Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 05 Issue 02
2023

Research Article
https://doi.org/10.56979/502/2023

An Efficient Approach for Solving Second Order or Higher Ordinary Differential Equations Using ANN

Muhammad Kashan Basit[1], Tahir Abbas Khan[2*], Jamshaid Iqbal Janjua[3,4], Asif Hussain[4], Hadi Abdullah[5], and Sadaqat Ali Ramay[2]

[1]Department of Computer Science, MNS-UET, Multan, 60000, Pakistan.
[2]Department of Computer Science, TIMES Institute, Multan, Pakistan.
[3]Al-Khawarizmi Institute of Computer Science, University of Engineering & Technology (UET), Lahore, Pakistan.
[4]School of Computer Sciences, National College of Business Administration & Economics, Lahore, Pakistan.
[5]Department of Computer Science, Lahore Garrison University, Lahore, 54000, Pakistan.
*Corresponding Author: Tahir Abbas Khan. Email: drtahirabbas@t.edu.pk.

Received: January 09, 2023 **Accepted:** September 12, 2023 **Published:** September 17, 2023.

Abstract: In this article, a computational technique called artificial neural networks (ANN's) merged with sequential quadratic programming (SQP). The feed forward neural networks are used which outperforms in an unsupervised manner. Activation function called Log-sigmoid is applied in the hidden layers of neural networks for the accuracy of results because it is more stable than any other activation function. This approach is best in terms of accuracy in solving linear and non-linear second or forth order ordinary differential equations. The main task is to implement the linear and nonlinear equations with their initial boundary conditions. This article also contains the application of the Jeffery fluid equation which is fourth order differential equation. Different cases of computational complexity in term of time and space are presented. Comparison of exact solution with the referenced techniques shows the correctness of the proposed scheme. Furthermore, the experimental results show the accuracy 99%-99.9% from the other techniques. This system also shows that it is stable at higher values while other systems deviate at higher values from the exact solution and remains unstable. In mathematics models, this technique outperforms to solve the linear and nonlinear differential equations. So, this will help the mathematicians and scientist to solve the higher order differential equations whom their solution does not exist.

Keywords: Jeffery Fluid, Artificial Neural Network, Numerical Approximation, Monte Carlo Simulations.

1. Introduction

A computational numerical technique is implemented for solving ODEs of second and fourth order. This system helps to solve those equations like Steller structure of earth, forth order or higher order of equations that are difficult to solve, or their solution does not exist. The equations contain function and the derivative of that function [1-3]. Sequential quadratic programming (SQP) optimized the weights that are used by the artificial neural networks to solve second order or higher order differential equations. Artificial neural networks and sequential quadratic programming both are combined to get the better results for the equations that cannot be solved in an unsupervised way. Many dynamical and classical systems can be solved by using artificial neural networks. Numerous applications in engineering and applied science including fluid mechanics, network synthesis, controls, diffusion problems, robust stabilization, stochastic theory, optimal filtering, and the mathematics involve the use of the differential equation [4-8].

It is difficult to solve the higher order ordinary differential equations which leads researchers to explore the solution of these equations. Many techniques are implemented to find the solution of these equations, but the accuracy and the implementation of such systems are restricted to some extent [9]. The solution of equations that are of linear or non-linear type is planned according to the architecture defined in Hopfield neural network. To provide the solution of the system the network's energy function is minimized in different order equations [10-14].

Other approach for solving these ordinary differential equations is based on the statistic that many different types of splines can be resultant by the super positioning of piecewise activation functions [15]. Wavelet scaling is the method that provide the differential equation solution, but the drawback of this solution is that it is affected by the ill matrices' conditions [16].

The comparison of the proposed method with other method is that the defined method outperforms in term of accuracy and efficiency at higher values and remain stable while many other methods become unstable, and it gives approximate same result as the exact solution.

To provide the solution of the differential equations, this work will motivate and provide easy and exact solution towards solving the differential equations. This method is compared with the power series neural networks (PSNN), adaptive PSNN [17.18]., Euler, Modified Euler, cosine neural networks and so on [19,20]. The accuracy of this systems is somehow can be improved due to which the computational burden can be minimized. The techniques based on artificial neural networks which is now extensively used in various chemistry, engineering and applied sciences as well.

2. Proposed Methodology

In different fields of mathematics second order differential equations are of great interest. Log-sigmoid function is suitable to be used as an activation function as it is the most stable function for solving linear and non-linear differential equations, it is mainly used and implemented in the hidden layer of neural network. In this context the following second derivative of log-sigmoid function is used.

$$\hat{y}(\mathrm{t}) = \sum_{\mathrm{i}=1} a_{\mathrm{i}} \frac{1}{1+\exp\left(-w_i + b_i\right)} \quad (1)$$

Where αi, wi and bi are the adaptive parameters which specify the range, the convergence and the shifting of the function. In this article the activation function is used in the hidden layer of artificial neural networks and is called log sigmoid function. The following equation is the basis of log sigmoid function.

$$\varphi(t) = \frac{1}{1+e^{-t}} \quad (2)$$

The sequential quadratic programming represents the nonlinear programing methods and is known as a local search algorithm. For nonlinear optimization an iterative method is used. Sequential quadratic programming has been tested on various methods and has shown good results in terms of efficiency, accuracy, and the percentage of success is better than other systems. SQP involves different steps to find out the unknown parameters of the activation function. These steps are discussed below:

First step is the tool activation while the tool setting criteria is given in the table 1. Second step is the program initialization in which the randomly initial boundary values are given according to the number of weights that are unknown at the time. Third step define to set the fitness evaluation function. The fourth step defined the termination criteria.

