i + b_j\right) \quad (9)$$

where n_j is each neuron output and g can be a simple threshold function or a sigmoid, Gaussian, hyperbolic tangent or radial basis function.

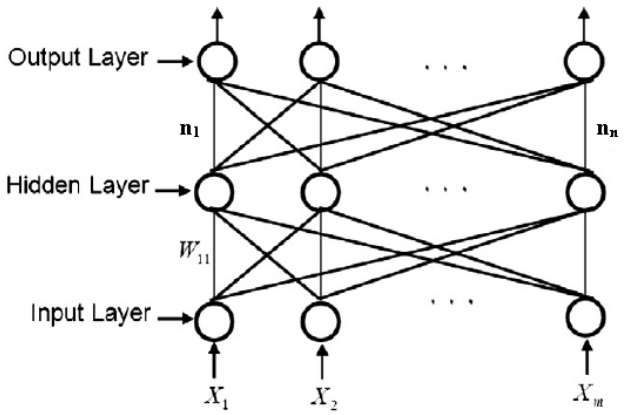

Fig. 2. A Multi-Layer Perceptron

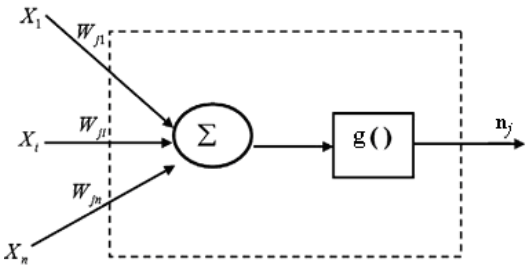

Fig. 3. Details of a neuron

From Kolmogorov existence theorem, any continuous function of n variable can be approximated using a three-layered perceptron with $n(2n+1)$ nodes [22, 23]. So, the accuracy of the approximation dose not depend on the number of the hidden layers, but it fully depends on the number of neurons in the hidden layer [13].

Fig.4 shows a three-layered perceptron with one input x, one hidden layer consisting of H neuron, and one output $N(x,p)$. This structure is used in the present method. For each entry x the network output is computed by $N(x,p)=\sum_{i=1}^{H}(v_i\, g(w_i\, x+b_i))$, which w_i is a weight parameter from input layer to ith neuron in hidden layer; v_i is a weight parameter from ith neuron in hidden layer to output layer, and b_i is a bias for ith neuron in hidden layer. For more details readers are referred to [24].

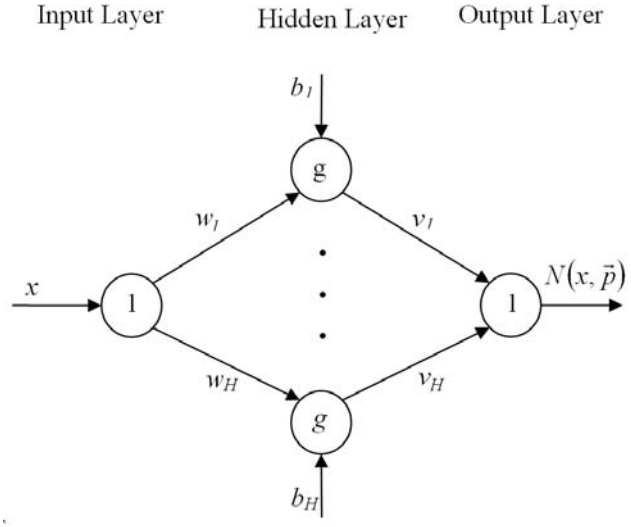

Fig. 4. Details of a three layerd perceptron with one input, one hidden layer (consist H neuron), and one output

Some of benefits of hybrid ANNs and optimization techniques are listed in [16].

The training of an MLP network involves the minimization of an error function. As it is mentioned subsequently, this study lets the network learn from the theory of differential equations in order to approximate a function consisting adjustable parameters.

