Mediterr. J. Math. (2020) 17:30
https://doi.org/10.1007/s00009-019-1452-5

Mediterranean Journal of Mathematics

Neural Network Solution of Single-Delay Differential Equations

Jie Fang, Chenglian Liu, T. E. Simos and I. Th. Famelis

Abstract. Following the ideas of Lagaris et al. (IEEE Trans Neural Netw 9(5):987–1000, 1998), we use Neural Networks to solve approximatively first-order single-delay differential equations and systems. We apply the proposed novel methodology to various problems with constant delay terms and the resulted continuous solutions prove to be very efficient. This is the case not only for nonstiff problems but for equations with stiffness too.

Mathematics Subject Classification. Primary 65L03, 65L05, 65L99, 68T05; Secondary 68Q32.

Keywords. Neural networks, delay differential equations.

1. Introduction

Differential equations (DEs) are used to model a wide variety of physical phenomena, especially those that are evolutionary. When the unknown quantity of a DE involves only one variable, we have ordinary differential equations (ODEs) which may be subject to initial or boundary value conditions. First-order ODEs subject to a initial condition are called Initial Value Problems (IVPs) and have the form:

$$y'(x) = f(x, y), \; y(x_0) = A. \tag{1.1}$$

The theory of DEs provide analytical solutions only for a relative small number of IVPs. In the case that an analytical solution cannot be found, numerical procedures, which furnish the approximation of the solution in a abscissa of the integration interval, are used. An overview of the DE applications (e.g., celestial mechanics, chemical kinetics, mechanics, electrical engineering, biology modeling, etc.), an introduction to the relative classical mathematical theory and the numerical solution theory of ODEs can be found in the classical reference books [2–4]. The numerical treatment of IVPs has been a live research area that interests our research group [5–7] from various aspects

Published online: 19 December 2019

such as the solution using Runge–Kutta [8–12], Runge–Kutta Nystr$\ddot{o}$m [13–16], hybrid Numerov type [17–23] and multistep [24,25] methods.

However, for evolutionary problems that the unknown quantity and some of its derivatives may depend on different argument values, ODEs are not suitable models. Such equations are called functional differential equations (FDEs). A special form of FDEs is the single-delay differential equations (DDEs) which have the form:

$$y' = f(x, y, y(x - \tau)). \tag{1.2}$$

DDEs arise in various fields such as optimization and control theory and in mathematical models in X-rays, biological and chemical models, population dynamics, electrostatic charge, economical phenomena, mechanical applications, transport problems and other [2,26–31]. The existence and the computation of analytical solutions have been examined [32,33] but once more only a small percentage of DDEs can be solved analytically. So, numerical methods have been proposed and studied. The analytical solutions of DDEs and its derivatives usually suffer from discontinuities, a fact that inherits problems in the numerical procedure as well. Moreover, the delayed term must be evaluated in every step and for so an interpolation approach, usually based on the numerical approximations on the solution grid or a continuous extension of the numerical scheme, should be considered [34–42].

Lagaris et al. in their classical paper [1] proposed a Neural Network approach solution for various classes of differential equations. Ever since, the field of neural network (NN)-based solution for differential equations has interested the scientific community. The reasons are many and among them the fact that the neural network-based DE solution features specific advantages over the standard numerical methods. First, the resulted solution is differentiable and has a closed analytic form that can be used in any off-step point approximation. To achieve this with a classical numerical scheme, its continuous extension must be constructed, a fact that leads either to the increase of the computational cost or the reduction of the overall accuracy of the numerical scheme. Moreover, the neural network solution has very good generalization properties. The methodology is general and can be applied to either a problem subject to initial or orthogonal box conditions but to any problems subject to irregular and arbitrary shaped boundaries. The Neural Network-based solutions can be implemented in parallel architectures and in general it can be programmed in hardware interfaces. The book of Yadav et. al [43] and their work [44] is a good reference reading for this specific area. The work of Mall and Chakraverty [45,46] extend the ideas of Lagaris in other types of Neural Networks.

In this work, we propose a trial solution suitable for the solution of first-order constant DDEs. In Sect. 2, we present the basic elements of the Lagaris approach and in Sect. 3 we extend the neural network solution for DDEs. Finally, in Sect. 4, we present our numerical tests.

2. The Neural Network Approach Solution for Differential Equations

Lagaris et al. [1] proposed a Neural Network approach solution to the general form of differential equation

$$F(\vec{x}, y(\vec{x}), \nabla(\vec{x}), \nabla^2 y(\vec{x})) = 0, \;\; \vec{x} \in D \subseteq \mathbb{R}^s \tag{2.1}$$

subject to initial or boundary conditions.