If any of the following given criteria defined in table 1 is occurred the system will terminate its execution and the result will be defined on the basis of predefined fitness value.

Fifth step is defined to first store the value for final weight calculation after this it will show the total computational time taken by the algorithm.

Table 1. Defining the Parameters for Sequential Quadratic Programming

Algorithm (SQP)	Values
Start / Initiation of Algorithm	Randomly defined between [15-15]
Variables defined	30
Total Iterations	1500
Hessian	BFGS
Derivation	Finite difference
Scaling Function	Objectives and constrains
Upper bound	15

Lower bound	-15
Value defined for fitness function	10^-18
Other	Default

3. Simulation and Results

In the following example, the ordinary differential equation is given in which its initial boundary condition values and first order and second order differentials are defined. The comparison is made by solving these equations with different techniques. The equation is solved by power series neural network using sign function. In this article log-sigmoid function is used with SQP-NN because it has shifting capabilities, it has convergence stabilities, shrinking and expansion capabilities. The domain is defined from 0 to 1 and the step size of 0.1.

3.1. Example

In this example, the differential equation of second order is given with the range from [0,1] and the initial conditions [21]:

$$y' = y - \frac{2x}{y} : x \in [0,1]$$
$$y(0) = 1; y'(0) = 1 \tag{3}$$

The estimated solution of the above second order differential equation is given as follows up to the sixth order term.

$$\hat{y}(x) = y(0) + y'(0)x + \sum_{k=2}^{6} w_k x^k \tag{4}$$

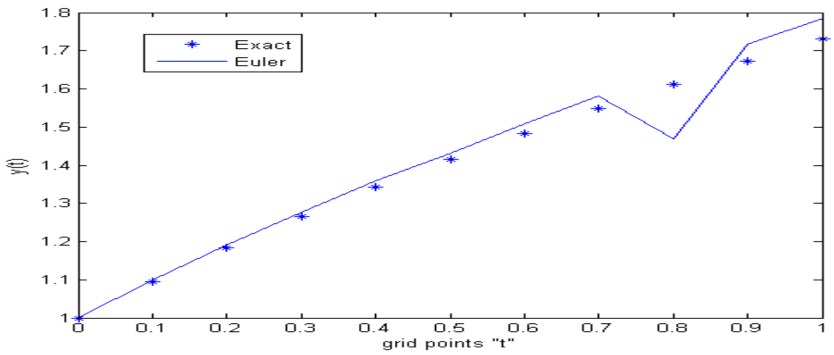

Figure 1. Euler method compared with exact method for [0-1] boundary values

Figure 1 shows the difference between the Euler and the exact solution of the given equation. There is a huge difference between the exact and the Euler method defined by John H. Mathews et al. It can be observed that only one initial value is matched at step size 0.1 while the rest of the values are above or below the exact values and at the higher values like at step size 0.9 the system deviates from the exact solution and become unstable.

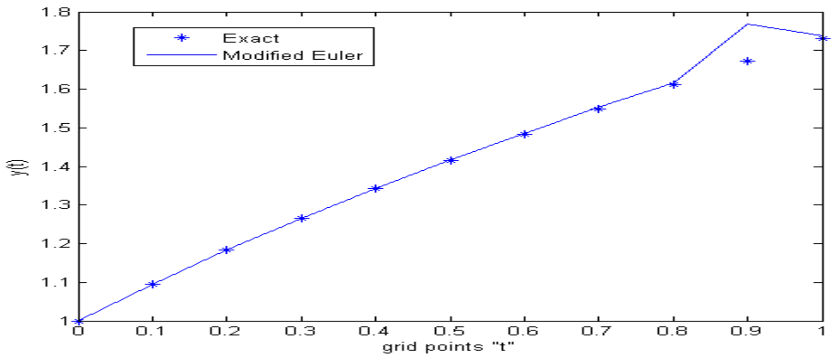

Figure 2. M-Euler method compared with exact method for [0-1] boundary values

Figure 2 shows the difference between the Modified-Euler and the exact solution of the given equation. The result of M-Euler is much better than the Euler method but still at higher value of step size 0.9 and above the Modified-Euler does not remain stable. Similar behavior has been observed by Richard L. Burden and his co-researcher, the literature reveals the same behavior.

