Power Series Neural Network Solution for Ordinary Differential Equations with Initial Conditions

Tarek I. Haweel
Biomedical Technology Systems Department
Faculty of Biomedical Applied Medical Sciences
Majmaah University
Majmaah, Saudi Arabia
e-mail: t.haweel@mu.edu.sa

Tarek .N. Abdelhameed
Mathematics Department, Faculty of Science,
Beni-Suef University, Beni-Suef, Egypt
Faculty of Engineering, Majmaah University
Majmaah, Saudi Arabia
e-mail: t.ahmed@mu.edu.sa

***Abstract*—Differential equations are very common in most academic fields. Modern digital control systems require fast on line and sometimes time varying solution schemes for differential equations. This paper presents new nonlinear adaptive numeric solutions for ordinary differential equations (ODE) with initial conditions. The main feature is to implement nonlinear polynomial expansions in a neural network-like adaptive framework. The transfer functions of the employed neural network follow a power series. The proposed technique does not use sigmoid or tanch non-linear transfer functions commonly adopted in conventional neural networks at the output. Instead, linear transfer functions are employed which leads to explicit power series formulae for the ODE solution. This allows extrapolation and interpolation which increase the dynamic numeric range for the solutions. The improved and accurate solutions for the proposed power series neural network (PSNN) are illustrated through simulated examples. It is shown that the performance of the proposed PSNN ODE solution outperforms existing conventional methods.**

Keywords—Ordinary differential equations; Power series; Neural network; Signed LMS adaptive algorithm.

I. INTRODUCTION

Differential equations (DE) whether ordinary, partial or algebraic are often employed to model and analyze systems in science and engineering that involve the change of some variables with respect to others [1]-[3], [22]. Most DE's are difficult to solve exactly which has motivated the exploration of approximate solutions for the DE. Traditional DE solution algorithms include Euler method, Modified Euler method, Rung-Kutta method and Adams method [4], [5]. The precession of such methods is low and the required computational burdens are high.

Other methods use basis-functions to represent the solution in analytic form and transform the original problem usually to a system of algebraic equations. Most of the previous work in solving DE's using neural networks is restricted to the case of solving the systems of algebraic equations which result from the discretization of the domain. The solution of a linear system of equations is mapped onto the architecture of a Hopfield neural network. The minimization of the network's energy function provides the solution to the system of equations [6]-[10].

Another approach to the solution of differential equations is based on the fact that certain types of splines can be derived by the superposition of piecewise linear activation functions [11], [12]. The solution of a DE using splines as basis functions can be obtained by solving a system of linear or nonlinear equations in order to determine the coefficients of splines. Such a solution form is mapped directly on the architecture of a feed forward neural network by replacing each spline with the sum of piecewise linear activation functions that correspond to the hidden units. This method considers local basis-functions and in general requires many splines (and consequently network parameters) in order to yield accurate solutions. Furthermore, it is not easy to extend these techniques to multidimensional domains.

Wavelet scaling functions method for the solution of partial differential equations is a kind of new numerical computational method [13], [14]. However, it is highly affected by ill conditioned matrices.

This paper proposes adaptive numeric solutions for ordinary DE (ODE) with initial conditions to alleviate problems encountered with conventional numeric solutions and to reduce the calculus effort. The main idea is to implement power series expansions in a neural network-like adaptive framework.

The rest of the paper is organized as follows. Section two presents the proposed technique. Simulated examples are investigated in section three to illustrate the potential of the proposed technique. Finally, section four concludes the work.

978-1-4799-6532-8/15/$31.00 ©2015 IEEE

II. PROPOSED POLYNOMIAL NEURAL NETWORK SOLUTION FOR DIFFERENTIAL EQUATIONS

Many applications give rise to DE's with solutions that can't be expressed in terms of elementary functions such as polynomials, rational functions, exponential and logarithmic functions, and trigonometric functions. The solutions of some of the most important of these equations can be expressed in terms of power series. Solutions to the second order linear DE having variable coefficients may be represented as a power series [15],[16]. These solutions are usually recursive and defined only in a neighborhood of a point where the power series is centered at. A functional analytic method was developed in [17] to prove that certain non-linear ODE's have a unique power series solution which converges absolutely in a specified disc of the complex plane. In [18], this method has been extended to certain systems of two non-linear ODE's and applied the result to a nonlinear differential system to obtain the power series solutions.