To approximate the solution of (2.1), we discretize the domain D and its boundary S into a grid and so we have to solve the following system of equations:

$$F(\vec{x}, y(\vec{x}_i), \nabla y(\vec{x}_i), \nabla^2 y(\vec{x}_i)) = 0, \;\; \vec{x}_i \in \hat{D} \tag{2.2}$$

subject to constrains imposed by initial or boundary conditions. Supposing $Y_{tr}(\vec{x}, \vec{p})$ is a suitably differentiable parameterized trial solution, problem (2.2) is transformed to the minimization problem with respect to the adjustable parameters $\vec{p}$:

$$\min_{\vec{p}} \sum_{\vec{x}_i \in \hat{D}} (G(\vec{x}_i, Y_{tr}(\vec{x}_i, \vec{p}), \nabla Y_{tr}(\vec{x}_i, \vec{p}), \nabla^2 Y_{tr}(\vec{x}_i, \vec{p})))^2 \tag{2.3}$$

subject to constrains imposed by the initial or the boundary conditions.

In his paper, Lagaris for trial solution suggested the function

$$Y_{tr}(\vec{x}) = A(\vec{x}) + (\vec{x} - \vec{x}_0) N(\vec{x}, \vec{p}), \tag{2.4}$$

where $N(\vec{x}, \vec{p})$ is a single-output feedforward neural network with parameters $\vec{p}$. For this NN, n input units are fed through the input vector $\vec{x}$. The $A(\vec{x})$ is chosen so as not to depend on parameters and to satisfy the conditions (initial or boundary).

The Neural Network's structure has a single hidden layer and given a input vector $\vec{x} = [x_1, x_2, \ldots, x_n]$ the network yields

$$N(\vec{x}, \vec{p}) = \sum_{i=1}^{H} v_i \sigma \left(\sum_{j=1}^{n} p_{ij} x_j + u_i \right) + \nu,$$

where as activation function σ the sigmoid function $\sigma(z) = \frac{1}{1+\exp(-z)}$ was used.

The training procedure is based on a grid of κ integration interval points and the network is trained to minimize a cost function. As a cost function the sum of squares of the residuals of the differential equation in the training points is considered. Note that the proposed trial function is a continuous differentiable function.

For the single-ODE IVP case (1.1), the trial solution is

$$Y_{tr}(x) = A + (x - x_0) N(x, \vec{p}),$$

where A is the initial condition and the function to be minimized is

$$E[\vec{p}] = \sum_{i=1}^{\kappa} \left[Y'_{tr}(x_i) - f(x_i, Y_{tr}(x_i)) \right]^2 . \tag{2.5}$$

When IVP systems of K equations are considered the problem is

$$y_j'(x) = f_j(x, y_1, y_2, \ldots, y_K),\ Y_j(x_0) = A_j,\ (j = 1, 2, \ldots, K),$$

the trial solutions are

$$Y_{tr_j}(x) = A_i + (x - x_0)N_j(x, \vec{p}_j),\ where\, j = 1, \ldots K,$$

and the function to be minimized is

$$E[\vec{p}] = \sum_{j=1}^{K} \sum_{i}^{\kappa} \left[Y'_{tr_j}(x_i) - f_j(x_i, Y_{tr_1}(x_i), Y_{tr_2}(x_i), \ldots, Y_{tr_K}(x_i)) \right]^2.$$

Lagaris used quasi-Newton BFGS method for the minimization of the training cost function instead of back propagation which is the classical choice.

3. Neural Network Solutions of Single-Delay Differential Equations

For the solution of the single-DDE IVP

$$y' = f(x, y, y(x - \tau)),\ \ \text{where}\, x \in [a, b],\ \ \text{when}\, y(x) = Y_0(x),\ x \le a, \quad (3.1)$$

we adapt the ideas of Lagaris and we propose to consider as trial solution the function

$$Y_{tr}(x) = Y_0(x) + u(x - x_0)(x - x_0)N(x, \vec{p}), \quad (3.2)$$

where $N(x, \vec{p})$ is a single-output feedforward neural network with parameters $\vec{p}$, where n input units are fed through the vector x. The function $u(x)$ is a continuous analytic approximation of the Heaviside step function $H(x)$, e.g.,

$$u(x) = \frac{1}{2}(1 + \tanh(kx)) = \frac{1}{1 + \exp(-2kx)} \approx H(x),$$

where a large k is considered. The trial solution is chosen to yield the initial condition for $x \le a$ and it is continuous and differentiable and so it can be used to evaluate the delayed terms. The corresponding cost function to minimize is

$$E[\vec{p}] = \sum_{i=1}^{\kappa} \left[Y'_{tr}(x_i) - f(x_i, Y_{tr}(x_i), Y_{tr}(x_i - \tau)) \right]^2. \quad (3.3)$$

