IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 15, 2021, accepted March 26, 2021, date of publication April 13, 2021, date of current version May 13, 2021.
Digital Object Identifier 10.1109/ACCESS.2021.3072944

Adaptive Polynomial Method for Solving Third-Order ODE With Application in Thin Film Flow

MOHAMMAD T. HAWEEL[1,2], O. ZAHRAN[2], AND FATHI E. ABD EL-SAMIE[2,3], (Member, IEEE)
[1]Department of Electrical Engineering, Shaqra University, Riyadh 11911, Saudi Arabia
[2]Department of Electronics and Electrical Communications Engineering, Faculty of Electronic Engineering, Menoufia University, Menouf 32951, Egypt
[3]Department of Information Technology, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh 21974, Saudi Arabia

Corresponding author: Mohammad T. Haweel (mhwail@su.edu.sa)

ABSTRACT Differential equations are commonly used to model several engineering, science, and biological applications. Unfortunately, finding analytical solutions for solving higher-order Ordinary Differential Equations (ODEs) is a challenge. Numerical methods represent a leading candidate for solving such ODEs. This work presents an innovated adaptive technique that uses polynomials to solve linear or nonlinear third-order ODEs. The proposed technique adapts the coefficients of the polynomial to obtain an explicit analytical solution. A signed least mean square algorithm is exploited to enhance the adaptation process and decrease both computational requirements and time. The efficiency of the proposed Adaptive Polynomial Method (APM) is illustrated through six well-known examples. The proposed technique is compared with recent analytical and numerical methods to validate its effectiveness in terms of Mean Square Error (MSE) and computation time. An application in a thin film flow system is modeled to a third-order ODE. The proposed technique is compared with recent numerical and analytical methods in solving the thin film flow equation, and it achieves better results. Furthermore, the proposed technique provides an analytical solution with an increased dynamic range and much lower computational time than those of the conventional numerical methods.

INDEX TERMS Adaptive algorithms, analytical solution, numerical solution, ordinary differential equations, polynomials.

I. INTRODUCTION

Several engineering, chemical, science, and biological applications can be modeled, analyzed and/or solved using ordinary differential equations (ODEs). Simple first-order ODEs or more complex higher-order ODEs can be used to solve several problems in robotics, nuclear magnetic response, fluid equations, analysis of electrical circuits, and other applications [1]–[3]. The ODEs can be found in different forms including partial and algebraic forms [4], [5]. Fractional-order differential equations have been proposed for modeling of modern and classical dynamical systems [6].

The absence of exact solutions is a challenging problem in most of these applications. Consequently, numerical methods represent the best way in solving such problem. Unfortunately, finding an analytical solution to solve such problem is a very difficult task, which calls for the need to find effective solutions for the ODEs. Euler, Runge-Kutta, Kepler, and Laplace transform are well-known numerical solution methods for ODEs [7]. Increasing the accuracy of such methods in finding analytical solutions triggers the interest of several researchers.

Recently, the predictor corrector method has been found to be inefficient because of the time-expensive process of developing separate predictors. In addition, the predictor order of accuracy is low [8]. A block method was proposed later to solve some problems of the predictor corrector method. However, the main drawback of the block method is the requirement of a large number of points to get a reasonable solution [9].

Initial Value Problems (IVPs) are solved using a hybrid block method, which has shown better stability properties [10]. In order to solve the general third-order ODEs, variable-step and modified Runge-Kutta methods were presented [11], [24]. Although these methods have shown reasonable approximations for such solutions, the required

The associate editor coordinating the review of this manuscript and approving it for publication was Bing Li.

computational time is large and the mathematical modeling is complicated.

Recently, semi-explicit and semi-implicit multi-step integration methods have been addressed. Although, these methods provided good convergence properties with less computational cost, they suffer from a decrease of the numerical stability with the increase of the accuracy order of the applied scheme. A semi-implicit multi-step extrapolation scheme was used to solve the stability problems with less computational cost [12].

Geometric integrators can be considered as a class of numerical integrators that are designed for Hamiltonian ordinary differential equations, and numerical schemes are designed to preserve the geometric properties. Possible applications might include dynamical systems, energy and phase space preservation, angular momentum, symplectic structure and symmetries [13].

One of the common methods in geometric integrators is the composition method [14]. It is based on the composition of several simpler integrators of the problem in order to increase the degree of accuracy of the ODE solver. Fourth-order composition methods for the numerical integration of IVPs defined by ODEs for dynamical systems were proposed [14].

Some other analytical tools were proposed such as the Differential Transform Method (DTM). The advantage of such tools is the ability of application to linear and nonlinear ODEs without linearization or discretization [15]. Furthermore, a Domain Decomposition Method (ADM) was recently proposed in [16].

Artificial Neural Networks (ANNs) have been effectively used as a strong tool to deal with the differential equations. The ANNs have been used for solving ordinary and partial differential equations. Different ANN types such as Chebyshev neural networks and Legendre neural networks were introduced in [17], [18]. Power series analysis has been recently presented as an important tool to find the analytical solutions of differential algebraic equations [19].

In this paper, an adaptive analytical solution for solving IVPs of the third-order ODEs is proposed. Such solution has the advantages of minimizing the drawbacks that occur with the majority of both analytical and numerical solutions, and providing closed-form solutions that are closer to the exact solution in a shorter time. Furthermore, the proposed technique adapts the coefficients of polynomials using an efficient adaptive algorithm. The limitation of the proposed technique is the lower accuracy obtained at the first few epochs due to the fact that the first four weights are not adapted in the APM.

The structure of this paper is as follows. The analysis of the proposed APM is presented in Section II. The simulation and numerical results are reported in Section III for comparison purposes. The effectiveness of providing an analytical solution in solving thin film flow equation is discussed in Section IV. Finally, conclusions are given in Section V.

