J. Basic. Appl. Sci. Res., 3(7)10-16, 2013
© 2013, TextRoad Publication

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Numerical Solution to Troesch's Problem Using Hybrid Heuristic Computing

Suheel Abdullah Malik[a], Ijaz Mansoor Qureshi[b], Muhammad Zubair[a], Muhammad Amir[a]

[a] Department of Electronic Engineering, Faculty of Engineering and Technology, International Islamic University, Islamabad, Pakistan
[b] Department of Electrical Engineering, Air University, Islamabad, Pakistan

ABSTRACT

In this paper an evolutionary computational technique, which is stochastic in nature, is applied for the first time to the numerical solution of the Troesch's problem. An approximate mathematical model employing the linear combinations of log sigmoid basis functions is deduced. A fitness function (FF), containing unknown adjustable parameters and representing the trial solution along with the initial conditions, is developed. The Genetic algorithm (GA) as global optimizer is hybridized with local optimizers, such as Pattern Search (PS) and Interior Point algorithm (IPA), for achieving the unknown adjustable parameters. Comparisons of the numerical results obtained for three special cases of the Troesch's problem are made with some important deterministic classical approximation methods, as well as, the exact solutions. The numerical results are found to be in sharp agreement with the exact solutions.

KEYWORDS*:* Troesch's Problem; Boundary Value Problem; Genetic Algorithm (GA), Interior point Algorithm (IPA), Pattern Search (PS), Evolutionary Computing (EC)

1. INTRODUCTION

Nonlinear ordinary differential equations (ODEs) frequently arise in a wide variety of problems in applied science and engineering. These problems are by and large formulated as initial and/or boundary value problems. Traditionally these nonlinear boundary value problems (BVPs) are solved by the approximate analytical and numerical methods such as homotopy perturbation method (HPM) [1], homotopy analysis method (HAM) [2], variational iteration method (VIM) [3], adomian decomposition method (ADM) [4], shooting method [5] etc.

This paper is devoted primarily to the investigation of the Troesch's problem [1-8]

$$u''(x) = \lambda \sinh(\lambda\, u(x)) \quad 0 \le x \le 1 \qquad (1)$$

with the boundary conditions $u(0) = 0, and\ u(1) = 1$

Troesch's problem appears in the investigation of the confinement of a plasma column by radiation pressure and applied physics [1-9]. This BVP was formulated and solved by Weibel [10]. Troesch's obtained the numerical solution of this problem by the shooting method [11]. The Troesch's problem has been extensively studied and various methods have been proposed for its numerical solution. Hassan et al. [2] applied homotopy analysis method (HAM) to the Troesch's problem for its numerical solution. Mirmoradi et al. [1] found the solution of Troesch's problem using homotopy perturbation method (HPM). Deeba et al. [4] employed decomposition based method (ADM) for the approximate solution of this problem. Recently Geng and Cui [8] proposed a method based on the combination of ADM and reproducing kernel method (RKM) for the numerical solution of the Troesch's problem. Several other authors have obtained numerical solution of the Troesch's problem by utilizing various approximate methods such as modified homotopy perturbation method (MHPM) [7], He's polynomials [6], B-spline collocation [9], and simple shooting method [5].

Although a rich variety of approximate numerical techniques have been proposed for handling nonlinear ODEs, an incredible amount of research work is still carried out in this direction. Besides the traditional numerical techniques, stochastic solvers based on evolutionary computing and neural networks (NN) have been successfully applied to nonlinear ODEs. The efficiency and reliability of these stochastic solvers has been demonstrated by several authors [12-15]. Khan et al. [12] used a neural network (NN) model optimized by hybrid Particle Swarm Optimization (PSO) for the solution of nonlinear ODEs including the Wessinger's equation. Behrang et al. [13] solved a nonlinear differential equation arising from similarity solution of inverted cone embedded in porous medium employing a PSO based neural network (NN). Malik et al. [14] obtained the numerical solution of Duffing van der pol equation using heuristic computing technique. Zahoor et al. [15] applied hybrid evolutionary computing approach for the solution of fractional order Riccati differential equation.