IV. Particle Swarm Optimization (PSO)

The Particle Swarm Optimization algorithm was first proposed by Eberhart and Kennedy [25], inspired by the natural flocking and swarming behavior of birds and insects. The concept of PSO gained in popularity due to its simplicity. Like other swarm-based techniques, PSO consists of a number of individuals refining their knowledge of the given search space. The individuals in a PSO have a position and a velocity and are denoted as particles. The PSO algorithm works by attracting the particles to search space positions of high fitness. Each particle has a memory function, and adjusts its trajectory according to two pieces of information, the best position that it has so far visited, and the global best position attained by the whole swarm. If the whole swarm is considered as a society, the first piece of information can be seen as resulting from the particle's memory of its past states, and the second piece of information can be seen as resulting from the collective experience of all members of the society. Like other optimization methods, PSO has a fitness evaluation function that takes each particle's position and assigns it a fitness value. The position of highest fitness value visited by the swarm is called the global best. Each particle remembers the global best, and the position of highest fitness value that has personally visited, which is called the local best.

Many attempts were made to improve the performance of the original PSO algorithm and several new parameters were introduced such as the inertia weight [26]. The canonical PSO with inertia weight has become very popular and widely used in many science and engineering problems [27-30].

In the canonical PSO, each particle i has position z_i and velocity υ_i that is updated at each iteration according to Eq.2.

$$\vec{\upsilon}_i = \omega\vec{\upsilon}_i + c_1\vec{\varphi}_{1i}(\vec{\rho}_i - \vec{z}_i) + c_2\vec{\varphi}_{2i}(\vec{\rho}_g - \vec{z}_i) \quad (10)$$

Where ω is the inertia weight described in [31, 32], $\vec{\rho}_i$ is the best position found so far by particle $\vec{\rho}_i$, and $\vec{\rho}_g$ is the global best so far found by the swarm. $\vec{\varphi}_1$ and $\vec{\varphi}_2$ weights that are randomly generated at each step for each particle component. c_1 and c_2 are positive constant parameters called acceleration coefficients (which control the maximum step size the particle can achieve). The position of each particle is updated at each iteration by adding the velocity vector to the position vector.

$$\vec{z}_i = \vec{z}_i + \vec{\upsilon}_i \quad (11)$$

The inertia weight ω (which is a user-defined parameter), together with c_1 and c_2, controls the contribution of past velocity values to the current velocity of the particle. A large inertia weight biases the search towards global exploration, while a smaller inertia weight directs toward fine-tuning the current solutions (exploitation). Suitable selection of the inertia weight and acceleration coefficients can provide a balance between the global and the local search [26]. The PSO algorithm is composed of 5 main steps:

1. Initialize the position vector z and associated velocity υ of all particles in the population randomly. Then set a maximum velocity and a maximum particle movement

amplitude in order to decrease the cost of evaluation and to get a good convergence rate.

2. Evaluate the fitness of each particle via the fitness function. There are many options when choosing a fitness function and trial and error is often required to find a good one.

3. Compare the particle's fitness evaluation with the particle's best solution. If the current value is better than previous best solution, replace it and set the current solution as the local best. Compare the individual particle's fitness with the population's global best. If the fitness of the current solution is better than the global best's fitness, set the current solution as the new global best.

4. Change velocities and positions by using Eqs.(10) and (11).

5. Repeat step 2 to step 4 until a stopping criterion is satisfied or a predefined number of iterations is completed. Particle size (n), inertia weight (ω) and maximum iteration number (t) are considered as important factors in PSO.

As it is mentioned earlier, the multi layered feed forward neural networks are trainable. So, many different kinds of training algorithm can be used to gain optimum adjustable parameters for the corresponding multi-layered perceptron. In some cases which are called "ill-conditioned problems", the traditional training algorithms can not determine the adjustable parameters (weights and biases) properly [13]. Using particle swarm optimization algorithm in presented method helps not suffer from such difficulties. For more details readers are referred to [33-35].

V. Problem formulation

Consider governing equation of fluid flow and heat transfer of full cone embedded in porous medium which is expressed by Eq. (8) for wall temperature boundary condition.

In order to solve Eq.(8), assume a discretization of the domain D with m arbitrary points. Now, the problem can be transformed to the following set of equations:

$$f'''(\eta_i)+\left(\frac{\lambda+3}{2}\right)f(\eta_i)f''(\eta_i)-\lambda\left(f'(\eta_i)\right)^2=0, \qquad (12)$$

$$\forall \eta_i \in D, \quad i=1,2,...,m$$

subject to given boundary conditions.