4. Numerical Experiments

In our numerical experiments, we considered a five-neuron hidden layer NN. We have noticed that increasing the number of neurons, increases the dimensionality of the problem, without a considerable gain in the method's performance. For the training we have used the Levenberg–Marquardt method [47] as implemented in Matlab. The number of training points was chosen to be about 10 points per unit of integration space and the maximum number of iterations (Epochs) was considered to be 20000. For validation reasons, the absolute maximum error was computed in a thinner grid with width one tenth

Table 1. The measured errors

Problem	Neural network	dd23
Problem 1	1.1×10^{-16}	1.1×10^{-3}
Problem 2	2.7×10^{-16}	530.3
Problem 3	1.6×10^{-2}	7.2×10^{-4}
Problem 4	1.4×10^{-15}	∞
Problem 5	2.9×10^{-3}	3.09×10^{-3}
Problem 6	2.59×10^{-13}	1.13×10^{-4}
Problem 7	8.32×10^{-16}	3.02×10^{-2}

of the training grid to form a more dense validation point set. All our the test problems can be found in Paul's test set [48]. For comparison reasons, we have solved the problems using the Matlab [49] `dde23` as well which is considered a standard and stable code for the solution of DDEs. The numerical results are summarized in Table 1. We did not restrict our numerical experiments to small integration intervals, as it is usually done [1], but we have chosen to solve over lengthier intervals something that it is more realistic.

4.1. Problem 1

The first problem is a DDE that has an analytical solution which is an analytic continuation of the initial function, it has no discontinuities, but at points $x = \{\frac{n\pi}{2}\}$ an $(n+1)$th order discontinuity in the numerical solution may occur:

$$\begin{aligned} y'(x) &= -y\left(x - \frac{\pi}{2}\right),\ x \geq 0 \\ Y_0(x) &= sin(x),\ x \leq 0. \end{aligned} \tag{4.1}$$

This problem has analytical solution

$$y(x) = \sin(x).$$

We train the neural network to solve in the interval $[0, 4\,\pi]$ with 20 training points. The neural network solution yields an error of 1.1×10^{-16} something quite normal as the analytical solution which is an analytic continuation of the initial function but it reveals the really strong approximating capability of neural networks as global approximators. The `dde23` results in an error of 0.0011 (Fig. 1).

4.2. Problem 2

Another DDE with an analytical solution which is an analytic continuation of the initial function too follows:

$$\begin{aligned} y'(x) &= y(x) + y(x - \pi) + 3\cos(x) + 5\sin(x),\ x \geq 0 \\ Y_0(x) &= 3\cos(x) - 5\sin(x),\ x \leq 0. \end{aligned} \tag{4.2}$$

Its analytical solution is

$$y(x) = 3\cos(x) - 5\sin(x),\ x \geq 0$$

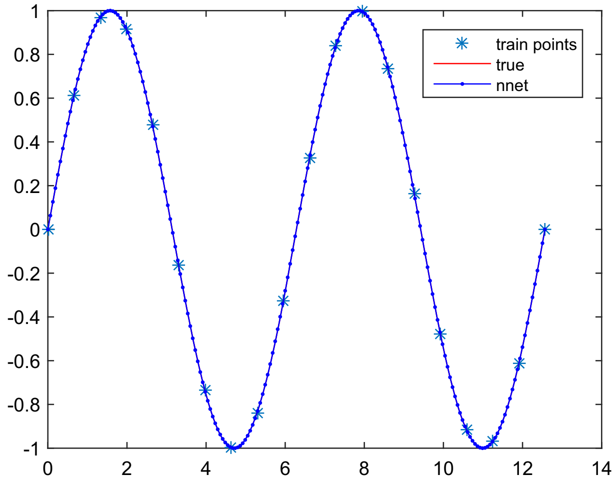

Figure 1. The neural network solution of Problem (4.1)