The main goal of this study is to obtain the approximate numerical solution of the Troesch's problem (1) using hybrid heuristic computing approach. Genetic algorithm (GA), pattern search (PS), interior point algorithm (IPA), and two hybrid schemes called as GA-PS (hybridization of GA and PS), and GA-IPA (hybridization of GA and IPA) have been utilized in this study. The efficiency and reliability of the proposed method are demonstrated by solving the Troesch's problem for three special cases of the constant λ $(0.5, 1, and\ 10)$. Comparisons of the numerical results are made with the exact solutions and the classical approximate techniques.

***Corresponding Author:** Suheel Abdullah Malik, Department of Electronic Engineering, Faculty of Engineering & Technology, International Islamic University Islamabad, Pakistan, Email: suheel.abdullah@iiu.edu.pk

The remaining paper is organized as follows: In section 2, proposed methodology is briefly discussed. In section 3, we give brief overview of heuristic search techniques such as Genetic algorithm (GA), Pattern Search (PS), and Interior Point algorithm (IPA). Numerical results and their discussion are presented in section 4. Finally concluding remarks and future work are given in section 5.

2. DESCRIPTION OF THE PROPOSED METHOD

To obtain the approximate numerical solution of the Troesch's problem (1) using the proposed method, we may assume that the solution $u(x)$ and its first and second derivatives, $\dot{u}(x)$, $and\ \ddot{u}(x)$ are a linear combinations of some basis functions which can be expressed as follows.

$$u(x) = \sum_{i=1}^{m} \alpha_i\, \varphi(\gamma_i x + \beta_i) \tag{2}$$

$$\dot{u}(x) = \sum_{i=1}^{m} \alpha_i \gamma_i\, \dot{\varphi}(\gamma_i x + \beta_i) \tag{3}$$

$$\ddot{u}(x) = \sum_{i=1}^{m} \alpha_i {\gamma_i}^2\, \ddot{\varphi}(\gamma_i x + \beta_i) \tag{4}$$

where α_i , γ_i , $and\ \beta_i$ are real valued unknown adjustable parameters, m is the number of basis functions, and $\varphi(x)$ is assumed to be the log sigmoid function which is given by

$$\varphi(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

The approximate numerical solution $u(x)$ of the Troesch's problem is obtained from (3) once the optimal values of the unknown adjustable parameters are attained. The optimal values of these unknown adjustable parameters are determined by formulating a fitness function (FF) which consists of the sum of two parts. First part represents the mean square error associated with the given ODE (ε_1) and second part represents the mean square error associated with the given boundary conditions(ε_2), which are given by (6) and (7) respectively.

$$\varepsilon_1 = \frac{1}{m+1} \sum_{i=0}^{m} [\ddot{u}(x_i) - \lambda \sin(\lambda u(x_i))]^2 \tag{6}$$

$$\varepsilon_2 = \frac{1}{2}\{\left(u(0)\right)^2 + (u(1) - 1)^2\} \tag{7}$$

where $u(x)$, $and\ \ddot{u}(x)$ are given by (2) , and (4) respectively.

Therefore the fitness function (FF) is given by

$$\varepsilon_j = \varepsilon_1 + \varepsilon_2 \tag{8}$$

where j is the cycle index.

The FF function given by (8) contains the unknown adjustable parameters $(\alpha_i, \gamma_i, \beta_i)$ given in (2) to (4). The minimization of FF is performed by heuristic optimization methods. The optimal values of the unknown adjustable parameters are acquired corresponding to the minimum FF. Consequently the approximate numerical solution $u(x)$ of the problem is obtained using (3).

3. HEURISTIC OPTIMIZATION TECHNIQUES

In last few decades Evolutionary computational techniques such as Genetic algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE) etc. have been pursued vigorously. These techniques represent powerful search and optimization paradigm which are based on the principle of natural evolution. GA originally discovered by J. H. Holland in 1970 is among these techniques which have been most widely used due to its simplicity and robustness. GA is a global stochastic search algorithm inspired by the mechanism of genetic inheritance and natural selection. The GA operates on a population of individual solutions. The algorithm evolves population toward the best solution over the successive generations by using three primary operations selection, crossover, and mutation [16].

The Interior Point Algorithm (IPA) also called as barrier method is a local search method. The method attempts to solve Karush-Kuhn-Tucker (KKT) equations at each iteration. The algorithm arrives at the best solution of the problem by computing iterates that lie in the interior of the feasible region while reducing the barrier parameter [17].