Let's assume $y_T(\eta,\vec{P})$ as an approximation solution to Eq.(8) where $\vec{P}$ is a vector which contains adjustable parameters. These parameters (i.e. adjustable parameters) should be determined by minimizing the following sum of squared errors, subject to given conditions in (8)

$$E(\vec{P})=\sum_{i=1}^{m}\left(\frac{d^3y_T(\eta_i,P)}{d\eta^3}+\left(\frac{\lambda+3}{2}\right)y_T(\eta_i,P)\frac{d^2y_T(\eta_i,P)}{d\eta^2}-\lambda\left(\frac{dy_T(\eta_i,P)}{d\eta}\right)^2\right)^2 \qquad (13)$$

$$, \ 0\le\eta_i\le\infty$$

In order to transform Eq.13 to an unconstrained problem $y_T(x,\vec{P})$ is written in the following form:

$$Y_T(\eta,\vec{P})=a\eta^3+\frac{1+3ac^2}{2c}\eta^2+\eta+\eta^2(\eta-c)^2N(\eta,\vec{q}) \qquad (14)$$

where a is an adjustable parameter and c is a large enough amount which represents physical infinity. $N(\eta,\vec{q})$ is a three layer perceptron neural network which involves adjustable parameters (including weights and biases). So $\vec{P}$ can be defined as follow: $\vec{P}=(a,\vec{q})$. Eq (14) satisfies all given conditions in (8).

In order to calculate $E(\vec{P})$, the trial function $\left(y_T(\eta,\vec{P})\right)$ derivations respect to independent variable η is needed (see Appendix). Although there are many different choices for the neural network transfer functions, sigmoid transfer function $\left(\frac{1}{1+\exp(-\eta)}\right)$ is used in this study.

Now, an optimization technique like PSO can be applied in order to determine optimal adjustable parameters of $y_T(\eta,\vec{P})$ (i.e. $\vec{P}$) to minimize $E(\vec{P})$ in Eq (2-13).

VI. Numerical results

The algorithm was coded with MATLAB 2007. In order to demonstrate the presented method (HNNPSO), a fair comparison is made between presented method in this study (HNNPSO) and the other numerical methods (using MAPLE software). To aim this purpose, same step size (i.e. a step size of 0.2) and physical infinity amount (i.e. c=10) are used in this study which has been used in other numerical methods.

The method is successfully used to solve Eq.(8). The following combination of user-specified parameters of PSO was used for this problem:

Inertia weight = 0.9
Population = 100
Maximum Iteration=150
Acceleration factors = 2.5

Table 1 shows the obtained optimal adjustable parameters in trial function (i.e. $\vec{P}$) and $E(\vec{p})$ for various values of λ.

The results for $f'(\eta)$ have been shown in Table2 with two selected λ as 0 and 3/4 and have been made comparison between the HNNPSO solution and numerical solution.

Relative errors show that HNNPSO gives analytical solution with high degree of accuracy. Fig.5 shows the results of this study and the other numerical solution. $f'(\eta)$ for various value of λ are shown in Fig.5.

VII. Conclusion

Nonlinear differential equation arising from similarity solution of inverted cone embedded in porous medium was solved using a hybrid neural network- particle swarm optimization method.

The proposed approach is quite general. PSO was applied in order to find the adjustable parameters of trial function regarding to minimize a fitness function including these parameters (i.e. adjustable parameters). These parameters were determined so that the trial function has to satisfy the boundary conditions. The obtained solution in comparison

with the numerical ones represents a remarkable accuracy. Future work is focused on comparing the effect of changing optimization technique on minimizing the error function.