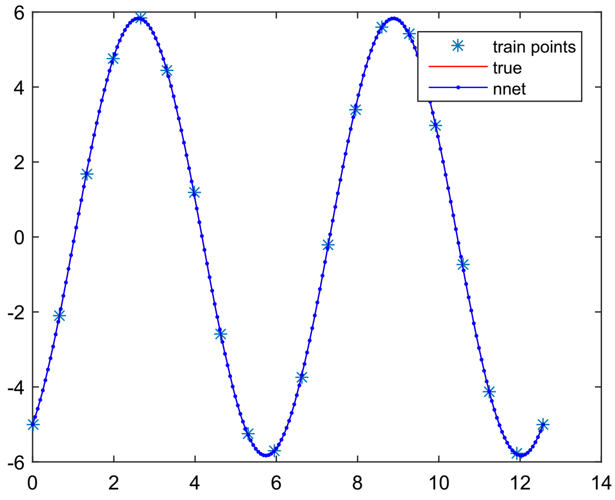

Figure 2. The neural network solution of Problem (4.2)

and has no discontinuities, but at $x = n\pi$ an $(n+1)$th order discontinuity in the numerical solution may occur. We train the neural network to solve in the interval $[0, 4\,\pi]$ with 20 training points. The NN solution yields an error of 2.7×10^{-16}. When we neglect the forcing term an unstable DDE results and due to this reason many numerical codes perform poorly. So does the `dde23` which diverges and yields an error 530.3 (Fig. 2).

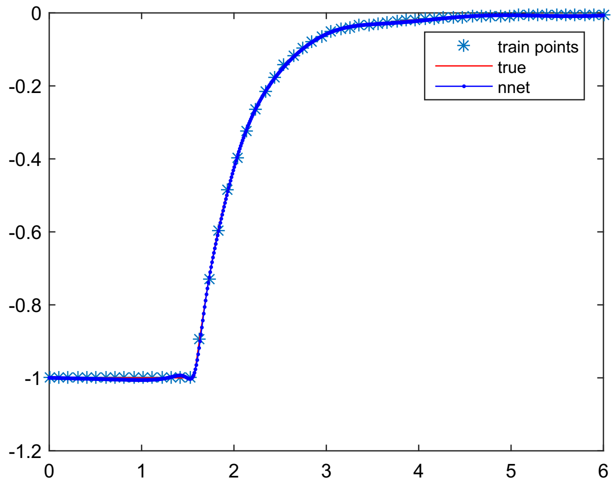

Figure 3. The neural network solution of Problem (4.3)

4.3. Problem 3

A DDE that has discontinuous initial function and an analytical solution with an nth order discontinuity at $x = \{\frac{(2n-1)\pi}{2}, n\pi\}$ is chosen:

$$y'(x) = y(x-\pi)y(x),\ \ x \in [0,6]$$
$$Y_0(x) = \begin{cases} 0 & \text{when}\, x < -\frac{\pi}{2} \\ -2 & \text{when}\, -\frac{\pi}{2} \le x < 0 \\ -1 & \text{when}\, x = 0 \end{cases} \tag{4.3}$$

with analytical solution

$$y(x) = \begin{cases} -1 & \text{when}\, 0 \le x \le \frac{\pi}{2} \\ -e^{\pi-2x} & \text{when}\, \frac{\pi}{2} \le x \le \pi \\ -e^{-x} & \text{when}\, \pi \le x \le \frac{3\pi}{2} \\ -e^{-\frac{3}{2}\pi+\frac{1}{2}(e^{3\pi-2x}-1)} & \text{when}\, \frac{3\pi}{2} \le x \le 6. \end{cases}$$

The neural network is trained to solve in the interval $[0,6]$ with 60 training points and in the validation points an error of 1.6×10^{-2} occurs. In this case, dde23 gets better results achieving an error 7.2×10^{-4}. It is known that neural networks are universal approximators [50, 51] that can approximate any continuous mapping. The discontinuity in the initial function explains this behavior (Fig. 3).

4.4. Problem 4

A stiff problem that dde23 fails to solve follows:

$$y'(x) = -(30 + e^{45\pi})(y(x) - \sin(x)) + y\left(x - \frac{3\pi}{2}\right),\ \ x \in [0,6]$$
$$Y_0(x) = e^{-(30+e^{45\pi})x} + \sin(x),\ \ x \le 0. \tag{4.4}$$

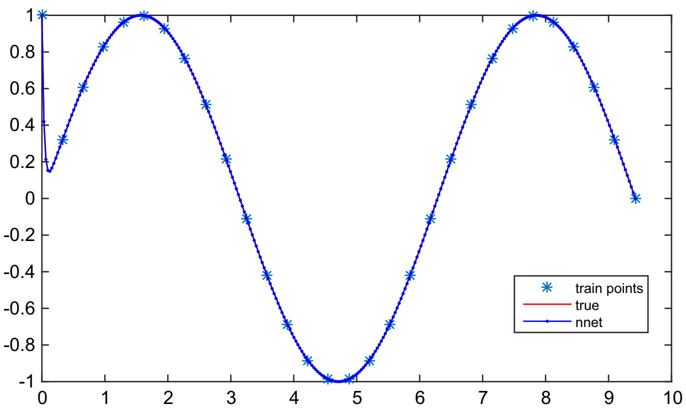

Figure 4. The neural network solution of Problem (4.4)

This problem has analytical solution

$$y(x) = e^{-(30+e^{45\pi})x} + \sin(x).$$

We train the neural network to solve in the interval $[0, 3\,\pi]$ with 30 training points. The neural network solution yields an error of 1.4×10^{-15} (Fig. 4).