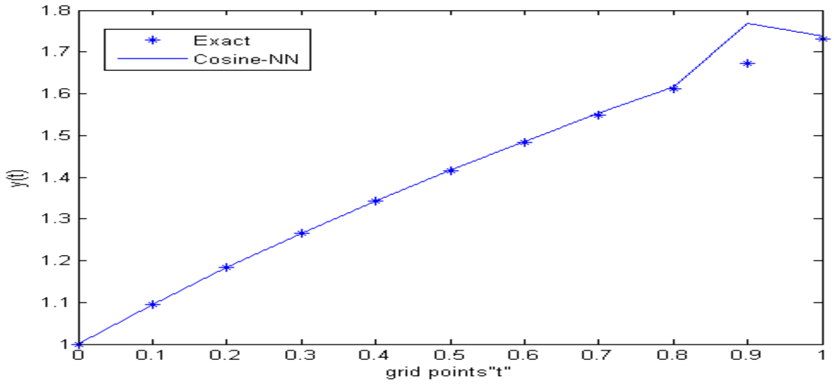

Figure 3. Cosine-NN method compared with exact method for [0-1] boundary values

Figure 3 shows the difference between the Cosine-NN and the exact solution of the given equation. The Cosine-NN [22], cannot give the better results than Euler and the M-Euler as it can be seen in the results.

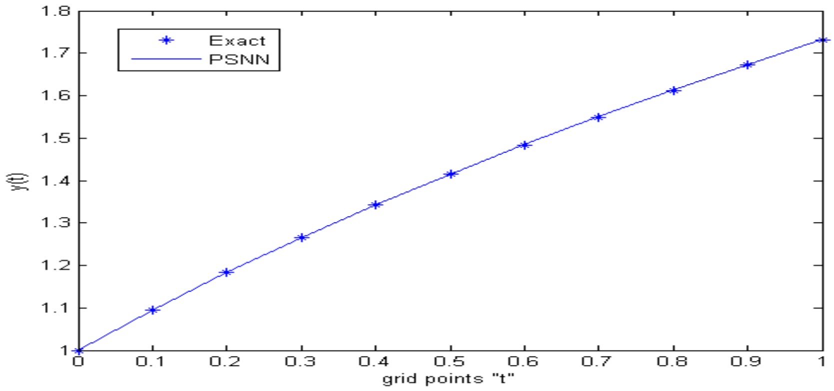

Figure 4. PSNN method compared with exact method for [0-1] boundary values

Figure 4 shows that PSNN performs better than the other techniques but still at higher values PSNN starts diverging and cannot performs better with the equations of higher order, this is the main drawback of the PSNN. Hence it has been observed that the proposed method performs well with the higher values, it converges at higher values. The difference between the values of Exact and PSNN is defined in the table 2.

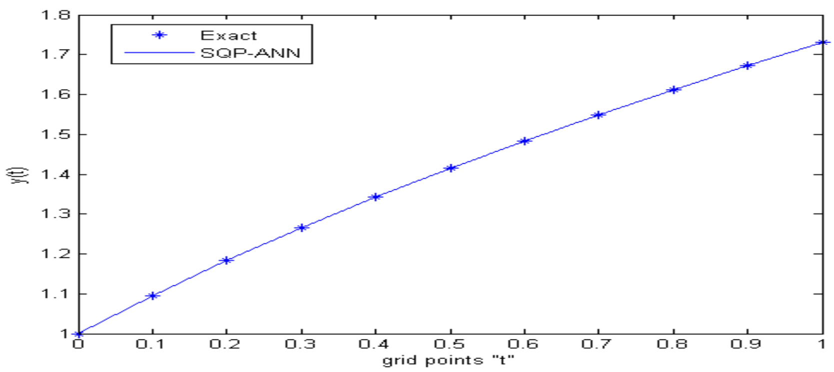

Figure 5. SQP-ANN method compared with exact method for [0-1] boundary values

In this research article SQP-ANN (figure 5) gives the best result than Euler, M-Euler, and Cosine-NN and even from the PSNN method though there is a small difference between PSNN and SQP-ANN but at higher values the PSNN starts divergent while the SQP-ANN converges and remain stable at all values which can be observed above.

It is clear from the above-mentioned result that the SQP-ANN provide accurate and best results as compared with the estimated values of other defined techniques for solving these differential equations. Absolute errors are calculated in decibel units so can be more elaborated. The result shown that the purposed method is more accurate than the Modified-Euler, Euler, cosine NN and from the PSNN. It is also observed that SQP-ANN is not only stable and accurate for solving second order differential equation but can also be used for solving higher order differential equations.

Table 2. Comparative Table Between Exact and Estimated Solutions.