II. PROPOSED ADAPTIVE POLYNOMIAL METHOD

The third-order IVP ODE can be considered as:

$$\begin{aligned} S'''(x) &= f(x, S(x), S'(x), S''(x)) \\ S(0) = C_0,\quad & S'(0) = C_1,\ S''(0) = C_2 \\ 0 \le x \le X,\quad & f : [x_0, X] \times \Re^n \to \Re^n \end{aligned} \tag{1}$$

where the constants C_0, C_1, and C_2 represent the initial values. In the proposed APM, the estimated solution of the third-order ODE, $\hat{S}(x)$, can be presented with the M^{th} order polynomial as follows:

$$\hat{S}(x) = \sum_{j=0}^{M-1} w_j x^j \tag{2}$$

After differentiation, the estimated solution is then given by:

$$\hat{S}'(x) = \sum_{j=1}^{M-1} j w_j x^{j-1} \tag{3}$$

$$\hat{S}''(x) = \sum_{j=2}^{M-1} j(j-1) w_j x^{j-2} \tag{4}$$

$$\hat{S}'''(x) = \sum_{j=3}^{M-1} j(j-1)(j-2) w_j x^{j-3} \tag{5}$$

From (2) and (5), the initial values are directly given by:

$$\hat{S}(0) = w_0;\quad \hat{S}'(0) = w_1;\ \hat{S}''(0) = 2w_2 \tag{6}$$

It can be noted from (6) that the initial conditions are easily defined. The proposed APM uses adaptive Least Mean Square (LMS) algorithm to find the analytical solution. The domain $[x_0, X]$ is then discretized. In order to update the adaptive system equation, the error function associated with the discrete step, x_k, is expressed, using (1), as:

$$e(x_k) = \hat{S}'''(x) - f\left(x, \hat{S}(x), \hat{S}'(x), \hat{S}''(x)\right) \tag{7}$$

Using (5), (7) can be rewritten as:

$$e(x_k) = \sum_{j=3}^{M-1} j(j-1)(j-2) w_j x^{j-3} - f\left(x, \hat{S}(x), \hat{S}'(x), \hat{S}''(x)\right) \tag{8}$$

In the proposed APM, for each training step x_k, a weight w_k is adapted. However, the first three weights ($k = 0, 1, 2$) are freezed (not adapted) at the boundary conditions according to (6). The proposed performance index J is:

$$J = \frac{1}{2} \sum_{k=3}^{L-1} e^2(x_k) \tag{9}$$

where L is the number of discrete steps involved in the adaptation. The k^{th} weight adjustment factor is:

$$\Delta w_k = -\mu \frac{\partial J}{\partial w_k} \tag{10}$$

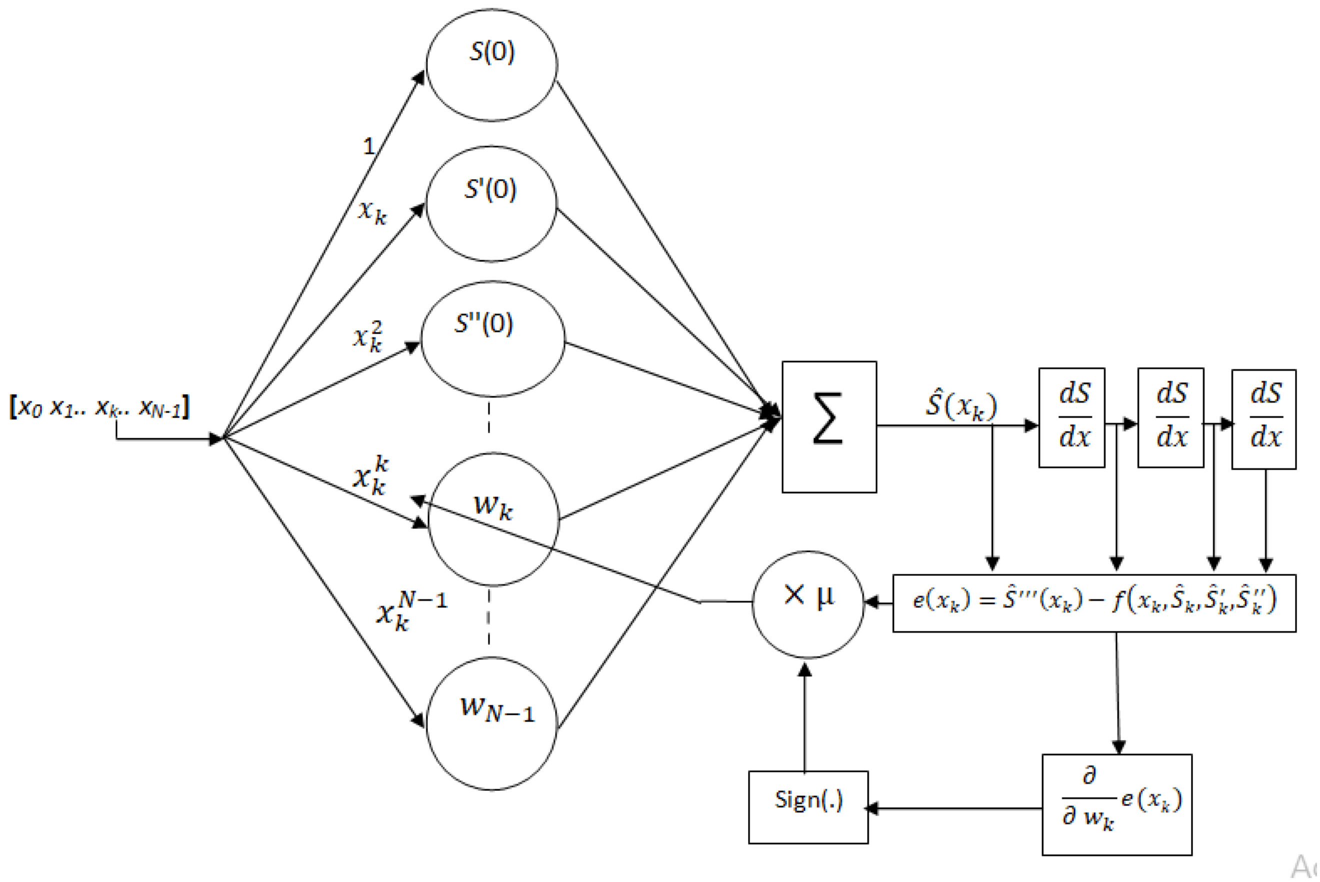

FIGURE 1. Schematic diagram for the proposed APM.

where the adaptation constant is μ. The time-consuming multiplication has been replaced by the low-cost sign change operation using the signed LMS [20]–[22]. In the APM, the update equation for the k^{th} weight is:

$$w_k(m+1) = w_k(m) - \mu\ e(k)\ sign\left\{\frac{\partial e(x_k)}{\partial w_k}\right\} \tag{11}$$

where the iteration cycle index is m, and $k \in [3, L-1]$. When all training steps are fully utilized, an iteration cycle is completed. The convergence is obtained, when the calculations reach a minimum MSE, which is accomplished via repeated cycles. The MSE of the iteration cycle, m, is computed as:

$$MSE(m) = \frac{1}{L}\sum_{k=3}^{L-1} e^2(x_k) \tag{12}$$

Signum function is known and formulated as [20]–[22]:

$$sign(U) = \begin{cases} 1 & U > 0 \\ 0 & U = 0 \\ -1 & U < 0 \end{cases} \tag{13}$$

For several problems, $\{\partial e(x_k)/\partial w_k\}$ in (11) is positive definite. Consequently, the weight update equation can be expressed as follows:

$$w_k(m+1) = w_k(m) - \mu\ e(x_k) \tag{14}$$

The schematic diagram that represents the proposed APM is shown in Fig. 1.