The Pattern Search (PS) belongs to the direct search optimization methods. The algorithm does not involve the gradient of the objective function. The prime objective of PS is to compute a sequence of points that reach an optimal point. At each step the algorithm finds a set of points called a mesh around the optimal point of previous step. The mesh can be obtained by adding the current point to a scalar multiple of vectors called a pattern. The new point becomes the current point in the next step of the algorithm, if the PS establishes that the point in the mesh improves the objective function at the current point. The PS method possesses flexibility for local search enhancement [18, 19].

In many problems the performance of GA is drastically improved when it is hybridized with local search methods such as Pattern Search (PS) and Interior Point algorithm (IPA) [20]. In this work GA has been hybridized with PS and IPA. The two schemes resulting from the hybridization are called GA-PS and GA-IPA. In these hybrid schemes GA has been utilized as global optimizer while PS and IPA have been used for local search fine-tuning. The parameter settings of these algorithms used in this work are given in Table 1. The procedural steps of the proposed heuristic hybrid schemes are given as follows.

Algorithm 1: GA hybridized with PS and IPA
Step 1: (Initialization of Population) Generate a population of N chromosomes (individual) at random using random number generator. Each chromosome consists of M number of genes. The number of genes is equal to the number of unknown adjustable parameters. Step 2: (Evaluation of Fitness) Determine the fitness of each individual in the current population using the fitness function (FF) of the given problem. Sort these individuals according to their fitness values. Step 3: (Stopping Criteria) The termination criteria set by the use are checked. The algorithm terminates if the maximum number of cycles (generations) has surpassed or a predefined fitness value is attained. If the stopping criterion is satisfied then go to step 6 for local search fine-tuning, else continue and repeat steps 2 to 5. Step 4: (Selection of Parents and Production of Offspring) Populate a new generation using the crossover operation. Parents are selected on the basis of their fitness which produces offspring (children) to act as parents for the next generation. Step 5: Mutation If there is no improvement in the fitness in the new generation then mutation operation is performed. Step 6: (Local Search Fine-tuning) The best chromosome acquired by the GA is given as a start point to the PS and IPA for fine tuning and improvement.

Table 1. Parameter Settings of algorithms

GA		**PS**		**IPA**	
Parameters	**Settings**	**Parameters**	**Settings**	**Parameters**	**Settings**
Population size	240	Start point	Optimal values from GA	Start point	Optimal values from GA
Chromosome size	30	Poll method	GPS positive basis 2N	Maximum iterations	1000
Selection function	Stochastic uniform	Polling order	consecutive	Maximum function evaluations	150000
Mutation function	Adaptive feasible	Maximum iterations	3000	Function tolerance	1e-18
Crossover function	Heuristic	Maximum function evaluation	150000	Derivative type	Central differences
Hybridization	PS/IPA	Function tolerance	1e-18	Hessian	BFGS
No. of generations	1000				
Function tolerance	1e-18				
Bounds	-20, +20				

4. NUMERICAL RESULTS AND DISCUSSION

In this section the proposed methodology is applied to the Troesch's problem (1). To prove the efficacy and reliability of the proposed method, comparisons of the numerical results are made with the exact solutions and somel classical approximate numerical methods including HPM [1], HAM [2], VIM [3], ADM [4], combined ADM-RKM [8], and cubic B-spline collocation method [9].

Example: We consider the Troesch's problem (1) and obtain its numerical solution for three special cases of the constant λ. ($\lambda = 0.5, 1, and\ 10$). The closed form solution of (1) is given by [1-4]

$$u(x) = \frac{2}{\lambda} sinh^{-1}\left\{\frac{\dot{u}(0)}{2}\, sc\left(\lambda x | 1 - \frac{1}{4}(\dot{u}(0))^2\right)\right\} \tag{9}$$

The approximate numerical solution of (1) using the proposed method is obtained by formulating its fitness function (FF) for each case of λ. The FF for the Troesch's problem by taking 10 number of basis function following from (6), (7), and (8) for λ = 0.5, λ = 1, and λ = 10 are given respectively as follows.