T	Exact Solution	Euler	M-Euler	Cosine-NN	PSNN	SQP-ANN
0	1	1	1	1	1	1
0.1	1.0954	1.1	1.0959	1.0959	1.0955	1.0955
0.2	1.1832	1.1918	1.1841	1.1839	1.1832	1.1832
0.3	1.2649	1.2774	1.2662	1.2654	1.2649	1.2649
0.4	1.3416	1.3582	1.3434	1.3424	1.3417	1.3417
0.5	1.4142	1.4325	1.4164	1.4153	1.4143	1.4142
0.6	1.4832	1.509	1.486	1.4846	1.4833	1.4833
0.7	1.5492	1.5803	1.5525	1.5507	1.5493	1.5492
0.8	1.6125	1.4698	1.6165	1.6135	1.6125	1.6125
0.9	1.6733	1.7178	1.7682	1.6734	1.6734	1.6734
1	1.7321	1.7848	1.7379	1.7322	1.7319	1.7321

Table 2 shows the exact and the estimated values of different techniques and the SQP-ANN gives more accurate results as compared to other techniques.

Table 3. Absolute Error Between Exact and Estimated Solutions.

T	Euler	M-Euler	Cosine NN	PSNN	SQP-ANN
0	0	0	0	0	0
0.1	0.0046	0.0005	0.0005	0.1	0.1
0.2	0.0086	0.0009	0.0007	0	0
0.3	0.0125	0.0013	0.0005	0	0
0.4	0.0166	0.0018	0.0008	0.1	0.1
0.5	0.0183	0.0022	0.0011	0.1	0
0.6	0.0258	0.0028	0.0014	0.1	0.1
0.7	0.0311	0.0033	0.0015	0.1	0
0.8	0.1427	0.004	0.001	0	0
0.9	0.0445	0.0949	0.0001	0.1	0.1
1	0.0527	0.0058	0.0001	0.2	0

Absolute errors are calculated from the values defined in the table and is finally drawn and shown fig 6, these errors are calculated at decibel scale so that these errors can be elaborated.

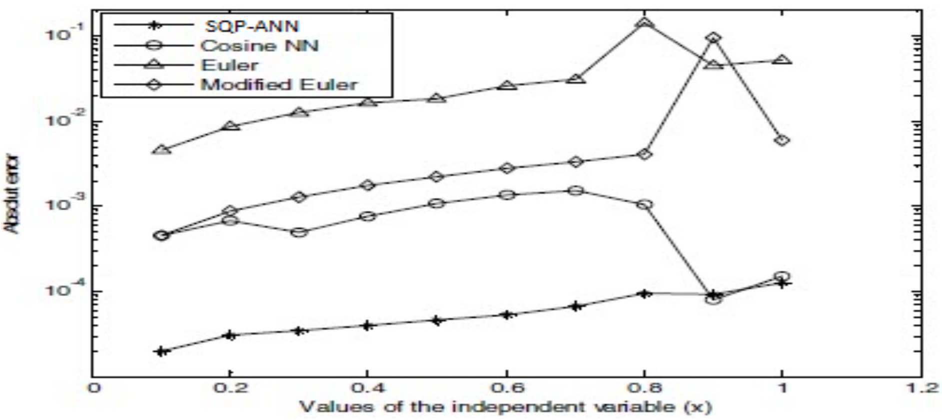

Figure 6. Absolute error of exact and estimated solution

3.2. Application of Jeffery Fluid Problem

Jeffry fluid is considered as a subclass of rate type fluids that could describe the characteristics of reduction and delay times. Due to the difficulties in this type of subclass fluid equation, it is not much elaborated in the literature. Al-Nimr et al. [23] discussed the passing coquette flow passing over finite domains wind-driven flow. Now a days, as compared to the other rate type equations Jeffery fluid equations are considered simpler, that is why these equations are most popular among the researchers [24-27]. The main Jeffry fluid equation is given as [28,29]:

$$\begin{cases} \frac{d^3 f}{dt^3} + (1+\lambda)\left(-\left(\frac{df}{dt}\right)^2 + f\frac{d^2 f}{dt^2}\right) + B\left(\left(\frac{d^2 f}{dt^2}\right)^2 + f\frac{d^4 f}{dt^4}\right) - (1+\lambda)M^2\frac{df}{dt} = 0 \\ f(0) = 0, \qquad \left.\frac{df}{dt}\right|_0 = 1 \end{cases} \tag{5}$$

The system given in above equation is solved by taking 5 and 15 neurons, respectively for αi, βi, wi adaptive parameters and the resultant profile is represented in Fig. 7, Fig. 8 respectively. It is observed from the figures that increase in number of neurons causes betterment in the fitness value however, the use of very large number of neurons can cause over fitting and the fitness value will overshoot.