Neural Networks based on Fourier progression have been employed successfully to solve ODE's [19], [20]. Such techniques use the cosine as the transfer function of the neural network. The output of the neural network which is a sum of weighted cosine functions represents the estimate for the ODE solution. Such progression cannot easily model the derivatives of the estimated output which are needed to estimate the error of the neural network.

The proposed technique in this paper employs power series progression for the neural network solution of the ODE. The transfer function of the neural network is in the form of, xn. To illustrate the proposed technique, consider the first order initial value problem ODE

$$\begin{aligned} y'(x) &= f\left(x, y(x)\right) \\ y(0) = a \quad &; \quad y'(0) = b \end{aligned} \tag{1}$$

The ODE solution estimate, $\hat{y}(x)$ is proposed to be an nth order power series as

$$\hat{y}(x) = \sum_{n=0}^{N-1} w_n x^n \tag{2}$$

In matrix form, we can define the weight vector W and the regression vector X to be respectively as

$$W = \left[w_0 \quad w_1 \quad \cdots \quad w_{N-1}\right] \tag{3}$$

$$X = \left[1 \quad x \quad x^2 \quad \cdots \quad x^{N-1}\right] \tag{4}$$

Now, (2) may be written as

$$\hat{y}(x) = WX^T \tag{5}$$

Using (2), the estimated derivative, $\hat{y}'(x)$ is given by

$$\hat{y}'(x) = \sum_{n=1}^{N-1} n w_n x^{n-1} \tag{6}$$

From (2) & (6), it is clear that the initial values are directly defined as

$$\hat{y}(0) = w_0 \quad ; \quad \hat{y}'(0) = w_1 \tag{7}$$

While the initial values are difficult to define using the NN based on cosine transfer functions as in [20], the initial conditions are easily defined in this proposed NN based on power series transfer functions.

The NN error function is expressed as

$$e(k) = f\left(x, \hat{y}(x)\right) - \hat{y}'(x) \tag{8}$$

Using (6)

$$e(k) = f\left(x, \hat{y}(x)\right) - \sum_{n=1}^{N-1} n w_n x^{n-1} \tag{9}$$

The performance index to be minimized by adapting the NN coefficients is

$$J(k) = \frac{1}{2} \sum_{k=1}^{M} e^2(k) \tag{10}$$

Where M is the number of training samples. The nth weight adjustment factor is

$$\Delta w_n = -\mu \frac{\partial J(k)}{\partial w_n} \tag{11}$$

Where μ is the adaptation constant. This can be written as

$$\Delta w_n = -\mu \frac{\partial J(k)}{\partial e(k)} \frac{\partial e(k)}{\partial w_n} \tag{12}$$

Using (2), (6), and (10), (12) can be written as

$$\Delta w_n = -\mu e(k)\left[x^n f_y\left(x_k, y(x_k)\right) - n x_k^{n-1}\right] \tag{13}$$

In the proposed algorithm, a weight is adapted for each training sample. That is the suffix n in (13) is the same as k. Moreover, the signed LMS is employed instead of the conventional LMS [21]. The weight update for the proposed power series NN (PSNN) is then

$$\begin{aligned} w_k(m+1) &= w_k(m) \\ &- \mu e(k) \, sign\left\{\left[x^k f_y\left(x_k, y(x_k)\right) - k x_k^{k-1}\right]\right\} \end{aligned} \tag{14}$$

Where m is the iteration index, $k \in (1, M)$ and sign(.) is the signum function defined as

$$sign\ (x) = \begin{cases} +1 & ; \quad x > 0 \\ 0 & ; \quad x = 0 \\ -1 & ; \quad x < 0 \end{cases} \tag{15}$$

The signed LMS reduces the computational burden. Moreover, the term between square brackets in (14) is always negative in many problems. In these situations, the update is reduced to the surprisingly simple form

$$w_k(m+1) = w_k(m) + \mu e(k) \tag{16}$$

The schematic diagram for the proposed PSNN is illustrated in Fig.1.