$$\varepsilon_j = \frac{1}{11}\sum_{i=1}^{11}(\ddot{u}(x_i) - 0.5 sinh(0.5u(x_i)))^2 + \frac{1}{2}\{(u(0))^2 + (u(1)-1)^2\}\Big|_j \tag{10}$$

$$\varepsilon_j = \tfrac{1}{11}\textstyle\sum_{i=1}^{11}(\ddot{u}(x_i) - 1 sinh(1u(x_i)))^2 + \tfrac{1}{2}\{(u(0))^2 + (u(1)-1)^2\}\Big|_j \tag{11}$$

$$\varepsilon_j = \frac{1}{11}\sum_{i=1}^{11}(\ddot{u}(x_i) - 10sinh(10u(x_i))^2 + \frac{1}{2}\{(u(0))^2 + (u(1)-1)^2\}\Bigg|_j \quad (12)$$

where $u(x)$ and $\ddot{u}(x)$ are given by (2) and (4) respectively.
The FFs given by (10), (11), and (12) are minimized by applying heuristic algorithms (GA, PS, IPA, GA-PS, and GA-IPA) for the learning of the unknown adjustable parameters (chromosome). For simulations Matlab 7.6.0. has been utilized.

The parameter settings for the execution of the optimization algorithms (GA, PS, IPA, GA-PS, and GA-IPA) are given in table 1. The input of the training set is taken from $x \epsilon\{0, 0.1, 0.2, 0.3, \ldots, 1\}$, thus the number of unknown adjustable parameter is equal to 30. The values of these unknown adjustable parameters are bounded between -20 and + 20. This was observed by numerous simulations that by restricting these unknown adjustable parameters to the specified interval we get good results.

The algorithms are executed according to the prescribed settings to achieve the minimum fitness value. The optimal values of the unknown adjustable parameters achieved by the hybrid heuristic schemes GA-PS, and GA-IPA are provided in Table 2.

Table 2. Unknown adjustable parameters achieved by hybrid schemes

		GA-PS			GA-IPA		
	i	α_i	γ_i	β_i	α_i	γ_i	β_i
	1	1.7158681603	1.5582360608	2.0631516835	2.7198348764	1.3023129579	2.7029889471
	2	-4.6783123067	-0.6687671519	1.4863655665	-6.8852582037	-0.6604649375	1.9211891289
	3	4.6091919363	0.2025176487	6.6903083812	6.8809590700	-0.7872728472	11.0064068543
	4	-0.5712269106	-1.3301003001	0.2897717189	-0.7906217762	-1.0790102116	-0.3645472097
λ=0.5	5	0.2863556257	0.0082082091	-1.4850131696	0.4717450376	0.1717571573	-2.2140470400
	6	1.2684808004	-0.1755886913	-1.5029140094	1.8253820009	-0.0012649339	-2.1873627814
	7	-1.8016237414	-1.5304456849	4.1613531171	-2.7283916096	-1.8210668340	6.7238448765
	8	0.6953501560	0.1337606705	-4.5161362820	1.7634657011	-0.8101186300	-7.4410117915
	9	-0.7450129556	-0.9923187515	3.3092241128	-1.0727261001	-1.5058716249	4.8666389823
	10	0.2801029005	-0.0254306260	1.2470413527	0.5266243902	0.2230996855	1.8974740978
	1	-1.9309001091	-0.7994681355	8.8558759767	-0.1372666353	-0.6229004681	6.4797099210
	2	-15.3919214990	-4.2390295361	-12.5779321536	2.6492265274	-2.7797553698	-10.1326830246
	3	19.9965010243	-8.8608017612	-12.9704722149	4.2094573306	-3.6318530927	-8.0170417892
	4	-17.9738660200	-19.9924471647	-15.7950512209	2.3399625238	1.3419376584	-3.9916138361
λ=1	5	12.9643626937	19.9924255508	15.6780419344	4.6751062804	2.8041306261	6.6193982214
	6	1.3738632820	3.5674960295	-7.2071047428	7.6424167162	2.8620847910	-7.5088232997
	7	1.4539622667	-1.4889229888	-3.9447580404	1.6333885989	0.0462620683	-2.6546172253
	8	-19.1932911641	-19.9962351134	-16.0577851074	0.2192117143	1.6716899409	-4.6563273492
	9	-11.0563796100	-1.0704409829	3.2560971518	-4.6508777867	-1.0621137337	2.4865115887
	10	-13.7739752422	-1.1366687668	-3.4685363787	-8.3942122341	-1.0788478894	-3.0064202662
	1	5.817776659	12.71542432	-17.18887722	1.341540516	0.203313267	-1.584276599
	2	19.11176207	5.036865426	-13.21874064	-1.205739356	0.358499171	-1.381071609
	3	2.461030913	-8.061608453	12.83352698	1.250597207	-16.99488461	17.06886302
	4	2.763894541	-0.957008411	7.773068366	1.389950712	3.662438546	3.379454408
λ=10	5	9.874208755	-11.34018211	17.30807435	1.358238427	2.538755228	3.158011208
	6	-19.98589831	10.75103145	-17.25642848	0.671209377	3.7426714	-2.366951517
	7	-12.51605473	-10.97068987	15.88135324	-5.845832642	-12.57777619	14.18877037
	8	5.06650745	5.50988936	-10.82412141	4.551042901	0.236032429	-3.717364387
	9	8.690680897	-2.733654226	-8.518343425	2.637267387	-0.765251666	-2.88903648
	10	-2.583343765	-4.624982016	10.63471015	1.842832969	-1.82007665	2.150456676