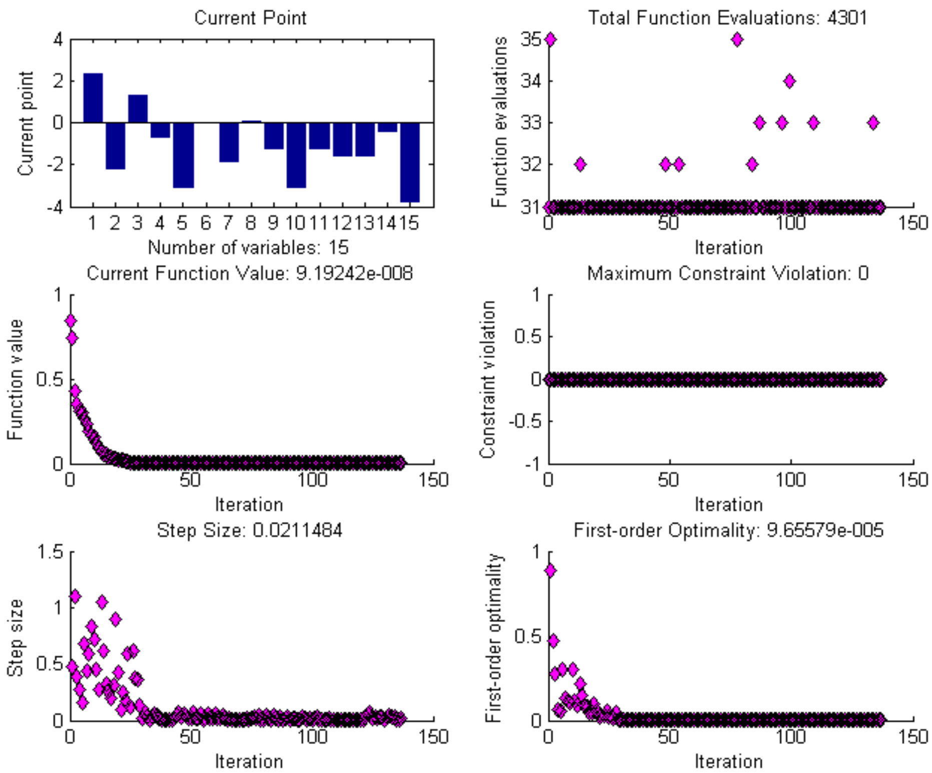

Figure 7. Profile using 5 number of neurons for input domain [0-1]

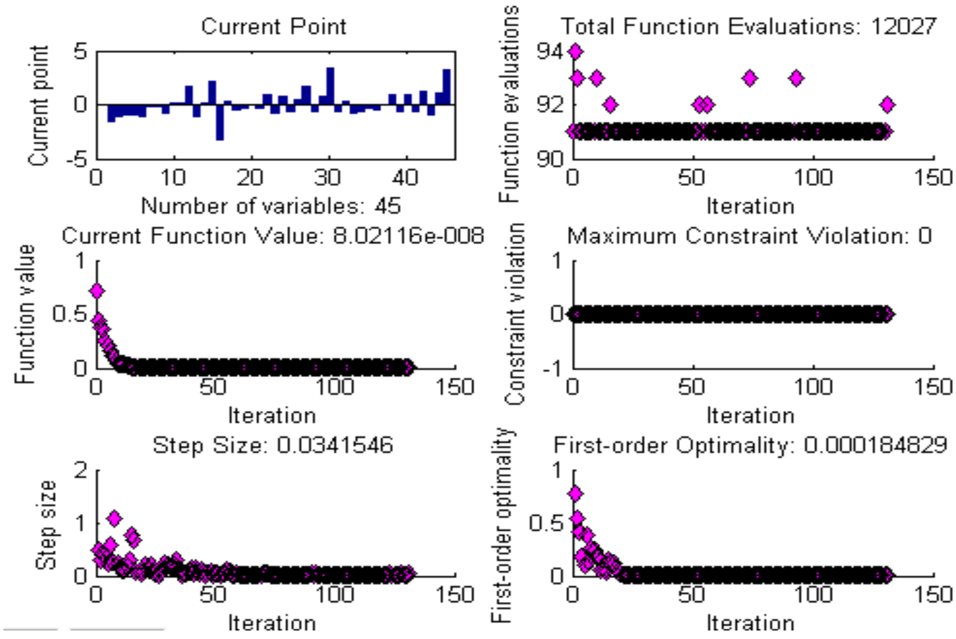

Figure 8. Profile using 15 number of neurons for input domain [0-1]

The numerical approximation for the input data set T= [0-3] second with a step size 0.1 with varying values of toke number is shown in Figure 9 while the velocity profile is shown in Figure 10, it is quite evident from the figures the proposed scheme approximates the Jeffery problem with an acceptable level of accuracy.

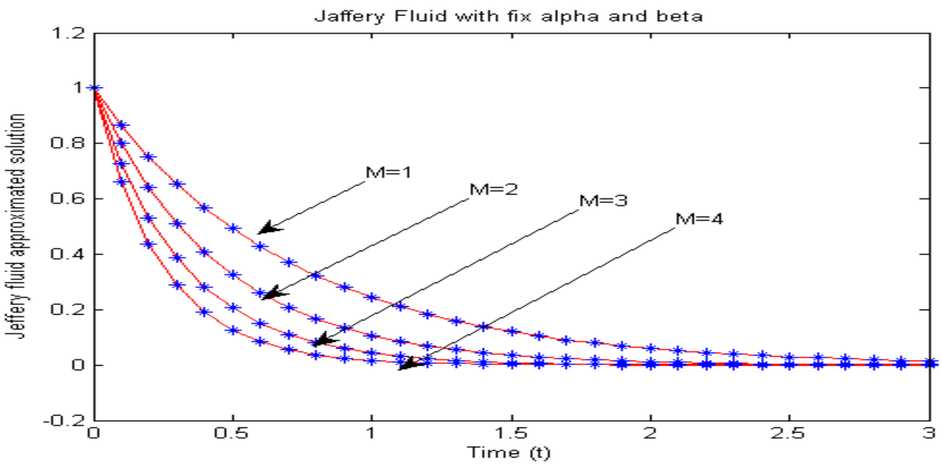

Figure 9. Jeffery Fluid profile for varying values of stokes number for input domain [0-3]