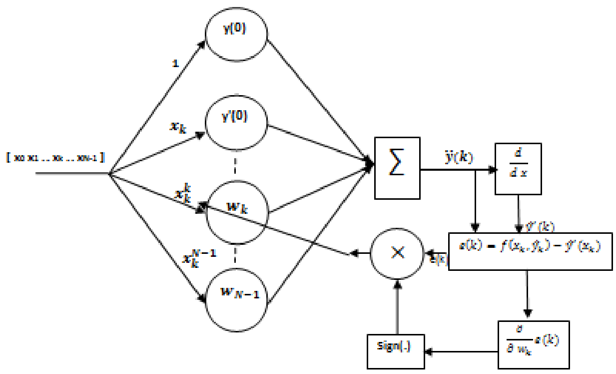

Fig.1 Schematic diagram for the Proposed PSNN

III. Simulated Examples

The following simulated examples illustrate the efficiency of the proposed PSNN in solving ODE.

Example-1

In this example, it is required to solve the first order ODE given by [20]

$$y' = y - \frac{2x}{y} \quad ; \quad x \in [0,1]$$
$$y(0) = 1 \quad ; \quad y'(0) = 1 \tag{17}$$

The domain is discretized in steps of 0.1 each. The proposed PSNN employs a 6th order power series as an estimate for the solution of this ODE. The weight associated with the zero power is freezed (not adapted) at y(0) while the weight associated with power one is freezed at y'(0). The estimated solution is then given by

$$\hat{y}(x) = y(0) + y'(0)x + \sum_{k=2}^{6} w_k x^k \tag{18}$$

In the PSNN, a weight is adapted for each of the discrete samples, 0.2, 0.3, 0.4, 0.5 and 0.6, which covers 50% only from the domain. These adapted five weights are initially zeroed. It is easy to prove that the term between square brackets in (14) is always negative and consequently the simple update in (16) has been employed. The convergent weights vector, W*, for the employed PSNN is

$$W^* = [-0.4947 \ \ 0.4486 \ \ 0.3907 \ \ 0.2330 \ \ -0.0643] \tag{19}$$

That is, the estimated PSNNODE solution, after convergence, for this example is given by

$$\hat{y}(x) = 1 + x - 0.4947x^2 + 0.4486x^3 + \\ + 0.3907x^4 + 0.2330x^5 - 0.0643x^6 \tag{20}$$

The exact and estimated solutions using the methods of Euler, modified Euler, Cosine NN [20] and proposed PSNN at the discrete points are illustrated in table-1.

Table-1 Exact and Estimated Solutions forExample-1

X	Exact	Euler	M-Euler	Cosine NN	PSNN
0.0	1.0000	1.0000	1.0000	1.0000	1.0000
0.1	1.0954	1.1000	1.0959	1.0959	1.0955
0.2	1.1832	1.1918	1.1841	1.1839	1.1832
0.3	1.2649	1.2774	1.2662	1.2654	1.2649
0.4	1.3416	1.3582	1.3434	1.3424	1.3417
0.5	1.4142	1.4325	1.4164	1.4153	1.4143
0.6	1.4832	1.5090	1.4860	1.4846	1.4833
0.7	1.5492	1.5803	1.5525	1.5507	1.5493
0.8	1.6125	1.4698	1.6165	1.6135	1.6125
0.9	1.6733	1.7178	1.7682	1.6734	1.6734
1.0	1.7321	1.7848	1.7379	1.7322	1.7319

It is clear from the table that the proposed PSNN method outperforms other methods. As a pictorial comparison, the absolute errors between estimated solutions using different methods and the exact solution are illustrated in Fig.2. The proposed PSNN has the smallest absolute error.

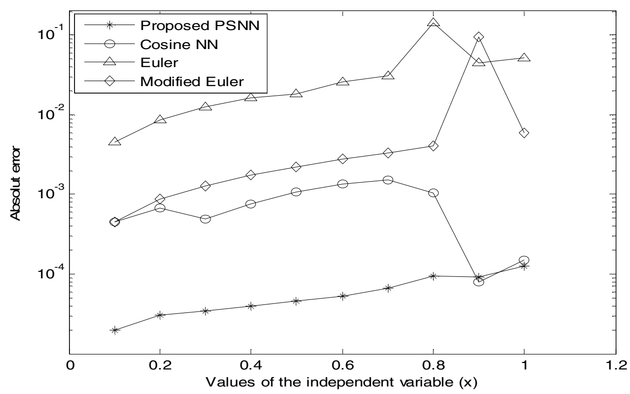

Fig. 2 Absolute Errors for Different Methods of Example-1

To estimate the interpolation and extrapolation capabilities for the proposed PSNN in solving ODE, the PSNN obtained solution of (20) has been employed to estimate the solution of (17) in discrete intervals of 0.01 within the [0,1] domain. Fig.3 shows the error power between the exact solution and the PSNN estimated solution in decibels. The k^{th} decibel error power, *EP*(*k*) is given by