III. SIMULATION AND NUMERICAL RESULTS

An infinitesimal step of 0.1 is adopted for the discretization of the domain. The proposed APM employs the seventh-order polynomial. The assigned weights associated with the null, the first, and the second powers of the variable x, *i.e.* 1, x and x^2, have been freezed (not adapted) at the initial conditions equivalent to $S(0)$, $S'(0)$, and $S''(0)/2$, according to (6). In the same sequence, the assigned weight associated with x^3 is freezed at $S'''(0)/6$ using (5). Consequently, The APM estimated solution is derived by:

$$\hat{S}(x) = S(0) + S'(0)x + \frac{S''(0)}{2}x^2 + \frac{S'''(0)}{6}x^3 + \sum_{k=4}^{7} w_k x^k \tag{15}$$

The first four epochs 0.0, 0.1, 0.2, and 0.3 are not considered within the adaptation process due to their high dependence on the freezed weights. Hence, for each of the

TABLE 1. Comparison between the absolute errors in both the LS-SVM method and the proposed APM for the first example.

x	Exact solution	LS-SVM solution	APM solution	Error in LS-SVM×10^{-04}	Error in APM×10^{-06}
0.0915	0.091372377	0.090633366	0.091372351	7.3901	0.025086393
0.1518	0.151217677	0.150527278	0.151217566	6.904	0.110485277
0.2410	0.238673845	0.238098672	0.238673451	5.7517	0.393579494
0.3604	0.352648564	0.352301106	0.352647723	3.474	0.840670589
0.5000	0.479425539	0.479425539	0.479426562	6.2766	1.023895796
0.6395	0.596794319	0.597180775	0.596810109	3.8646	15.79035361
0.7590	0.688196265	0.688892884	0.688256066	6.9662	59.80159002
0.8482	0.750091219	0.750973103	0.750225365	8.8188	134.1459582
0.9084	0.788520736	0.789490968	0.788739198	9.7023	218.4623756
1.0000	0.841470985	0.842497209	0.841900000	0.1026	429.0151921

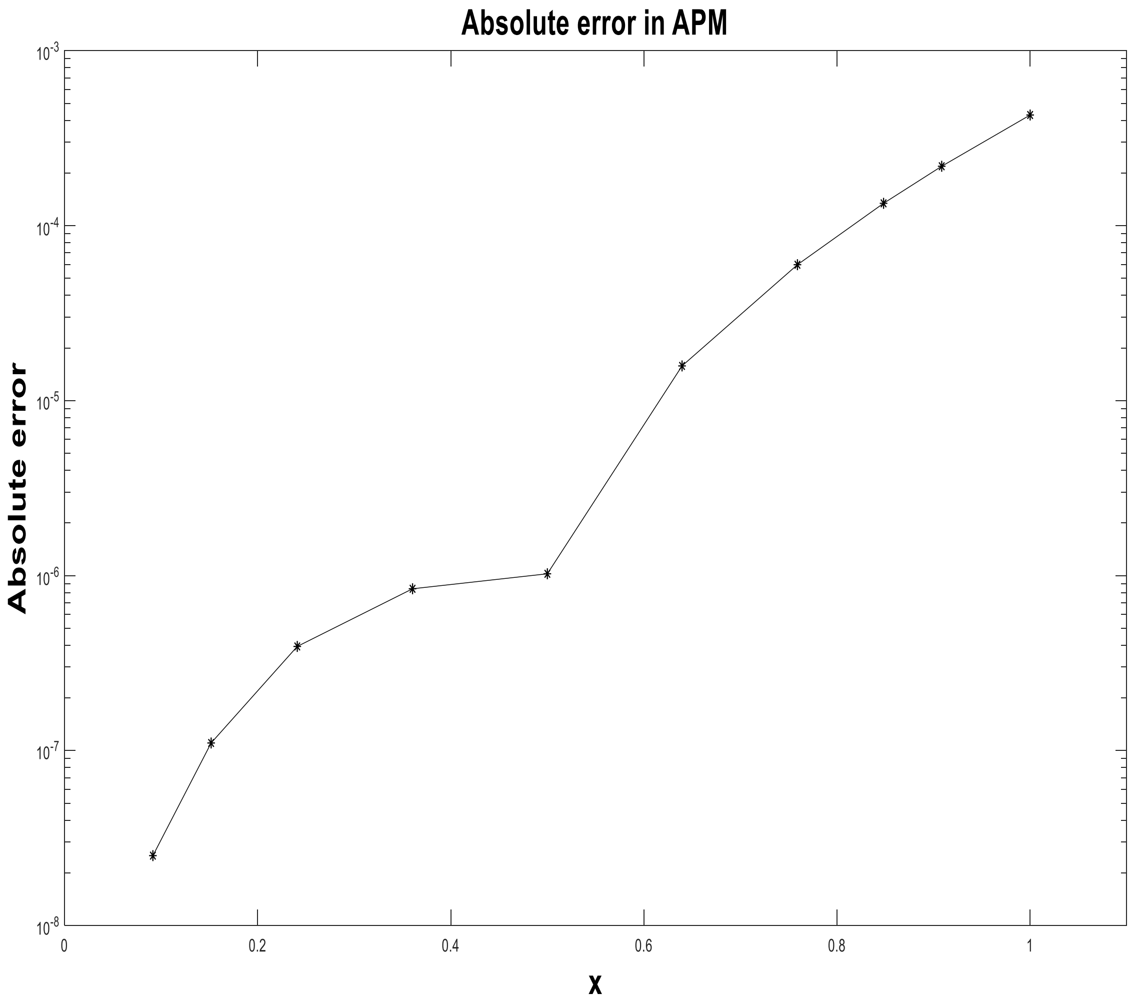

FIGURE 2. Absolute error values using APM compared to the exact solution for the first example.

discretized steps of 0.4, 0.5, 0.6 and 0.7, a one-to-one weight is adapted. This is one of the major advantages of the proposed APM over other methods mandating the employment of all discrete values in the domain. Such adapted weights are initially set to zero. Therefore, it is easy to prove that $\{\partial e(x_k)/\partial w_k\}$ has a positive definite value, and consequently, (14) is directly utilized.