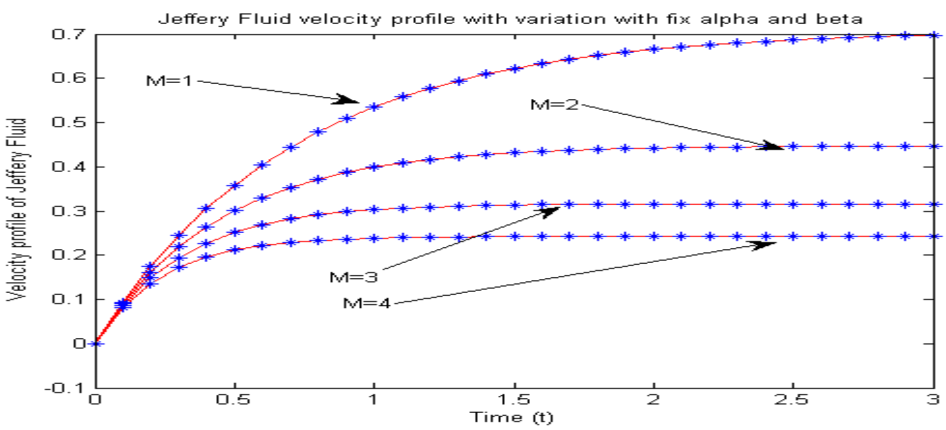

Figure 10. Jeffery Fluid velocity profile for varying values of stokes number for input domain [0-3]

The Jeffery fluid profile for λ=2, λ=3 and λ=4 is shown in figure 11 for a discrete input domain with N=30 no of neurons in the hidden layers.

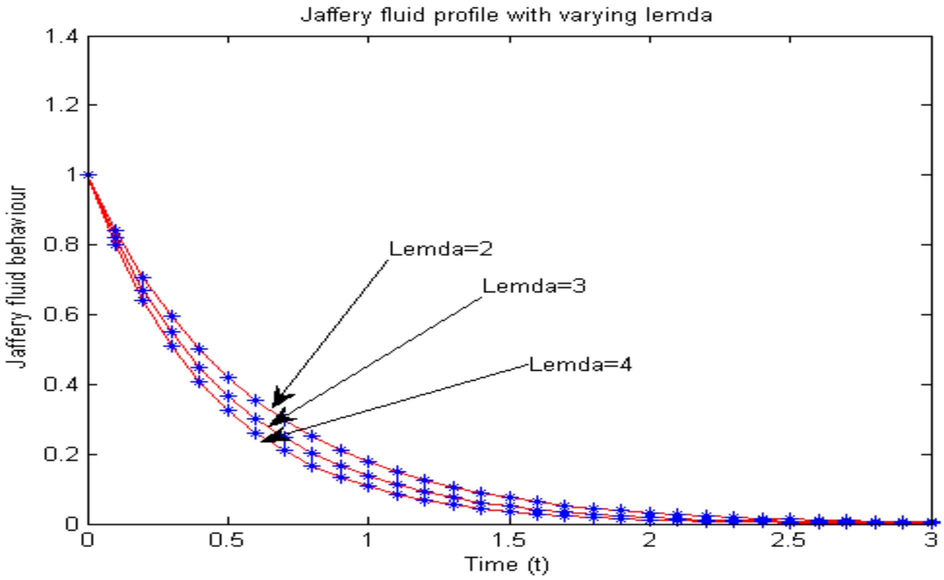

Figure 11. Jeffery Fluid profile for varying values of λ parameter for input domain [0-3]