$$EP(k) = 10 \log \left[y(x_k) - \hat{y}(x_k) \right]^2 \qquad (21)$$

Where the exact solution for this example is given by

$$y(x) = \sqrt{1 + 2x} \qquad (22)$$

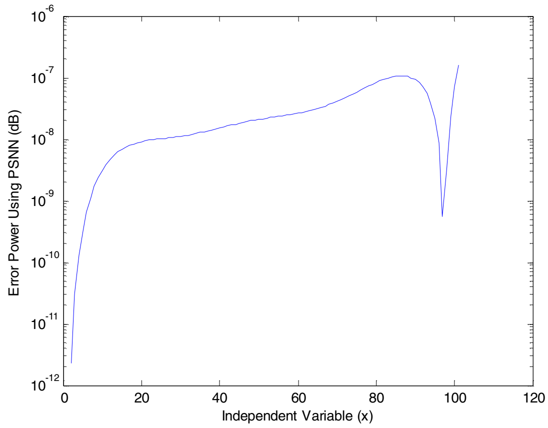

Fig. 3 Error Power Using PSNN for example-1

It is clear that the error power is very low for the whole domain. The error is comparatively higher for the range which has not been adapted. The average error power is 3.1854e-08db.

Example-2

In this example, it is required to solve the first order ODE given by [20]

$$y' = y - x^2 + 1 \quad ; \quad x \in [0,2]$$
$$y(0) = 0.5 \quad ; \quad y'(0) = 1.5 \qquad (23)$$

The domain is discretized in steps of 0.2 each. The proposed PSNN employs a 7th order power series as a solution for this ODE. Again, the weight associated with the zero power is freezed at y(0) while the weight associated with power one is freezed at y'(0). The solution is then given by

$$\hat{y}(x) = y(0) + y'(0)x + \sum_{k=2}^{7} w_k x^k \qquad (24)$$

A weight is adapted for each of the discrete samples, 0.4, 0.6, 0.8, 1.2, 1.4 and 1.6, in the PSNN which represents 60% only from the domain. These adapted six weights are initially zeroed. It is also easy to prove that the term between square brackets in (14) is always negative and again the simple update in (16) has been employed. The convergent weight vector, W*, for the employed PSNN are

$$W^* = \begin{bmatrix} 0.7340 & -0.0221 & -0.1236 & 0.0836 \\ -0.0382 & 0.0063 & & \end{bmatrix} \qquad (25)$$

That is, the estimated PSNN ODE solution, after convergence, for the second example is given by

$$\hat{y}(x) = 0.5 + 1.5x + 0.7340x^2 - 0.0221x^3 - 0.1236x^4 + $$
$$+ 0.0836x^5 - 0.0382x^6 + 0.0063x^7 \qquad (26)$$

The exact and estimated solutions using the methods of modified Euler, Heun's method, Cosine NN [20] and proposed PSNN at the discrete points are illustrated in table-2.

Table-2 Exact and Estimated Solutions forExample-2

X	Exact	Heun	M-Euler	Cosine NN	PSNN
0.0	0.5000	5.0000	5.0000	5.0000	5.0000
0.2	0.8292	0.8273	0.8260	0.8299	0.8290
0.4	1.2141	1.2099	1.2069	1.2137	1.2136
0.6	1.6489	1.6421	1.6372	1.6466	1.6483
0.8	2.1272	2.1176	2.1102	2.1257	2.1265
1.0	2.6408	2.6280	2.6177	2.6426	2.6400
1.2	3.1799	3.1635	3.1499	3.1755	3.1788
1.4	3.7324	3.7120	3.6973	3.7289	3.7311
1.6	4.2835	4.2588	4.2351	4.2780	4.2820
1.8	4.8152	4.7858	4.7556	4.8109	4.8150
2.0	5.3055	5.2713	5.2330	5.2865	5.3125