In order to prove the efficiency of the proposed APM, the following six well-known examples are introduced.

A. FIRST EXAMPLE

The first example is the nonlinear ODE problem reported in [23], [24], which can be summarized as follows:

$$S''' = -S^2 - \cos(x) + \sin^2(x), \quad x \in [0,\ 1]; \\ S(0) = 0, \quad S'(0) = 1,\ S''(0) = 0 \tag{16}$$

It can be mentioned that the exact or analytical solution for this problem can be obtained by:

$$S_{exact}(x) = \sin(x), \quad x \in [0,\ 1] \tag{17}$$

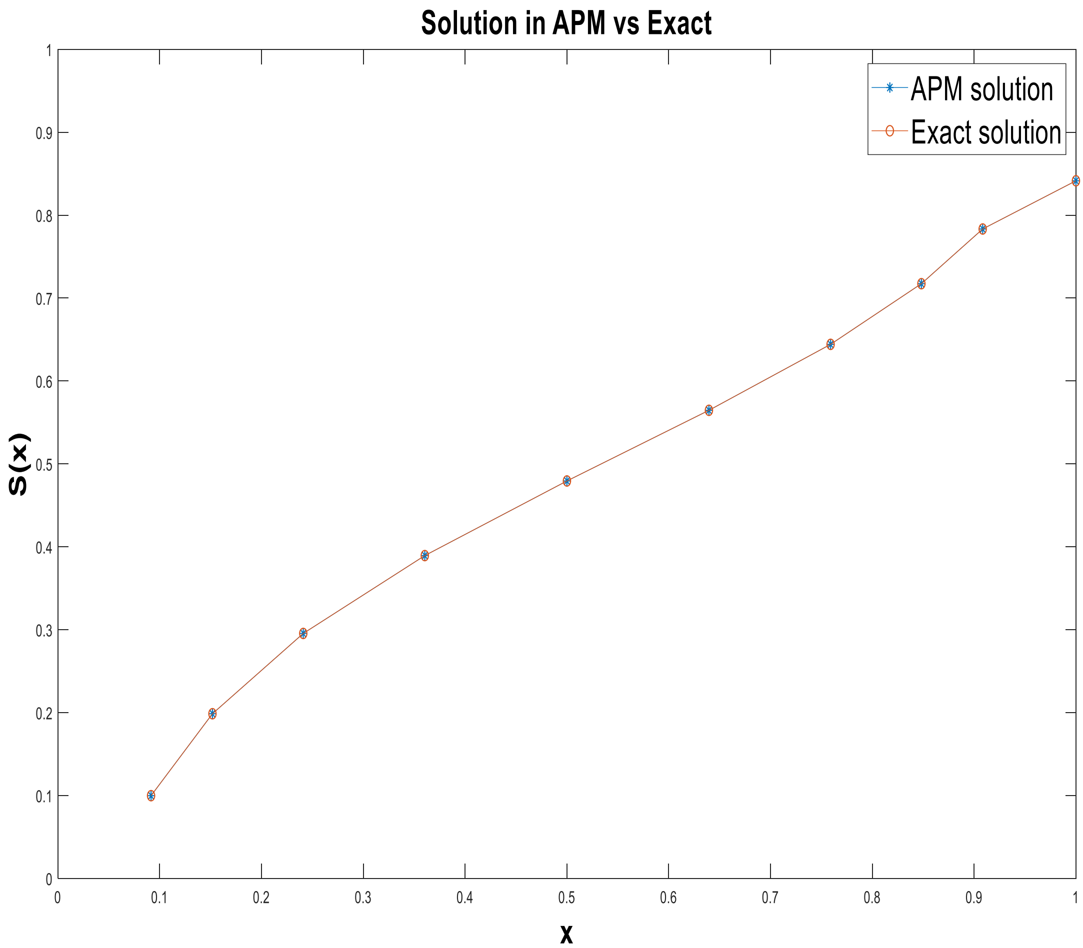

FIGURE 3. Exact and estimated solution using APM in the first example.

The estimated APM analytical solution, after convergence, can be obtained using (15) with $S'''(0) = -1$, as follows:

$$S_{APM}(x) = x - 0.1667\, x^3 + 0.0084\, x^5 + 0.0002\, x^7 \quad (18)$$

The norms of the computed errors between the exact and the estimated solutions using both the Least Square Support Vector Machines (LS-SVMs) [23] and the proposed APM, at the discretized points of interest, are tabulated in Table 1.

The absolute error values between the estimated solution using the proposed APM and the exact solution at all discretized points of the first example are shown in Fig. 2.

The obtained MSE values for the proposed APM after convergence and the LS-SVM are 2.5312×10^{-8} and 4.3564×10^{-7}, respectively, which indicates the good performance of the proposed method. Moreover, the CPU time is only 1.6406 s. The same problem has been solved numerically using a fourth-order improved Runge–Kutta technique [24], but the analytical APM solution has proved sufficient results.

Fig. 3 shows excellent agreement between the APM estimated solution and the exact solution of the ODE, with better precision than that of the LS-SVM.

B. SECOND EXAMPLE

The second example is the nonlinear ODE problem reported in [10], and [25], which can be summarized as follows:

$$\begin{aligned} S''' &= S'(2xS'' + S'), \quad : x \in [0, 1]; \\ S(0) &= 1, \quad S'(0) = 1/2, \; S''(0) = 0 \end{aligned} \quad (19)$$

It can be mentioned that the exact or analytical solution for this problem can be obtained by:

$$S_{exact}(x) = 1 + \frac{1}{2} \ln\left(\frac{2+x}{2-x}\right), \quad x < 2 \quad (20)$$

The same conditions of the first example have been undertaken. Using (15), the APM estimated analytical solution, after convergence, is:

$$\begin{aligned} S_{APM}(x) = 1 &+ 0.5x + 0.04167\, x^3 - 0.00160\, x^4 \\ &+ 0.010626\, x^5 - 0.005215\, x^6 + 0.003657\, x^7 \end{aligned} \quad (21)$$

Fig. 4 shows that the APM estimated solution is very close to the exact solution of the ODE compared to that of the hybrid method with block extension reported in [10].