TABLE I OBTAINED $E(\vec{p})$ AND OPTIMAL ADJUSTABLE PARAMETERS IN TRIAL FUNCTION (i.e. $\vec{P}$) FOR DIFFERENT VALUES OF $\lambda = 1$

λ	a	$E(\vec{p})$	Adjustable parameters in ANN ($\vec{q}$)		
			ω_i [1]	b_i	v_i
0	0.00812144	4.65E-05	-0.84466	-5.55654	0.930942
			-0.56966	-7.1947	-5.48311
			-0.08243	-9.82999	-29.9622
1/4	0.00823658	1.88E-05	-0.91726	-4.78682	0.38473
			-0.61797	-7.15912	-5.22255
			-0.09206	-9.77156	-31.9281
1/2	0.00833206	1.10E-05	-1.11502	-5.42388	0.32241
			-0.53856	-7.25598	-3.96877
			-0.09064	-9.8666	-35.0701
3/4	0.00841269	7.06E-06	-1.32533	-5.70586	0.246702
			-0.50177	-7.25416	-3.8014
			-0.08855	-9.82857	-33.3897
1	0.00848659	5.24E-06	-1.56399	-5.86258	0.174057
			-0.48938	-7.1582	-3.60679
			-0.08676	-9.90936	-35.9438

[1] i is the index for the weights and biases

TABLE II COMPARISON BETWEEN HNNPSO SOLUTION AND NUMERICAL SOLUTION FOR $f'(\eta)$ WITH λ=0 AND λ=1/2

η	λ=0, f ' (η)			λ=3/4, f ' (η)		
	HNNPSO Solution	Numerical Solution	Relative Error	HNNPSO Solution	Numerical Solution	Relative Error
0	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000
0.2	0.8477	0.8471	0.0008	0.7984	0.7983	0.0001
0.4	0.7036	0.7032	0.0006	0.6281	0.6282	0.0002
0.6	0.5732	0.5737	0.0009	0.4882	0.4886	0.0008
0.8	0.4598	0.4611	0.0030	0.3757	0.3762	0.0013
1	0.3641	0.3658	0.0047	0.2869	0.2873	0.0014
1.2	0.2854	0.2868	0.0053	0.2178	0.2180	0.0009
1.4	0.2218	0.2227	0.0039	0.1645	0.1645	0.0001
1.6	0.1713	0.1714	0.0005	0.1239	0.1238	0.0006
1.8	0.1316	0.1310	0.0046	0.0930	0.0929	0.0004
2	0.1007	0.0997	0.0103	0.0697	0.0698	0.0019

APPENDIX

$$N(x,\vec{p}) = \sum_{i=1}^{H} v_i S(w_i x + b_i), \quad S(x) = \left(\frac{1}{1+\exp(-x)}\right)$$

where *S* is sigmoid transfer function.

Derivation of $y_T(x,\vec{P})$ with sigmoid transfer function $\left(\frac{1}{1+\exp(-x)}\right)$ is as follow:

$$Y_T(x,\vec{P}) = ax^3 + \frac{1+3ac^2}{2c}x^2 + x + M(x)N(x,\vec{P})$$

$$\frac{d\,y_T(x,\vec{P})}{dx} = 3ax^2 + \frac{1+3ac^2}{c}x + 1 + M'(x)N(x,\vec{P}) + M(x)N'(x,\vec{P})$$

$$\frac{d^2\,y_T(x,\vec{P})}{dx^2} = 6ax + \frac{1+3ac^2}{c} + M''(x)N(x,\vec{P}) + 2M'(x)N'(x,\vec{P}) + M(x)N''(x,\vec{P})$$

$$\frac{d^3\,y_T(x,\vec{P})}{dx^3} = 6a + M'''(x)N(x,\vec{P}) + 3M''(x)N'(x,\vec{P}) + 3M'(x)N''(x,\vec{P}) + M(x)N'''(x,\vec{P})$$