4.5. Problem 5

We know choose a system of DDEs

$$\begin{aligned} y_1'(x) &= y_1(x-\pi) + y_2(x), \;\; x \geq 0 \\ y_2'(x) &= y_1(x) + y_1(x-1), \;\; x \geq 0 \\ Y_1^0(x) &= e^x, \;\; x \leq 0 \\ Y_2^0(0) &= 1 - e^{-1}, \;\; x \leq 0 \end{aligned} \tag{4.5}$$

with analytical solution

$$\begin{aligned} y_1(x) &= e^x \\ y_2(x) &= e^x - e^{x-1}. \end{aligned}$$

We train the neural network to solve in the interval $[0, 2\,\pi]$ with 20 training points. The NN solution yields an error of 2.9×10^{-3} whereas the `dde23` 3.09×10^{-3} (Fig. 5).

4.6. Problem 6

A coupled system of DDEs follows:

$$\begin{aligned} y_1'(x) &= -\,y_1\left(x - \frac{\pi}{2}\right), \;\; x \geq \frac{\pi}{2} \\ y_2'(x) &= -\,y_2\left(x - \frac{\pi}{2}\right), \;\; x \geq \frac{\pi}{2} \\ Y_1^0(x) &= \sin(x), \;\; x \leq \frac{\pi}{2} \\ Y_2^0(x) &= \cos(x), \;\; x \leq \frac{\pi}{2} \end{aligned} \tag{4.6}$$

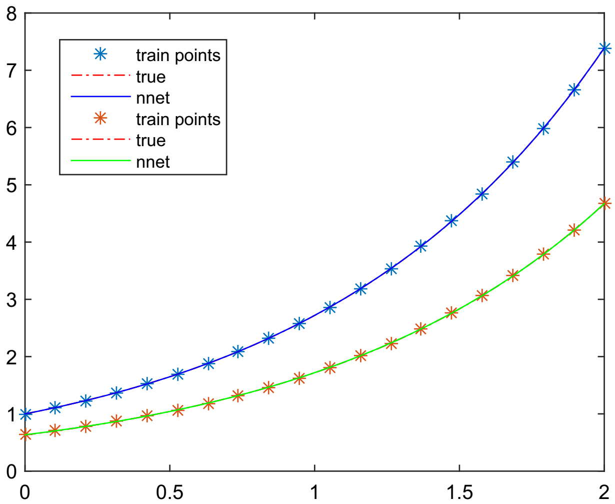

Figure 5. The neural network solution of Problem (4.5)

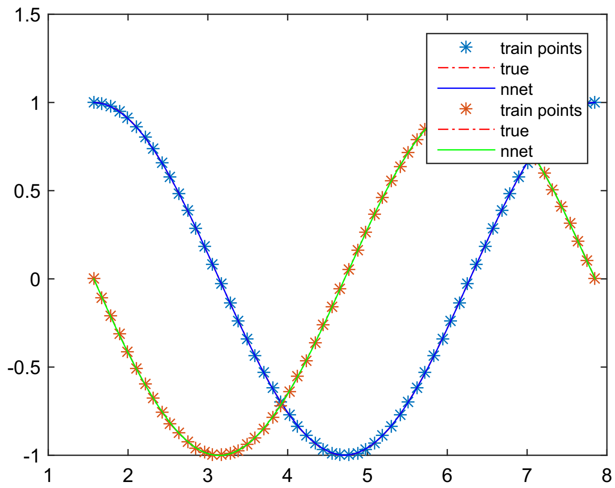

Figure 6. The neural network solution of Problem (4.6)

with analytical solution

$$y_1(x) = \sin(x), \;\; x \geq \frac{\pi}{2}$$
$$y_2(x) = \cos(x), \;\; x \geq \frac{\pi}{2}.$$

We train the neural network to solve in the interval $[\frac{\pi}{2}, \frac{5\pi}{2}]$ with 20 training points. The NN solution yields an error of 2.59×10^{-13} whereas the `dde23` 1.13×10^{-4} (Fig. 6).