The approximate numerical solution $u(x)$ of the Troesch's problem (1) is obtained by substituting the optimal values of unknown adjustable parameters in (3). The approximate numerical results obtained by the proposed schemes are presented in Table 3 for $\lambda = 0.5, 1$ and Table 4 for $\lambda = 10$. In Table 5 we also present numerical results for $\lambda = 10$ obtained by some classical methods given in [8].

The absolute errors have been calculated relative to the exact solution and these are presented in Table 6, Table 7, and Table 8 for $\lambda = 0.5$, $\lambda = 1$, and $\lambda = 10$ respectively. For the efficiency, accuracy, and reliability of the proposed method comparisons are made with the exact solutions and some classical approximate numerical methods such as HPM [1], HAM [2], VIM [3], ADM [4], combined ADM-RKM [8], and cubic B-spline collocation method [9]. The comparison of the absolute errors from Table 6 and Table 7 and Table 8 clearly reveals that the proposed method yields the results of the Troesch's problem (1) for three special cases $\lambda = 0.5$, $\lambda = 1, and\ \lambda = 10$ with greater accuracy. From the comparison of Table 6 for $\lambda = 0.5$ it is evident that the proposed method yields the approximate results similar to the classical methods HPM [1], HAM [2], ADM [4] and B-spline [9], while the proposed method gives significantly better results than VIM [3]. Moreover for $\lambda = 1$, the proposed method gives comparable results to the methods given in [4, 9], slightly better than HPM [1] and significantly better than VIM [3]. From the comparison of Table 8 for $\lambda = 10$ for which the problem is difficult to be solved [8] the proposed method has achieved the numerical results with fairly good accuracy and quite comparable to ADM-RKM [8]. Further it is observed from the comparison of approximate solution of Troesch's problem for $\lambda = 10$ that the absolute errors relative to the exact solutions obtained from the proposed method are considerably smaller than the classical methods ADM, VIM, and MHPM.

Table 3. Numerical results of Troesch's problem using proposed method for λ=0.5 and λ=1

	x	Exact[1-4]	GA	PS	IPA	GA-PS	GA-IPA
	0.1	0.0951769020	0.0959526958	0.0958153064	0.0959444555	0.0959444449	0.0959443773
	0.2	0.1906338691	0.1921346585	0.1920184778	0.1921287750	0.1921283944	0.1921287413
	0.3	0.2866534030	0.2887980948	0.2887161527	0.2887942822	0.2887935126	0.2887943631
	0.4	0.3835229288	0.3861875502	0.3861177986	0.3861846919	0.3861843534	0.3861848222
λ=0.5	0.5	0.4815373854	0.4845498566	0.4844633973	0.4845471190	0.4845478455	0.4845471807
	0.6	0.5810019749	0.5841357919	0.5840266020	0.5841333437	0.5841349006	0.5841332885
	0.7	0.6822351326	0.6852024912	0.6850922717	0.6852012756	0.6852027564	0.6852011751
	0.8	0.7855717867	0.7880157994	0.7879305298	0.7880165464	0.7880172754	0.7880165161
	0.9	0.8913669875	0.8928518696	0.8927922260	0.8928541458	0.8928545292	0.8928541952
	0.1	0.0817969966	0.084686933	0.0846764683	0.0846616547	0.0846569098	0.0846610966
	0.2	0.1645308709	0.170193466	0.1701657627	0.1701715056	0.1701683195	0.1701708205
	0.3	0.2491673608	0.257410747	0.2574019571	0.2573934088	0.2573911869	0.2573917712
λ = 1	0.4	0.3367322092	0.347235463	0.3472547732	0.3472220283	0.3472208650	0.3472197256
	0.5	0.4283471610	0.440610797	0.4406227777	0.4405993798	0.4406002701	0.4405972632
	0.6	0.5252740296	0.538545435	0.5384912500	0.5385347088	0.5385382430	0.5385333420
	0.7	0.6289711434	0.642138934	0.6419805108	0.6421293058	0.6421345857	0.6421285075
	0.8	0.7411683782	0.752615188	0.7523736164	0.7526083064	0.7526134176	0.7526074131
	0.9	0.8639700206	0.871365585	0.8711200368	0.8713620340	0.8713662992	0.8713607069