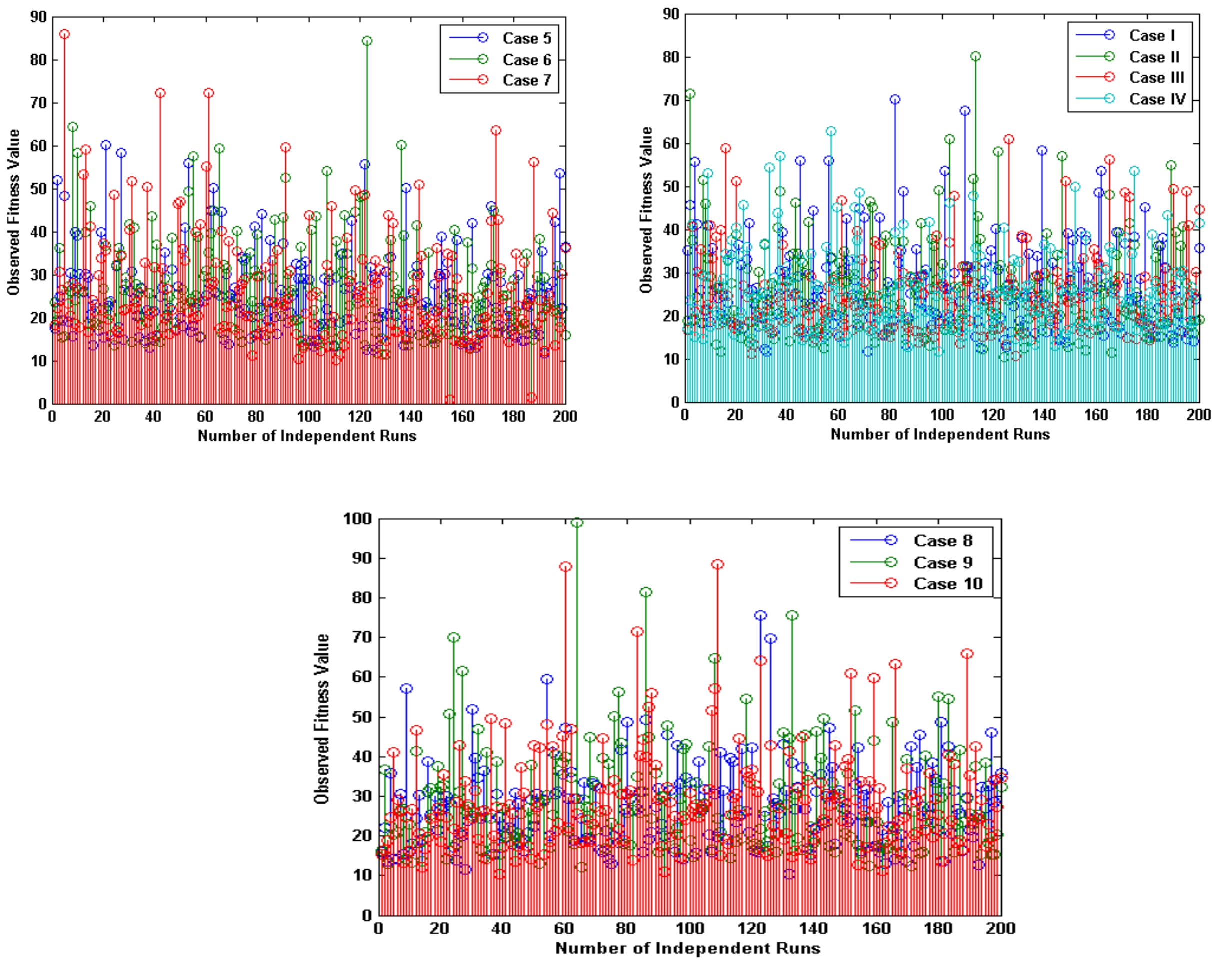

Figure 12. Computational complexity of the various cases for 200 independents

The complexity of the system in terms of time and space is calculated and is expressed in terms of 200 independent runs.

4. Conclusion

For complex non-linear and linear system, the accuracy in the result of proposed log sigmoid used with neural networks is achieved more than the techniques like sign functions, signed LMS algorithm, Euler, Modified Euler and the power series. The proposed methodology also defined the advantage of giving solid and exploiting formulas for these ordinary differential equations. Comparison with the exact solutions and the solutions obtained by the other conventional methods shows the potential of ANN in solving singular and nonlinear problems. Secondly, the advantage of the proposed technique is simple to understand, ease in implementation good convergence capability and more reliable in terms of accuracy. The average computational complexity for the optimization of the model, the different cases and equation of Jeffery fluid is used. In future, one can use many other equations like Thin film fluid. Many other search algorithms can be used to find the unknown parameters like Genetic algorithm, line search algorithm and other Hybrid techniques.