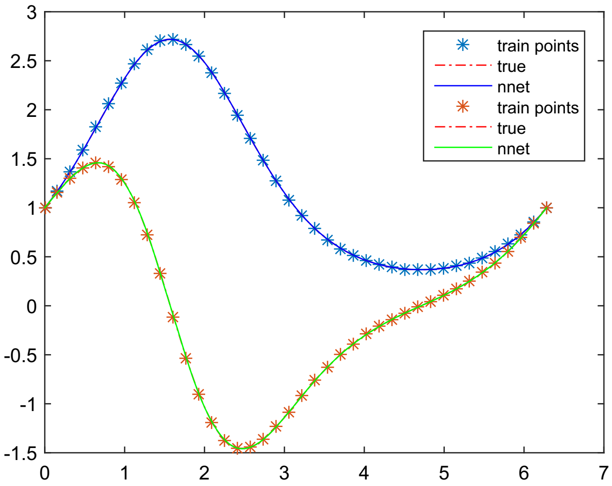

Figure 7. The neural network solution of Problem (4.7)

4.7. Problem 7

Finally, we solve a second-order DDE which we transform to the following system of first-order DDEs:

$$
\begin{aligned}
&y_1'(x) = y_2(x),\ x \geq 0\\
&y_2'(x) = 2y_1\left(x - \frac{\pi}{2}\right) + e^{\sin(x)}(\cos^2(x) - \sin(x)) - 2e^{-\cos(x)},\ x \geq 0\\
&Y_1^0(x) = e^{\sin(x)},\ x \leq 0\\
&Y_2^0(x) = \cos(x)e^{\sin(x)},\ x \leq 0
\end{aligned}
\tag{4.7}
$$

with analytical solution

$$
\begin{aligned}
y_1(x) &= e^{\sin(x)},\ x \geq 0\\
y_2(x) &= \cos(x)e^{\sin(x)},\ x \geq 0.
\end{aligned}
$$

We train the NN to solve in the interval $[0, 2\pi]$ with 40 training points. The neural network solution yields an error of 8.32×10^{-16} whereas the `dde23` 3.02×10^{-2} (Fig. 7).

5. Discussion and Conclusions

We have introduced a neural network approach for the solution of the first-order single-delay differential equations. To do so, we have adopted a very simple feedforward neural network with a hidden layer with only five neurons which keeps the dimensionality of the corresponding minimization problem low. The proposed procedure solves very efficiently the test problems, especially those that the analytical solution is an analytic continuation of the initial function. Moreover, in other cases, the solution is remarkably successful. Especially in the case of stiff problems, the solution does not face the difficulties that traditional numerical methodologies do. In almost all cases, the global error achieved is smaller compared to the error the Matlab function `dde23` achieves. We intend to extend the ideas presented here for the approximative solution of other classes of functional differential equations. Simultaneously, we plan to investigate more advanced neural network setups and techniques to improve the already very good results.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

[1] Lagaris, I.E., Likas, Fotiadis, A, D.I.: Artificial neural networks for solving ordinary and partial differential equations. IEEE Trans. Neural Netw. **9**(5), 987–1000 (1998)

[2] Hairer, E., Norsett, S.P., Wanner, G.: Solving Ordinary Differential Equations I, Nonstiff Problems, Second edn. Springer, Berlin Heidelberg (1993)

[3] Hairer, E., Wanner, G.: Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems, Second Revised edn. Springer, Berlin Heidelberg (1996)

[4] Butcher, J.C.: Numerical Methods for Ordinary Differential Equations, 3rd edn. Wiley, Hoboken (2016)

[5] Simos, T.E.: High order closed Newton-Cotes trigonometrically-fitted formulae for the numerical solution of the Schrödinger equation. Appl. Math. Comput. **209**, 137–151 (2009)

[6] Simos, T.E.: Closed Newton–Cotes trigonometrically-fitted formulae of high order for long-time integration of orbital problems. Appl. Math. Lett. **22**, 1616–1621 (2009)

[7] Simos, T.E.: Exponentially and trigonometrically fitted methods for the solution of the Schrödinger equation. Acta Appl. Math. **110**, 1331–1352 (2010)

[8] Monovasilis, Th, Kalogiratou, Z., Simos, T.E.: A family of trigonometrically fitted partitioned Runge–Kutta symplectic methods. Appl. Math. Comput. **209**, 91–96 (2009)

[9] Tsitouras, Ch., Famelis, I.Th, Simos, T.E.: On modified Runge–Kutta trees and methods. Comput. Math. Appl. **62**, 2101–2111 (2011)