It is again clear from the table that the proposed PSNN method outperforms other methods. The absolute errors between estimated solutions using different methods and the exact solution are illustrated in Fig.4. Again, the proposed PSNN has the smallest absolute error. The exact solution for example-2 is given by

$$y(x) = -0.5e^x + x^2 + 2x + 1 \qquad (27)$$

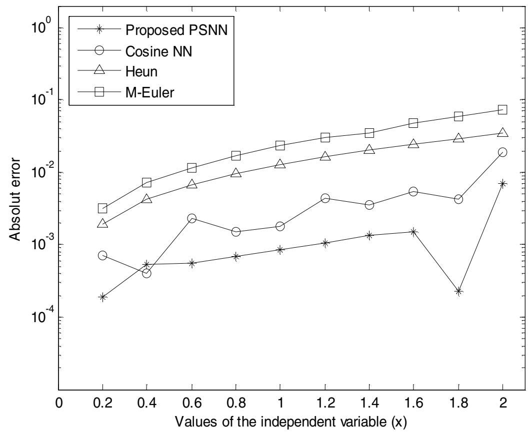

Fig. 4 Absolute Errors for Different Methods of Example-2

Fig. 5 shows the low decibel error power between the exact solution and the PSNN. It is again clear that the error power is very low for the whole domain. The error is also comparatively higher for the range which has not been adapted.

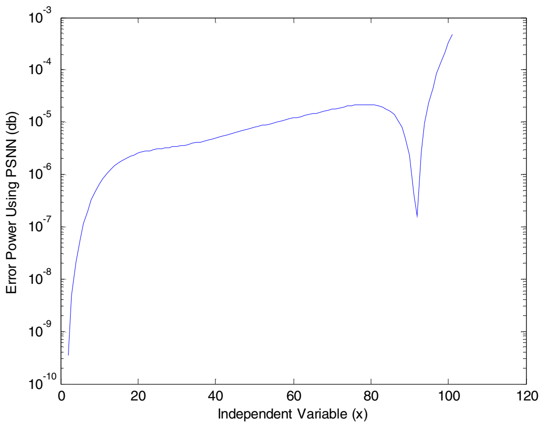

Fig. 5 Error Power Using PSNN for example-2

IV. CONCLUSIONS

Ordinary differential equations (ODE) are very common in many fields. This paper has presented an efficient adaptive solution for ODE with initial conditions. The proposed method suggests the employment of a neural network with power series transfer functions. The proposed power series neural network (PSNN) is fed with discrete values of the domain of the ODE. Part of the domain discrete values is involved in the adaptation of the neural network. The estimated solution is formed as a power series. The computationally efficient signed LMS algorithm is employed for adapting the PSNN. The adaptation is reduced to a very simple form for many ODE problems. The proposed technique has the great advantage of providing solid and explicit formulae for the ODE solutions. This adds extrapolation and interpolation capabilities and increases the dynamic range for the ODE numeric solutions. The adaptive nature of the proposed technique allows on line solutions for systems having time varying ODE. Simulated examples are provided to prove the efficiency of the proposed PSNN in solving ODE. It is shown that the proposed PSNN outperforms many other existing methods conventionally used to solve ODE.

V. REFERENCES

[1] Dongsheng Yang and Qi Wang, Differential equations of Mixed-systems modeling and simulation based on VHDL-AMS, Electromagnetic Compatibility (APEMC), 2010 Asia-Pacific Symposium on Digital Object Identifier, pp. 1398-1400, 2010.

[2] Mei Song Tong and Weng Cho Chew; Coupled Integral Equations for Microwave Induced Elastic Wave in Elastic Media, IEEE Transactions on Antennas and Propagation, vol.58 , Issue: 7, pp. 2309-2317, 2010.

[3] Guangchen Wang, Zhiyong Yu andA Pontryagin's,“Maximum Principle for Non-Zero Sum Differential Games of BSDEs with Applications,” IEEE Transactions on Automatic Control, vol. 55, Issue: 7, pp. 1742-1747, 2010.