where

$$\frac{d^k N(x,\vec{P})}{dx^k} = \sum_{i=1}^{H} v_i w_i^k S^{(k)}(t_i)$$

$$t_i = w_i x + b_i$$

,

$$S' = -S^2 + S$$

$$S'' = 2s^3 - 3S^2 + S$$

$$S''' = -6S^4 + 12s^3 - 7S^2 + S$$

,

$$M(x) = x^2(x-c)^2$$

$$M'(x) = 2x(x-c)^2 + 2x^2(x-c)$$

$$M''(x) = 2x^2 + 8x(x-c) + 2(x-c)^2$$

$$M'''(x) = 12(2x-c)$$

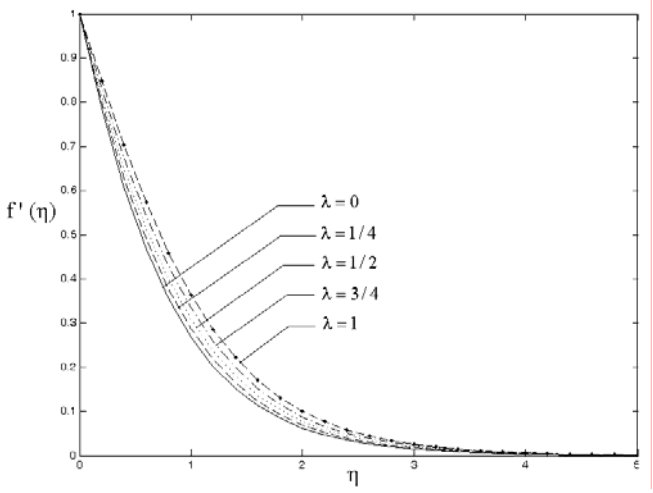

Fig. 5. HNNPSO solution for different value of λ as λ=0, 1/4, 1/2, 3/4, 1.

REFERENCES

[1] A.R. Sohouli, D. Domairry, M. Famouri, A. Mohsenzadeh. Analytical solution of natural convection of Darcian fluid about a vertical full cone embedded in porous media prescribed wall temperature by means of HAM. International Communications in Heat and Mass Transfer. 2008; 35: 1380–1384.

[2] D.A. Nield, A. Bejan, Convection in Porous Media, third ed., Springer-Verlag, New York, 2006.

[3] P. Cheng, I.D. Chang, On buoyancy induced flows in a saturated porous medium adjacent to impermeable horizontal surfaces, International Journal Heat Mass Transfer. 1976; 19: 1267–1272.

[4] J. H. Merkin, Free convection boundary layers in a saturated porous medium with lateral mass flux, International Journal Heat Mass Transfer. 1978; 21: 1499–1504.

[5] J. H. Merkin, Free convection boundary layers on axisymmetric and two-dimensional bodies of arbitrary shape in a saturated porous medium, International Journal Heat Mass Transfer. 1979; 22: 1461–1462.

[6] R.H. Nilson, Natural convective boundary layer on two-dimensional and axisymmetric surfaces in high-Pr fluids or in fluid saturated porous media, ASME J. Heat Transfer. 1981; 103: 803–807.

[7] P. Cheng, T.T. Le, I. Pop, Natural convection of a Darcian fluid about a cone, International Communications Heat Mass Transfer. 1985; 12: 705-717.

[8] I. Pop, T. Y. Na, Naturnal convection of a Darcian fluid about a wavy cone, International Communications Heat Mass Transfer. 1994; 21: 891-899.

[9] I. Pop, P. Cheng, An integral solution for free convection of a Darcian fluid about a cone with curvature effects, International Communications Heat Mass Transfer. 1986; 13: 433–438.

[10] Ching-Yang Cheng, An integral approach for heat and mass transfer by natural convection from truncated cones in porous media with variable wall temperature and concentration, International Communications Heat Mass Transfer. 2000; 27: 537-548.

[11] K.A. Yih, The effect of uniform lateral mass flux on free convection about a vertical cone embedded in a saturated porous medium, International Communications Heat Mass Transfer. 1997; 24: 1195–1205.

[12] I. Pop, T. Y. Na, Natural convection over a frustum of a wavy cone in a porous medium, Mech. Res. Comm. 1995; 22: 181–190.

[13] A. Malek, R.S. Beidokhti, Numerical solution for high order differential equations using a hybrid neural network—Optimization method. Applied Mathematics and Computation. 2006; 183: 260-271.

[14] H. Lee, I.S. Kang, Neural algorithms for solving differential equations, Journal of Computational Physics 1990; 91: 110–131.

[15] A.J. Meade Jr, A.A. Fernandez, The numerical solution of linear ordinary differential equations by feedforward neural networks, Mathematical and Computer Modelling. 1994; 19 (12): 1–25.