Table 4. Numerical results of Troesch's problem by the proposed method for λ= 10

x	Exact [8]	GA	PS	IPA	GA-PS	GA-IPA
0.1	0.0000763	-0.0001750	0.0689290	0.0006507	0.0000959	-0.0022146
0.2	0.0001299	-0.0000617	-0.0250253	0.0003901	-0.0001366	-0.0008775
0.3	0.0003589	0.0000972	-0.0665243	0.0001604	-0.0001633	-0.0005250
0.4	0.0009779	0.0003808	-0.0671847	0.0006313	0.0001016	0.0002556
0.5	0.0026590	0.0009327	-0.0374145	0.0013073	0.0008483	0.0018240
0.6	0.0072289	0.0021076	0.0136070	0.0039797	0.0025192	0.0043090
0.7	0.0196640	0.0048757	0.0779259	0.0096452	0.0062620	0.0104308
0.8	0.0537303	0.0120360	0.1487887	0.0306525	0.0154046	0.0300570
0.9	0.1521140	0.0318843	0.2206004	0.0640741	0.0403545	0.0801700

Table 5. Numerical results of Troesch's problem by classical methods given in [8] for λ= 10

x	ADM-RKM [8]	VIM [8]	ADM [8]	MHPM [8]
0.1	0.0000576	0.1186109866	667081.1874	17.61750
0.2	0.0001902	0.4461962517	1333955.1189	33.69333
0.3	0.0005676	3.8003366781	1999860.1189	46.78583
0.4	0.0016654	79.89147273	2661970.7366	55.65333
0.5	0.0048331	1880.3539472	3310585.4201	59.35417
0.6	0.0137488	41642.365193	3914127.8659	57.34667
0.7	0.0374013	878764.64189	4374578.5342	49.58917
0.8	0.0936540	18064027.967	4406724.4178	36.64000
0.9	0.2189270	366613074.02	3290268.6374	19.75750

Table 6. Comparison of absolute errors between proposed method and classical methods given in [1-4, 9] for λ= 0.5.

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
GA	7.758E-04	1.501E-03	2.145E-03	2.665E-03	3.012E-03	3.134E-03	2.967E-03	2.444E-03	1.485E-03
PS	6.384E-04	1.385E-03	2.063E-03	2.595E-03	2.926E-03	3.025E-03	2.857E-03	2.359E-03	1.425E-03
IPA	7.676E-04	1.495E-03	2.141E-03	2.662E-03	3.010E-03	3.131E-03	2.966E-03	2.445E-03	1.487E-03
GA-PS	7.675E-04	1.495E-03	2.140E-03	2.661E-03	3.010E-03	3.133E-03	2.968E-03	2.445E-03	1.488E-03
GA-IPA	7.675E-04	1.495E-03	2.141E-03	2.662E-03	3.010E-03	3.131E-03	2.966E-03	2.445E-03	1.487E-03
HPM [1]	7.711E-04	1.502E-03	2.151E-03	2.674E-03	3.023E-03	3.144E-03	2.977E-03	2.453E-03	1.492E-03
HAM [2]	7.677E-04	1.495E-03	2.142E-03	2.663E-03	3.011E-03	3.133E-03	2.968E-03	2.446E-03	1.488E-03
VIM [3]	4.865E-03	9.700E-03	1.447E-02	1.915E-02	2.370E-02	2.808E-02	3.223E-02	3.611E-02	3.964E-02
ADM [4]	7.615E-04	1.484E-03	2.127E-03	2.646E-03	2.993E-03	3.115E-03	2.952E-03	2.434E-03	1.481E-03
Spline[9]	7.671E-04	1.494E-03	2.140E-03	2.660E-03	3.009E-03	3.130E-03	2.965E-03	2.443E-03	1.486E-03

Table 7. Comparison of absolute errors between proposed method and classical methods given in [1-4, 9] for λ= 1.