[10] Tsitouras, Ch., Famelis, I.Th, Simos, T.E.: Phase-fitted Runge–Kutta pairs of orders 8(7). J. Comput. Appl. Math. **321**, 226–231 (2017)

[11] Simos, T.E., Tsitouras, Ch.: Fitted modifications of classical Runge–Kutta pairs of orders 5(4). Math. Meth. Appl. Sci. **41**, 4549–4559 (2018)

[12] Simos, T.E., Tsitouras, Ch.: Fitted modifications of Runge–Kutta pairs of orders 6(5). Math. Meth. Appl. Sci. **41**, 6184–6194 (2018)

[13] Kalogiratou, Z., Monovasilis, Th, Simos, T.E.: New modified Runge–Kutta–Nystrom methods for the numerical integration of the Schrödinger equation. Comput. Math. Appl. **60**, 1639–1647 (2010)

[14] Monovasilis, Th, Kalogiratou, Z., Simos, T.E.: Exponentially fitted symplectic Runge–Kutta–Nyström methods. Appl. Math. Inf. Sci. **7**, 81–85 (2013)

[15] Monovasilis, T., Kalogiratou, Z., Simos, T.E.: Construction of exponentially fitted symplectic Runge–Kutta–Nyström methods from partitioned Runge–Kutta methods. Mediterr. J. Math. **13**, 2271–2285 (2016)

[16] Papadopoulos, D.F., Simos, T.E.: A modified Runge–Kutta–Nyström method by using phase lag properties for the numerical solution of orbital problems. Appl. Math. Inf. Sci. **7**, 433–437 (2013)

[17] Simos, T.E.: Multistage symmetric two-step P-stable method with vanished phase-lag and its first, second and third derivatives. Appl. Comput. Math. **14**, 296–315 (2015)

[18] Hui, Fei, Simos, T.E.: Four stages symmetric two-step P-stable method with vanished phase-lag and its first, second, third and fourth derivatives. Appl. Comput. Math. **15**, 220–238 (2016)

[19] Monovasilis, T., Kalogiratou, Z., Ramos, Higinio, Simos, T.E.: Modified two-step hybrid methods for the numerical integration of oscillatory problems. Math. Methods Appl. Sci. **40**, 5286–5294 (2017)

[20] Simos, T.E., Tsitouras, Ch.: Evolutionary generation of high order, explicit, two step methods for second order linear IVPs. Math. Meth. Appl. Sci. **40**, 6276–6284 (2017)

[21] Simos, T.E., Tsitouras, Ch.: A new family of 7 stages, eighth-order explicit Numerov-type methods. Math. Meth. Appl. Sci. **40**, 7867–7878 (2017)

[22] Simos, T.E., Tsitouras, Ch., Famelis, I.Th: Explicit numerov type methods with constant coefficients: a review. Appl. Comput. Math. **16**, 89–113 (2017)

[23] Tsitouras, Ch., Simos, T.E.: On ninth order, explicit Numerov type methods with constant coefficients. Mediterr. J. Math. **15** (2018) Article No:46

[24] Alolyan, I., Simos, T.E.: A family of high-order multistep methods with vanished phase-lag and its derivatives for the numerical solution of the Schrödinger equation. Comput. Math. Appl. **62**, 3756–3774 (2011)

[25] Simos, T.E.: New stable closed newton-cotes trigonometrically fitted formulae for long-time integration. Abstract Appl. Anal. **Volume 2012**, Article ID 182536, 15 (2012). https://doi.org/10.1155/2012/182536

[26] Bellen, A., Zennaro, A.: Numerical Methods for Delay Differential Equations. Oxford University Press, New York (2003)

[27] Baker, C.T.H., Bocharov, G.A., Rihan, F.A.: A report on the use of delay differential equations in numerical modelling in the biosciences. Num. Anal. Rep. **343** (1999), Un. of Manchester/UMIST

[28] Bocharov, G.A., Rihan, F.A.: Numerical modelling in biosciences using delay differential equations. J. Comput. Appl. Math. **125**(1–2), 183–199 (2000)

[29] Rihan, F.A., Abdelrahman, D.H., Al-Maskari, F., Ibrahim, F., Abdeen, M.A.: Delay differential model for tumour-immune response with chemoimmunotherapy and optimal control. Comput. Math. Methods Med. **Article ID 982978**, 15 (2014)

[30] Baker, C.T.H., Bocharov, G.A., Paul, C.A.H., Rihan, F.A.: Modelling and analysis of time-lags in some basic patterns of cell proliferation. J. Math. Biol. **37**(4), 341–371 (1998)