TABLE 2. Comparison between the absolute errors in both the 2-step 4-point hybrid method and the proposed APM for the second example.

x	Exact solution	2-Step 4-Point Hybrid Method solution	APM solution	Error in 2-Step 4-Point Hybrid Method	Error in APM×10^{-03}
0.1	1.0500417292784914	1.0500418242095606	1.050041607315773	0.49E-08	0.000121962718369
0.2	1.1003353477310756	1.1003366644736043	1.100333874561465	1.32E-06	0.001473169610522
0.3	1.1511404359364668	1.1511460842057299	1.151134798012412	5.65E-06	0.005637924054724
0.4	1.2027325540540821	1.2027483597246535	1.202718955297695	1.58E-05	0.013598756387179
0.5	1.2554128118829952	1.2554482979429176	1.255387014014122	3.55E-05	0.025797868872912
0.6	1.3095196042031119	1.3095893044089619	1.309477015186962	6.97E-05	0.042589016150174
0.7	1.3654437542713962	1.3655690060595540	1.365379274776617	1.25E-04	0.064479494779723
0.8	1.4236489301936017	1.4238603658481614	1.423556746430464	2.11E-04	0.092183763137799
0.9	1.4847002785940517	1.4850413496636110	1.484572688679046	3.41E-04	0.127589915005499
1.0	1.5493061443340548	1.5498382025385242	1.549127479775827	5.32E-04	0.178664558228236

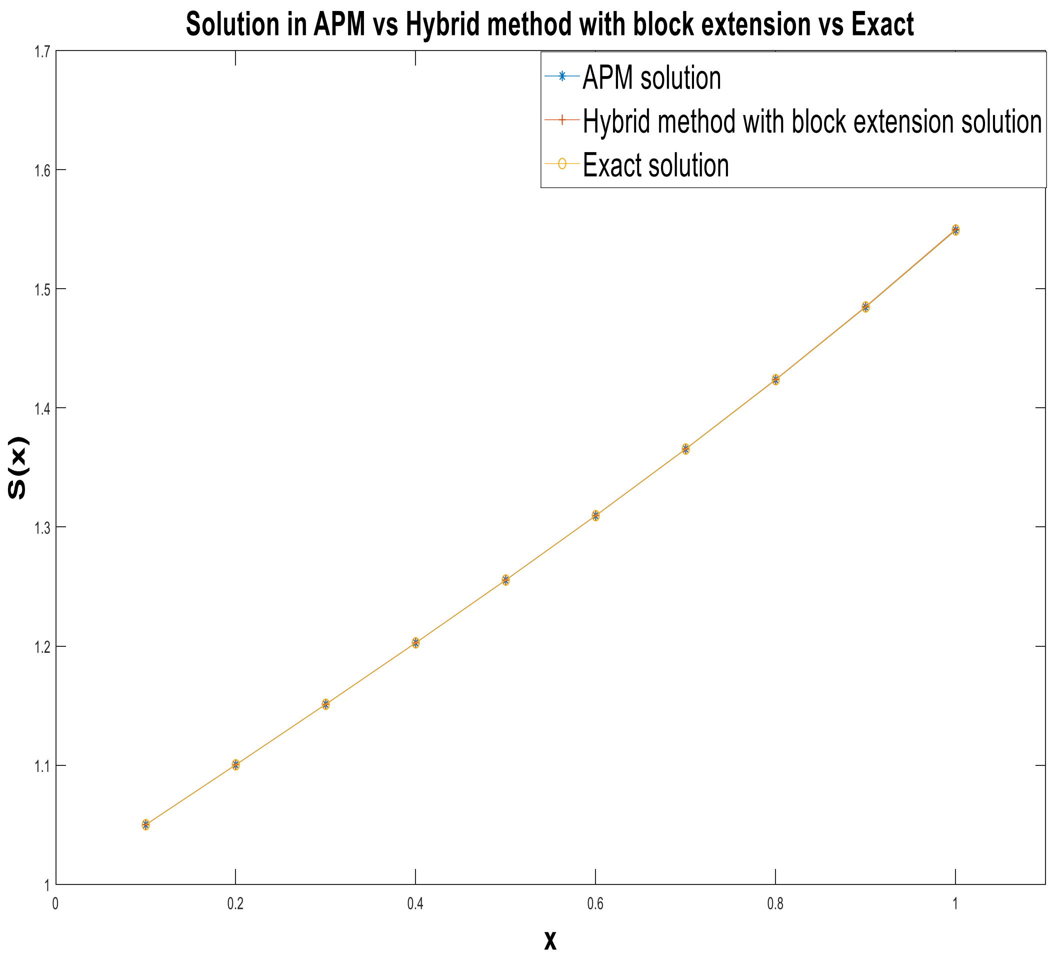

FIGURE 4. Exact and estimated solutions using APM and the hybrid method with block extension.

This example has a solution using the 2-step 4-point hybrid method addressed in [25]. The absolute values of the computed errors between the exact and the estimated solutions using the proposed APM and the method in [25], at the discretized points of interest, are tabulated in Table 2.

Fig. 5 shows a comparison of the absolute error values for the estimated solutions using the 2-step 4-point hybrid method, the proposed APM, and the exact solutions at all discretized points. As shown, the proposed APM achieves better accuracy with lower absolute error values, except for the first two discrete points, where the weights are freezed. Moreover, the CPU time is only 1.75 s.

C. THIRD EXAMPLE

The ODE problem in [15] is reproduced in the third example. It can be summarized as follows:

$$S''' = e^x \quad x \in [0, 1] \; ; \\ S(0) = 3, \quad S'(0) = 1, \; S''(0) = 5 \tag{22}$$