[4] John H. Mathews and Kurtis D. Fink, Numerical methods using Matlab, seventh edition, Publishing House of Electronic Industry, pp. 318-362, June 2002.

[5] Richard L. Burden and J. Douglas Faires, Numerical Analysis, Seventh edition, Higher Education Press, pp. 21-289, Aug. 2001.

[6] H. Lee and I. Kang, “Neural algorithms for solving differential equations,”J. Comput. Phys., vol. 91, pp. 110–117, 1990.

[7] Khater AH, Callebaut DK, Abdelhameed TN and Abdul-Aziz SF,“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 Application, pp. 107-122, Oct. 2004.

[8] Khater A.H, Callebaut D.K and Abdelhameed T.N“Potential symmetry and invariant solutions of Fokker-Planck equation in cylindrical coordinates related to magnetic field diffusion in magnetohydrodynamics including the Hall current,”European Physical Journal, B 50 (1-2), pp. 17-21, Mar. 2006.

[9] L. Wang and J. M. Mendel, “Structured trainable networks for matrixalgebra,” IEEE Int. Joint Conf. Neural Networks, vol. 2, pp. 125–128,1990.

[10] R. Yentis and M. E. Zaghoul, “VLSI implementation of locally connectedneural network for solving partial differential equations,” IEEETrans. Circuits Syst. I, vol. 43, no. 8, pp. 687–690, 1996.

[11] A. J. Meade, Jr., and A. A. Fernandez, “The numerical solution of linearordinary differential equations by feedforward neural networks,” Math.Comput. Modeling, vol. 19, no. 12, pp. 1–25, 1994.

[12] J. Abdul Samath andP. Balasubramaniam, “Solution of differential-algebraic equation systems using neuralnetworks,” International Conference on Computational Intelligence and Multimedia Applications, pp. 3-8, 2007.

[13] Xiong Lei et al, "Summarizing the Method of Solution of the Majority of Differential Equation By Using Tensor Product B-Spline Wavelet Scaling Functions In Network Security and Software Security area," the Imitating Von Neumann Stability Analysis and a Typical Example, IEEE International Conference on Intelligent Human-Machine Systems and Cybernetics, pp. 3-6, 2009.

[14] Zhi Shi, Li-Yuan Deng and Qing-Jiang Chen"Numerical solution of differential equations by using Haar wavelets,"International Conference onWavelet Analysis and Pattern Recognition, ICWAPR '07 , vol. 3, pp. 1039-1044, 2007.

[15] Gabriel Nagy, "Ordinary Differential Equations," Mathematics Department,Michigan State University,USA, November 2013.

[16] William F. French, "Elementary Differential Equations," Department of Mathematics, Trinity University, San Antonio, Texas, USA, Free Edition, July 2013.

[17] Ifantis andK.Evangelos , "Analytic Solutions for Nonlinear Differential Equations,"Journal of Mathematical Analysis and Applications, 124, pp. 339-380, 1987.

[18] XU Liang, "A functional Analytic Approach to the Power Series Solutions of an nonlinear differential equations," IEEE Power and Energy Engineering Conference, (APPEEC) Asia-Pacific, pp. 1-4, 2012.

[19] Y. Wang, J. W. Lei, H. Li and J. Li, "Design of a New Kind of RBF Neural Network Based on Fourier Progression," Proceeding of the CICA, pp. 98-102, 2005.

[20] XU Li Ling, Wen Hui and Zeng ZENH Zhe Zhao, "The Algorithm of Neural Networks on the initial Value Problems in Ordinary Differential Equations," Second IEEE Conference on industrial Electronics and Applications (ICIEA), pp. 813-816, 2007.

[21] Baby Paul and P. Mythili,"ECG Noise Removal using GA tuned Sign-DataLeast Mean Square Algorithm," (ICACCCT),pp. 100-103, 2012.

[22] H. Ullah, S. Islam, I. Khan, S. Sharidan , T. N. Abdelhameed, M. Fiza, “Approximate Solution of the Generalized Hirota-Satsuma Coupled Kdv-Equation by Extended Optimal Homotopy Asymptotic Method,” MAGNT Research Report (ISSN. 1444-8939), vol. 2 (7), pp. 3302-3036, 2014.