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
GA	2.890E-03	5.663E-03	8.243E-03	1.050E-02	1.226E-02	1.327E-02	1.317E-02	1.145E-02	7.396E-03
PS	2.879E-03	5.635E-03	8.235E-03	1.052E-02	1.228E-02	1.322E-02	1.301E-02	1.121E-02	7.150E-03
IPA	2.865E-03	5.641E-03	8.226E-03	1.049E-02	1.225E-02	1.326E-02	1.316E-02	1.144E-02	7.392E-03
GA-PS	2.860E-03	5.637E-03	8.224E-03	1.049E-02	1.225E-02	1.326E-02	1.316E-02	1.145E-02	7.396E-03
GA-IPA	2.864E-03	5.640E-03	8.224E-03	1.049E-02	1.225E-02	1.326E-02	1.316E-02	1.144E-02	7.391E-03
HPM [1]	1.415E-02	2.760E-02	3.964E-02	4.946E-02	5.621E-02	5.887E-02	5.624E-02	4.686E-02	2.889E-02
HAM [2]	2.876E-03	5.665E-03	8.263E-03	1.054E-02	1.231E-02	1.333E-02	1.237E-01	1.152E-02	7.455E-03
VIM [3]	1.837E-02	3.681E-02	5.537E-02	7.411E-02	9.303E-02	1.121E-01	1.312E-01	1.501E-01	1.685E-01
ADM [4]	2.452E-03	4.900E-03	7.247E-03	9.354E-03	1.105E-02	1.209E-02	1.211E-02	1.062E-02	6.939E-03
Spline[9]	2.858E-03	5.629E-03	8.210E-03	1.047E-02	1.223E-02	1.323E-02	1.313E-02	1.142E-02	7.379E-03

Table 8. Comparison of absolute errors between proposed method and classical methods given in [8] for λ= 10

	Proposed Method					Other Methods			
x	**GA**	**PS**	**IPA**	**GA-PS**	**GA-IPA**	**ADM-RKM [8]**	**ADM [8]**	**MHPM [8]**	**VIM [8]**
0.1	2.513E-04	6.885E-02	5.744E-04	1.961E-05	2.291E-03	1.871E-05	6.671E+05	6.671E+05	1.185E-01
0.2	1.916E-04	2.516E-02	2.602E-04	2.665E-04	1.007E-03	6.030E-05	1.334E+06	1.334E+06	4.461E-01
0.3	2.617E-04	6.688E-02	1.985E-04	5.222E-04	8.839E-04	2.087E-04	2.000E+06	2.000E+06	3.800E+00
0.4	5.971E-04	6.816E-02	3.466E-04	8.763E-04	7.223E-04	6.875E-04	2.662E+06	2.662E+06	7.989E+01
0.5	1.726E-03	4.007E-02	1.352E-03	1.811E-03	8.350E-04	2.174E-03	3.311E+06	3.311E+06	1.880E+03
0.6	5.121E-03	6.378E-03	3.249E-03	4.710E-03	2.920E-03	6.520E-03	3.914E+06	3.914E+06	4.164E+04
0.7	1.479E-02	5.826E-02	1.002E-02	1.340E-02	9.233E-03	1.774E-02	4.375E+06	4.375E+06	8.788E+05
0.8	4.169E-02	9.506E-02	2.308E-02	3.833E-02	2.367E-02	3.992E-02	4.407E+06	4.407E+06	1.806E+07
0.9	1.202E-01	6.849E-02	8.804E-02	1.118E-01	7.194E-02	6.681E-02	3.290E+06	3.290E+06	3.666E+08

5. CONCLUSIONS AND FUTURE WORK

A heuristic computing based stochastic method has been successfully applied for the approximate numerical solution of the Troesch’s boundary value problem. On the basis of the numerical results and comparisons made with the classical approximate numerical methods and the exact solutions, it can be concluded that the proposed method is efficient and useful for solving the Troesch’s problem. The accuracy and the reliability of the proposed method are demonstrated by solving the Troesch’s problem with three special cases of constant . The proposed method gives remarkably accurate results of the Troesch’s problem which are quite similar to some of the classical approximate methods in comparison with the exact solutions. Furthermore the proposed method can provide the approximate numerical solution of the problem conveniently and on the continuous grid of time once the optimal values of the unknown adjustable parameters have been attained.