[31] Lakshmanan, S., Rihan, F.A., Rakkiyappan, R., Park, J.H.: Stability analysis of the differential genetic regulatory networks model with time-varying delays and Markovian jumping parameters. Nonlinear Anal. Hybrid Syst. **14**, 1–15 (2014)

[32] Driver, R.D.: Ordinary and Delay Differential Equations. Springer, New York (1977)
[33] Hale, J.: Theory of Functional Differential Equations. Springer, Berlin Heidelberg (1977)
[34] Baker, C.T.H., Paul, C.A.H., Wille, D.R.: Issues in the numerical solution of evolutionary delay differential equations. Num. Anal. Report **248** (1994), Un. of Manchester/UMIST
[35] Higham, D.J., Famelis, I.Th: Equilibrium states of adaptive algorithms for delay differential equations. J Comp. App. Math. **58**, 151–169 (1995)
[36] Oberle, H.J., Pesch, H.J.: Numerical treatment of delay differential equations by hermite interpolation. Numer. Math. **37**, 235–255 (1981)
[37] Arndt, H.: The influence of interpolation on the global error in retarded differential equations. Differential Difference Equations, Edited by Collatz, Meinardus and Wetterling, ISNM 62 **1983**, Birkhauser, Boston, pp. 9–17
[38] Arndt, H.: Numerical solution of retarded initial value problems: local and global error and stepsize control. Numer. Math. **43**, 343–360 (1984)
[39] Arndt, H., Van Der Houwen, P.J., Sommeijer, B.P.: Numerical integration of retarded differential equations with periodic solutions. Int. Ser. Num. Math. **74**, 41–51 (1985)
[40] Papageorgiou, G., Famelis, I.Th: On using Explicit Runge Kutta Nyström methods for the treatment of retarded differential equations with periodic solutions. Appl. Math. Comp. **102**, 63–76 (1999)
[41] In't Hout, K.J., Spijker, M.N.: The θ-methods in the numerical solution of delay differential equations. Report **TW-89-03** (Second edition 1989), Dep. of Math. and Comp. Sc., Leiden University
[42] Weiderholt, L.F.: Stability of multistep methods for delay differential equations. Math. Comput. **30**, 283–290 (1976)
[43] Yadav, N., Yadav, A., Kumar, M.: An Introduction to Neural Network Methods for Differential Equations. Springer, New York (2015)
[44] Kumar, M., Yadav, N.: An introduction to Neural Network Methods fro Differential Equations. Comput. Math. Appl. **62**, 3796–3811 (2011)
[45] Mall, S., Chakraverty, S.: Application of Legendre neural network for solving ordinary differential equations. App. Soft Comp. **43**, 347–356 (2016)
[46] Mall, S., Chakraverty, S.: Numerical Solution of nonlinear singular initial value problems of Emden-Fowler type using Chebyshev Neural Network method. NeuroComputing **149**, 975–982 (2015)
[47] Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, New York (2006)
[48] Paul, C.A.H.: A test set of functional differential equations. Num. Anal. Rep. **243** (1994), Un. of Manchester/UMIST
[49] Matlab, MATLAB version 7.10.0. Natick, Massachusetts: The MathWorks Inc., 2010
[50] Cybenko, G.: Approximation to superpositions pf a sigmodial fucntion. Math. Control Signals Syst. **2**, 303–314 (1989)
[51] Hornik, K.: Approximation capabilities of multilayer feedforward networks. Neural Netw. **4**(2), 251–257 (1991)

Jie Fang
School of Electronics and Information Engineering
Fuqing Branch of Fujian Normal University
Fuqing 350300
China

Chenglian Liu
Department of Computer Science and Technology
Neusoft Institute Guangdong
Foshan 528228
China

T. E. Simos
College of Mathematics and Systems Science
Shandong University of Science and Technology
Qingdao 266590
China

and

South Ural State University
76, Lenin Aven.
Chelyabinsk
Russia 454080

and

Group of Modern Computational Methods
Ural Federal University
19 Mira st.
620002 Yekaterinburg
Russian Federation

and

Data Recovery Key Laboratory of Sichuan Province
Neijiang Normal University
Dongtong Road 705
Neijiang 641100
China

and

Section of Mathematics, Department of Civil Engineering
Democritus University of Thrace
Xanthi
Greece
e-mail: tsimos.conf@gmail.com

I. Th. Famelis
microSENSES Laboratory, Department of Electrical and Electronic Engineering
University of West Attica
Athens
Greece

Received: March 22, 2019.
Revised: June 9, 2019.
Accepted: December 2, 2019.