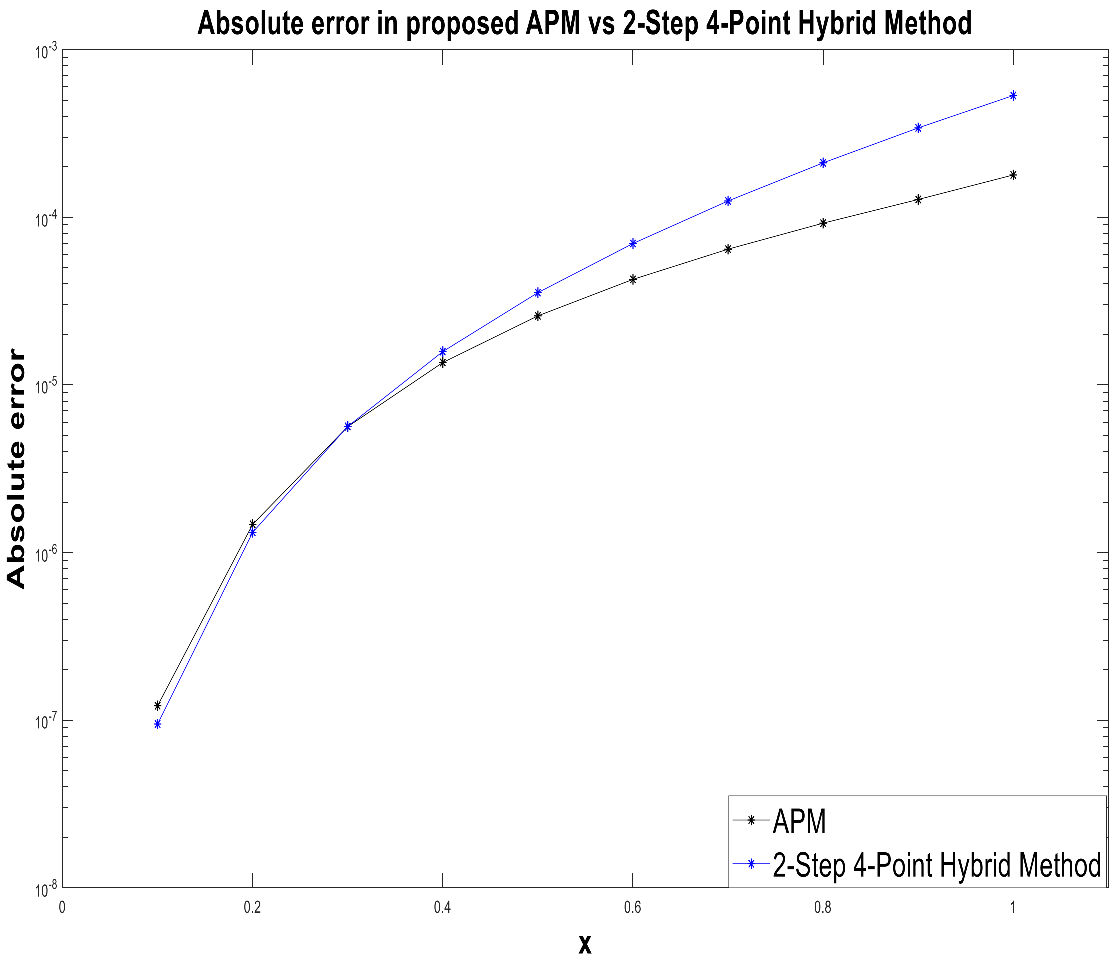

FIGURE 5. Absolute error values using the APM and the 2-Step 4-point hybrid method compared to the exact solution for the second example.

TABLE 3. Comparison between the absolute errors in both the DTMELZ method and the proposed APM for the third example.

x	Exact solution	DTMELZ solution	APM solution	Error in DTMELZ	Error in APM$\times10^{-05}$
0.1	3.125170918	3.150170918	3.125170914943767	0.0250	0.0003
0.2	3.301402758	3.401402759	3.301402721196208	0.1000	0.0037
0.3	3.529858807	3.754858807	3.529858669158602	0.2250	0.0138
0.4	3.811824697	4.211824698	3.811824370718431	0.4000	0.0327
0.5	4.148721270	4.773721270	4.148720663711130	0.6250	0.0607
0.6	4.542118800	5.442118801	4.542117821770927	0.9000	0.0979
0.7	4.993752707	6.218752707	4.993751265583418	1.2250	0.1442
0.8	5.505540928	7.105540929	5.505538931552585	1.6000	0.1997
0.9	6.079603111	8.104603111	6.079600453894901	2.0250	0.2657
1.0	6.718281828	9.218281828	6.718278316173190	2.5000	0.3512

It can be mentioned that the exact or analytical solution for this problem is obtained by:

$$S_{exact}(x) = 2 + 2x^2 + e^x \quad x \in [0, 1] \tag{23}$$

The same conditions of the previous examples have been considered. Using (15), the APM estimated analytical solution, after convergence, is:

$$S_{APM}(x) = 3 + x + 2.5\,x^3 + 0.1667\,x^4 + 0.0085\,x^5 + 0.0012\,x^6 + 0.0003\,x^7 \tag{24}$$

Fig. 6 shows that the APM estimated solution is more close to the exact solution of the ODE than that of the Differential Transform and Elzaki (DTMELZ) method [15].

The norms of the computed errors between the exact and estimated solutions, using both the DTMELZ method [15] and the proposed APM, at the discretized points of interest, are tabulated in Table 3. Moreover, the CPU time is only 0.8438 s.

D. FOURTH EXAMPLE

The ODE problem in [26] is reproduced in the fourth example. It can be summarized as follows:

$$S''' = -S, \quad x \in [0, 2];$$

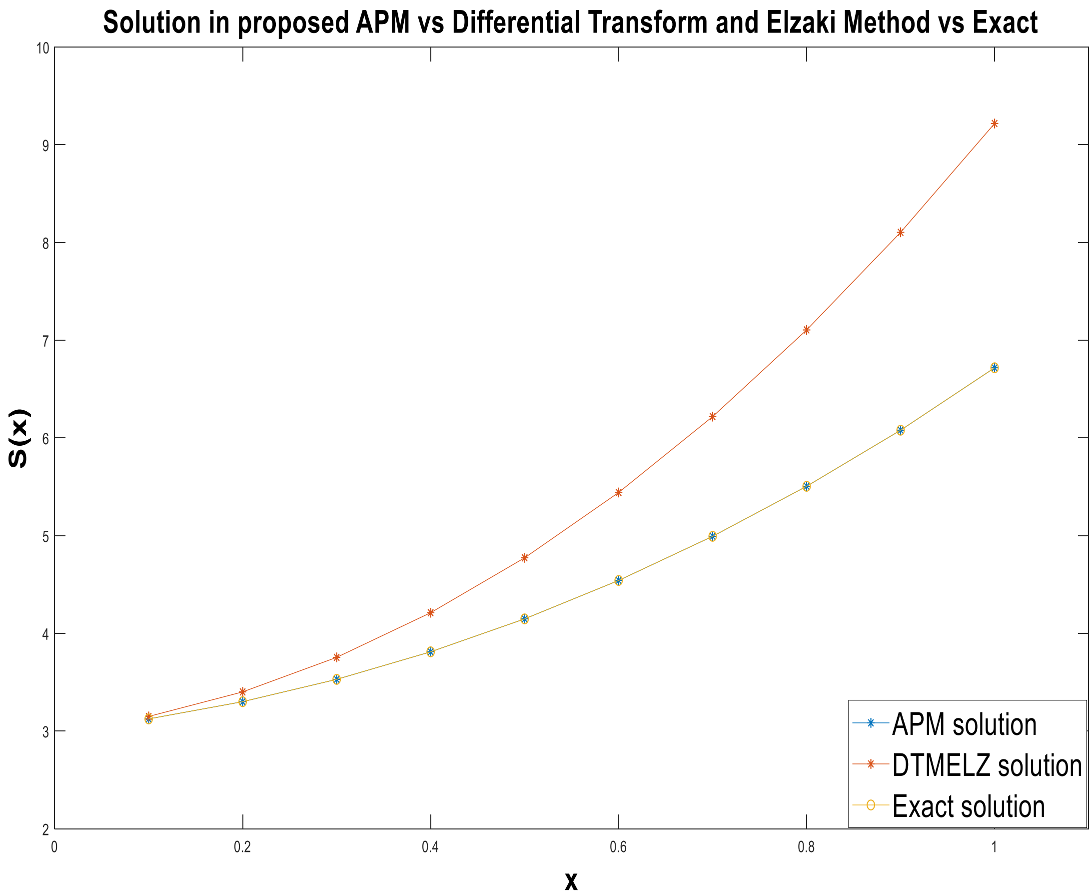

FIGURE 6. **Exact and estimated solutions using APM and DTMELZ method.**

$$S(0) = 1, \quad S'(0) = -1, \; S''(0) = 1 \tag{25}$$

It can be mentioned that the exact solution for such ODE can be obtained as follows:

$$S_{exact}(x) = e^{-x} \quad x \in [0, 2] \tag{26}$$

Considering $S'''(0) = 1$, the converged APM analytical solution can be estimated as follows:

$$S_{APM}(x) = 1 - x + 0.5\,x^2 - 0.1667\,x^3 + 0.0416\,x^4 - 0.00832\,x^5 + 0.0013\,x^6 - 0.0001x^7 \tag{27}$$

The estimated analytical solution using the DTMELZ method has the following form [26]:

$$S_{DTM}(x) = 1 - x + \frac{x^2}{2} - \frac{x^3}{6} + \frac{x^4}{24} - \frac{x^5}{120} + \frac{x^6}{720} - \frac{x^7}{5040} \tag{28}$$

The norms of the computed errors between the exact solution and the estimated solution using the DTMELZ method [26] and the proposed APM, at the discretized points of interest, are shown in Fig. 7.

In general, the proposed APM solution shows relatively small absolute errors compared to those of the DTMELZ method with a higher accuracy and 1.6094 s CPU time. On the other hand, the lower accuracy obtained at the first few epochs is attributed to the first four weights that are not adapted in the APM.

Considering the same example, the absolute values of the computed errors between the estimated solutions using both the ADM [27] for $x \in [0.2, 1]$ and the proposed APM, at the discretized points of interest, and the closed-form solution are shown in Fig. 8. Additionally, the proposed APM shows relatively small absolute errors compared to the ADM with higher accuracy and a 1.5894 s CPU time.

E. FIFTH EXAMPLE

The ODE problem in [27] is reproduced in the fifth example. It can be summarized as follows:

$$S''' + 2S'' - S' - 2S = e^x, \quad x \in [0, 3]$$
$$S(0) = 1, \quad S'(0) = 2, \; S''(0) = 0 \tag{29}$$

It can be mentioned that the exact solution for such ODE can be obtained as follows:

$$S_{exact}(x) = \frac{43}{36}e^x + \frac{1}{4}e^{-x} - \frac{4}{9}e^{-2x} + \frac{1}{6}xe^x \tag{30}$$

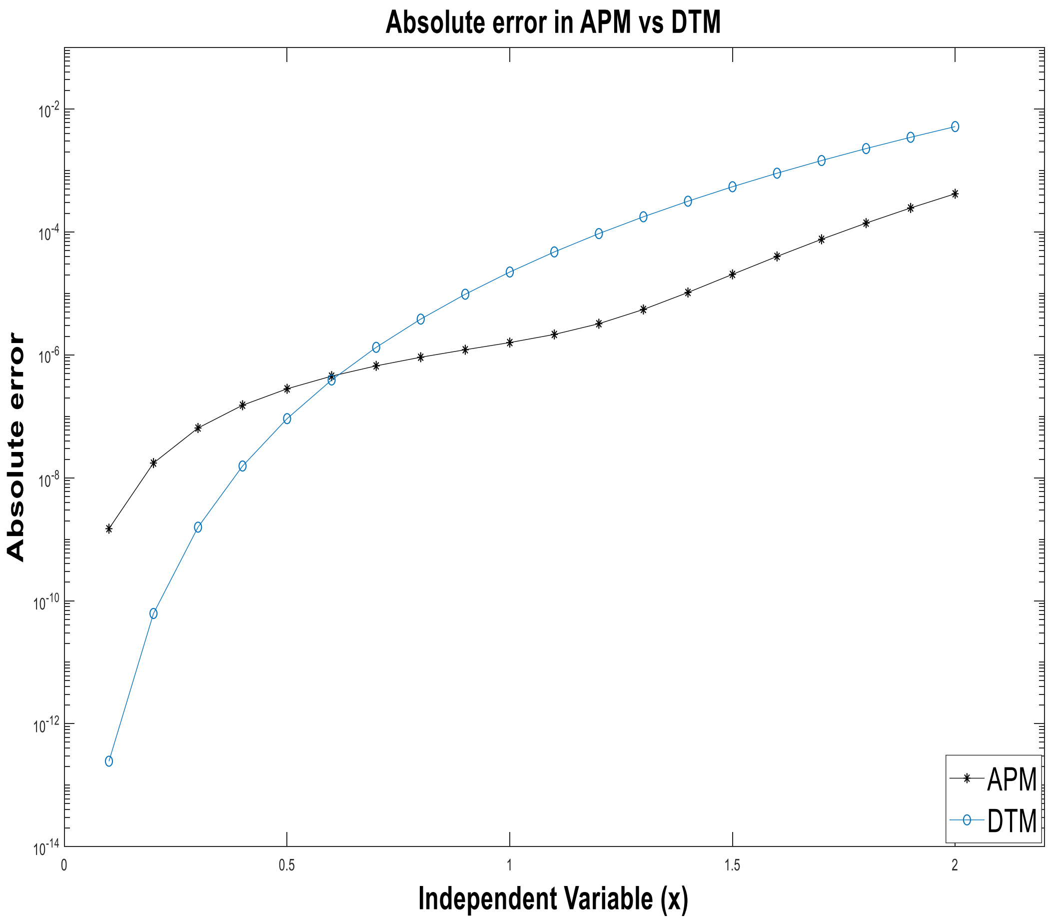

FIGURE 7. Absolute errors using the APM and the DTMELZ method for the fourth example.