References

1. Eves, Howard. An introduction to the history of mathematics. Saunders College Publishing, 1983.
2. Billings, S. A., and S. Chen. "The determination of multivariable nonlinear models for dynamic systems using neural networks." (1996).
3. Zhang, Ke, Bin Jiang, and Vincent Cocquempot. "Adaptive observer-based fast fault estimation." International Journal of Control, Automation, and Systems 6, no. 3 (2008): 320-326.
4. Reid, William Thomas, and William Thomas Reid. Riccati differential equations. Vol. 86. Academic Press, 1972.
5. Reid, William Thomas, and William Thomas Reid. Riccati differential equations. Vol. 86. Academic Press, 1972.
6. Bittanti, S. "History and prehistory of the Riccati equation." In Proceedings of 35th IEEE Conference on Decision and Control, vol. 2, pp. 1599-1604. IEEE, 1996.
7. Goldstine, Herman Heine. A History of the Calculus of Variations from the 17th through the 19th Century. Vol. 5. Springer Science & Business Media, 2012.
8. Mitter, Sanjoy K. "Filtering and stochastic control: A historical perspective." IEEE Control Systems Magazine 16, no. 3 (1996): 67-76.
9. Pirzada, Aliya. "Modification of Heun's Iterative Method for the Population Growth Rate Problems." University of Sindh Journal of Information and Communication Technology 2, no. 1 (2018): 11-16.
10. Haweel, Tarek I., and A. M. Alhasan. "Adaptive power series solution for second order ordinary differential equations with initial conditions." In 2015 International Conference on Communications, Signal Processing, and their Applications (ICCSPA'15), pp. 1-6. IEEE, 2015.
11. Mathews, John H., and Kurtis D. Fink. Numerical methods using MATLAB. Vol. 4. Upper Saddle River, NJ: Pearson prentice hall, 2004.
12. Richard, L. "BURDEN AND J. DOUGLAS FAIRS. 2011. Numerical Analysis." Ninth Education, Youngstown state University (2001).
13. Meade Jr, Andrew J., and Alvaro A. Fernandez. "The numerical solution of linear ordinary differential equations by feedforward neural networks." Mathematical and Computer Modelling 19, no. 12 (1994): 1-25.
14. Lee, Hyuk, and In Seok Kang. "Neural algorithm for solving differential equations." Journal of Computational Physics 91, no. 1 (1990): 110-131.
15. Khater, A. H., D. K. Callebaut, S. F. Abdul-Aziz, and T. N. Abdelhameed. "Potential symmetry and invariant solutions of Fokker–Planck equation modelling magnetic field diffusion in magnetohydrodynamics including the Hall current." Physica A: Statistical Mechanics and its Applications 341 (2004): 107-122.
16. Khater, A. H., D. K. Callebaut, S. F. Abdul-Aziz, and T. N. Abdelhameed. "Potential symmetry and invariant solutions of Fokker–Planck equation modelling magnetic field diffusion in magnetohydrodynamics including the Hall current." Physica A: Statistical Mechanics and its Applications 341 (2004): 107-122.
17. Wang, Lixin, and Jerry M. Mendel. "Structured trainable networks for matrix algebra." In 1990 IJCNN International Joint Conference on Neural Networks, pp. 125-132. IEEE, 1990.
18. Yentis, R., and M. E. Zaghloul. "VLSI implementation of locally connected neural network for solving partial differential equations." IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 43, no. 8 (1996): 687-690.
19. Meade Jr, Andrew J., and Alvaro A. Fernandez. "The numerical solution of linear ordinary differential equations by feedforward neural networks." Mathematical and Computer Modelling 19, no. 12 (1994): 1-25.
20. Samath, J. Abdul, and P. Balasubramaniam. "Solution of differential-algebraic equation systems using neural networks." In International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), vol. 1, pp. 3-8. IEEE, 2007.
21. Waseem, Waseem, Muhammad Sulaiman, Poom Kumam, Muhammad Shoaib, Muhammad Asif Zahoor Raja, and Saeed Islam. "Investigation of singular ordinary differential equations by a neuroevolutionary approach." Plos one 15, no. 7 (2020): e0235829.
22. Pirzada, Aliya. "Modification of Heun's Iterative Method for the Population Growth Rate Problems." University of Sindh Journal of Information and Communication Technology 2, no. 1 (2018): 11-16.
23. Li-ying, Xu, Wen Hui, and Zeng Zhe-zhao. "The algorithm of neural networks on the initial value problems in ordinary differential equations." In 2007 2nd IEEE Conference on Industrial Electronics and Applications, pp. 813-816. IEEE, 2007.
24. Chowdhury, M. S. H., and Ishak Hashim. "Solutions of a class of singular second-order IVPs by homotopy-perturbation method." Physics Letters A 365, no. 5-6 (2007): 439-447.
25. Khadrawi, A. F., M. A. Al-Nimr, and Ali Othman. "Basic viscoelastic fluid flow problems using the Jeffreys model." Chemical engineering science 60, no. 24 (2005): 7131-7136.
26. Hayat, T., Niaz Ahmad, and N. Ali. "Effects of an endoscope and magnetic field on the peristalsis involving Jeffrey fluid." Communications in Nonlinear Science and Numerical Simulation 13, no. 8 (2008): 1581-1591.
27. Kothandapani, M., and S. Srinivas. "Peristaltic transport of a Jeffrey fluid under the effect of magnetic field in an asymmetric channel." International Journal of Non-Linear Mechanics 43, no. 9 (2008): 915-924.
28. Hayat, Tasawar, and Nasir Ali. "Peristaltic motion of a Jeffrey fluid under the effect of a magnetic field in a tube." Communications in Nonlinear Science and Numerical Simulation 13, no. 7 (2008): 1343-1352.
29. Nadeem, Sohail, and Noreen Sher Akbar. "Peristaltic flow of a Jeffrey fluid with variable viscosity in an asymmetric channel." Zeitschrift für Naturforschung A 64, no. 11 (2009): 713-722.