[16] I.E. Lagaris, A. Likas, D.I. Fotiadis. Artificial neural networks for solving ordinary and partitial differential equations. IEEE Transactions on Neural Networks. 1998; 9 (5): 987–1000.

[17] J.A. Khan, R.M.A. Zahoor, I.M. Qureshi, Swarm intelligence for the problem of non-linear ordinary differential equations and its application to well known Wessinger's equation. European Journal of scientific research. 2009; 34(4): 514-525.

[18] Z.Y. Lee, Method of bilaterally bounded to solution blasius equation using particle swarm optimization. Applied Mathematics and Computation 2006; 179: 779–786.

[19] D.T. Pham, E. Koc, A. Ghanbarzadeh, S. Otri. Optimisation of the weights of multi-layered perceptrons using the bees algorithm. Proceedings of 5th International Symposium on Intelligent Manufacturing Systems. Sakarya University, Department of Industrial Engineering, 2006; pp. 38–46, May 29–31.

[20] A.S. Yilmaz, Z. Ozer,. Pitch angle control in wind turbines above the rated wind speed by multi-layer percepteron and Radial basis function neural networks. Expert Systems with Applications. 2009; 36: 9767–9775.

[21] Pham, D.T., Liu, X. Neural Networks for Identification, Prediction and Control. Springer Verlag, London. 1995.

[22] R.P. Lippmann, An introduction to computing with neural nets, IEEE ASSP Magazine 1987; 4–22.

[23] K. Hornick, M. Stinchcombe, H. white, Multilayer feedforward networks are universal approximators, Neural Networks 1989; 2 (5): 359–366.

[24] M.A. Behrang, E. Assareh, A. Ghanbarzadeh, A.R. Noghrehabadi. The potential of different artificial neural network (ANN) techniques in daily global solar radiation modeling based on meteorological data. Solar Energy 2010; 84: 1468–1480.

[25] Kennedy J, Eberhart R. Particle swarm optimization. Proc Neural Networks. Proceedings,vol.1944. IEEE International Conference on, 1995. p.1942-1948.

[26] Engelbrecht A P. Fundamentals of computational swarm intelligence. Hoboken, N.J.: Wiley, 2005.

[27] Brits R, Engelbrecht AP, van den Bergh F. Locating multiple optima using particle swarm optimization. Applied Mathematics and Computation 2007; 189(2): 1859-1883.

[28] Liu X, Liu H, Duan H. Particle swarm optimization based on dynamic niche technology with applications to conceptual design. Advances in Engineering Software 2007; 38(10): 668-676.

[29] Pan H, Wang L, Liu B. Particle swarm optimization for function optimization in noisy environment. Applied Mathematics and Computation 2006; 181(2): 908-919.

[30] Yang IT.Performing complex project crashing analysis with aid of particle swarm optimization algorithm. International Journal of Project Management 2007; 25(6): 637-646.

[31] Shi Y, Eberhart R. A modified particle swarm optimizer. Proceedings of the IEEE International Conference on Evolutionary Computation. Anchorage, Alaska, 1998a. p.69-73.

[32] Shi Y, Eberhart R. Parameter selection in particle swarm optimization. Proceedings of the Seventh Annual Conference on Evolutionary Programming. New York, 1998b. p.591-600.

[33] E. Assareh, M.A.Behrang, M.R.Assari, A.Ghanbarzadeh. Application of particle swarm optimization (PSO) and genetic algorithm (GA) techniques on demand estimation of oil in Iran. Energy 35 (2010) 5223- 5229.

[34] M.A. Behrang., E. Assareh, M.R. Assari, M.R., and A. Ghanbarzadeh. Assessment of electricity demand in Iran's industrial sector using different intelligent optimization techniques. Applied Artificial Intelligence 2011; 25: 292–304. doi:10.1080/08839514.2011.559572

[35] M.A. Behrang, E. Assareh, A.R. Noghrehabadi, and A. Ghanbarzadeh. New sunshine-based models for predicting global solar radiation using PSO (particle swarm optimization) technique. Energy 2011; 36: 3036-3049. doi:10.1016/j.energy.2011.02.048.