In future we aim to apply heuristic computational techniques to various other nonlinear ODEs, nonlinear coupled ordinary differential equations, and integro-differential equations appearing in the applications of applied science and engineering. Further other nature inspired techniques like Bee colony and Ant colony may be looked into as candidates for solving such nonlinear and even coupled differential equations.

REFERENCES

[1] Mirmoradi, S. H., I. Hosseinpour , S. Ghanbarpour and A. Barari, 2009. Application of an approximate analytical method to nonlinear Troesch’s problem. Appl. Math. Sci., 3(32): 1579 – 1585.

[2] Hassan, H. N. and M. A. El-Tawil, 2011. An efficient analytic approach for solving two-point nonlinear boundary value problems by homotopy analysis method. Math. Meth. Appl. Sci., 34: 977–989.

[3] Momani, S., S. Abuasad and Z. Odibat, 2006. Variational iteration method for solving nonlinear boundary value problems. Appl. Math. Comput., 183:1351-1358.

[4] Deeba, E., S. A. Khuri and S. Xie, 2000. An Algorithm for Solving Boundary Value Problems.J.Comput. Phy., 159: 125–138.

[5] Chang, S. H., 2010. Numerical solution of Troesch’s problem by simple shooting method. Appl. Math. Comput., 216: 3303–3306.

[6] Mohyu-din, S. T., 2011. Solution of Troesch’s problem using He’s polynomials. Revista De La, Union Mathematica Argentina, 52: 143–148.

[7] Feng, X., L. Mei and G. He, 2007. An e□cient algorithm for solving Troesch’s problem. Appl. Math. Comput., 189: 500–507.

[8] Geng, F. and M. Cui,2011. A novel method for nonlinear two-point boundary value problems: combination of ADM and RKM. Appl. Math. Comput., 217: 4676–4681.

[9] Khuri, S. A. and A. Sayfy,2011. Troesch's problem: A B-spline collocation approach. Math. Comput. Modeling, 54: 1907-1918.

[10] Weibel, E. S. and R. K. M. Landshoff, 1958. The Plasma in magnetic field. Stanford university press, Stanford, pp. 130.

[11] Troesch, B. A., 1976. A simple approach to a sensitive two-point boundary value problem. J. Comput. Phy., 21: 279—290.

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

[13] Behrang, M. A., M. Ghalambaz , E. Assareh and A. R. Noghrehabadi, 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. World Acad. Sci. Eng.. Tech., 49: 1098-1103.

[14] Malik, S. A., I. M. Qureshi, M. Zubair and I Haq, 2012. Solution to force-free and forced duffing-van der pol oscillator using memetic computing. J. Basic Appl. Sci. Res., 2(11): 11136-11148.

[15] Zahoor, R. M. A., J. A. Khan and I. M. Qureshi, 2009. Evolutionary computation technique for solving Riccati differential equation of arbitrary order. World Acad. Sci. Eng. Tech., 48: 303-309.

[16] Mitchell, M., 1995. Genetic algorithms: An overview. Complexity, 1: 31–39.

[17] Lesaja, G., 2009. Introducing interior-point methods for introductory operations research courses and/or linear programming courses. The Open Oper. Res. J., 3: 1-12.

[18] Al-Othman , A. K. and K. M. EL-Nagger, 2008. Application of pattern search method to power system security constrained economic dispatch. Int. J. Eng.. Appl. Sci., 4(1): 25-30.

[19] Torczon, V. 1997. On the convergence of pattern search algorithms. SIAM J. Opt., 7(1): 1-25.

[20] Garg, P., 2009. A comparison between memetic algorithm and genetic algorithm for the cryptanalysis of simplified data encryption standard algorithm. Int. J. Net. Sec. Appl., 1(1): 34-42.