Using (15) with $S'''(0) = 5$, the converged APM analytical solution can be estimated as follows:

$$S_{APM}(x) = 1 + 2x + 0.8333\,x^3 - 0.2070\,x^4 + 0.1287\,x^5 - 0.0288x^6 + 0.0055\,x^7 \quad (31)$$

The estimated analytical solution using the DTMELZ method has the following form [27]:

$$S_{DTMELZ}(x) = 1 + 2x + \frac{5}{6}x^3 - \frac{5}{24}x^4 + \frac{2}{15}x^5 - \frac{13}{360}x^6 + \frac{59}{5040}x^7 \quad (32)$$

Fig. 9 shows the absolute error values between the estimated solutions using both the DTMELZ method [27] and the proposed APM compared to the exact solutions, at all the discretized points. As shown, the proposed APM has lower absolute error values, which indicates better accuracy except for the first few discrete points, where the weights are freezed.

The proposed APM achieves an excellent agreement with the ODE exact solution in (30), compared with the DTMELZ method, with higher accuracy as shown in Fig. 10 with a 1.8802 s CPU time.

F. SIXTH EXAMPLE

The ODE problem in [28] is reproduced in the sixth example. It can be summarized as follows:

$$S''' - 4S'' + S' + 6x = 6, \quad x \in [0, 1]$$
$$S(0) = 3, \quad S'(0) = 5, \; S''(0) = 13 \quad (33)$$

The domain of this example was discretized using a step of 1/8. Using (15) and considering that $S'''(0) = 35$, the APM analytical solution, after convergence, can be estimated as follows:

$$S_{APM}(x) = 3 + 5x + 6.5\,x^2 + 5.8333\,x^3 + 3.4040\,x^4 + 4.4607\,x^5 - 1.7489\;x^6 + 2.0198\,x^7 \quad (34)$$

Table 4 summarizes the exact solution, the numerical solution using the fourth-order Runge-Kutta method, the numerical solution using the method thoroughly explained in [28], and the proposed APM solution, which can be obtained by (34) at the discretized intervals.

The absolute error values between the numerical solutions and the exact solutions are illustrated in Fig. 11, which shows that the proposed APM is better than the hybrid-domain one-shot integration matrices method [28]. It is well-known that the fourth-order Runge-Kutta solution is very close to the exact solution, and consequently, this method outperforms the proposed APM.

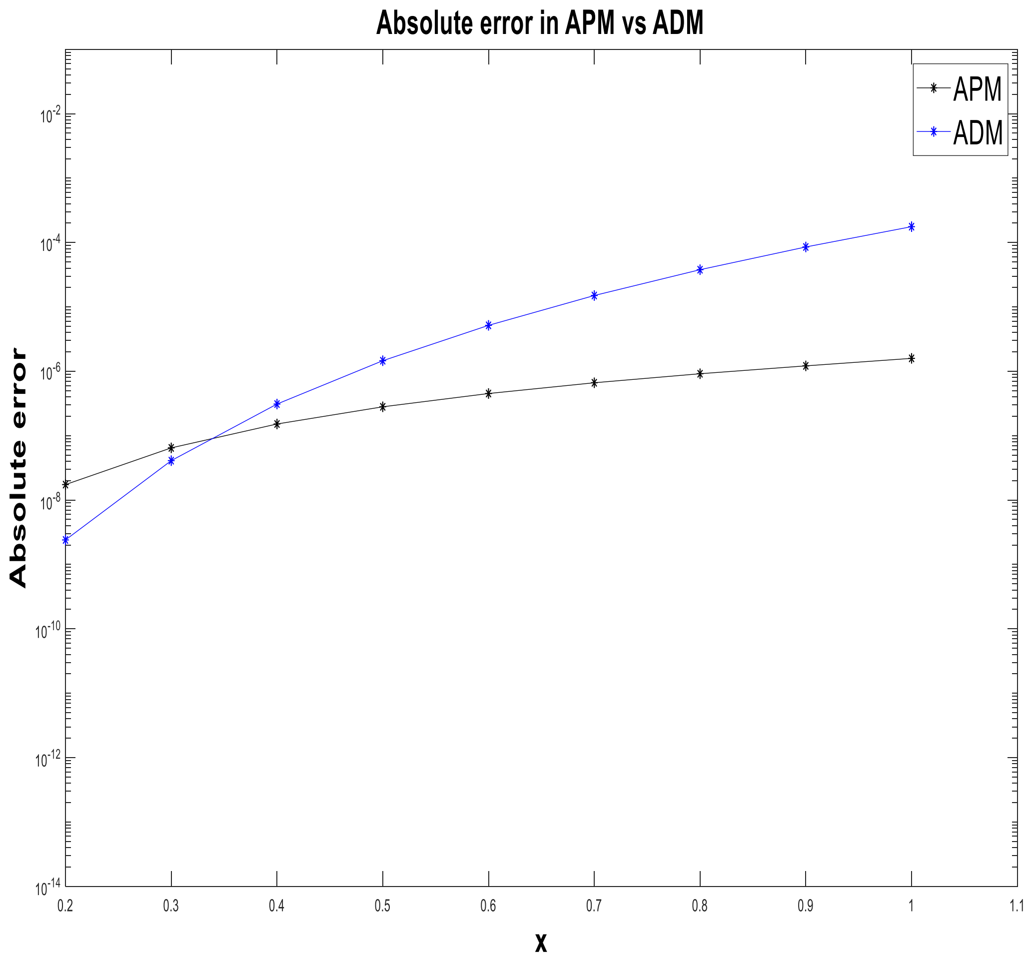

FIGURE 8. Absolute errors using APM and ADM for the fourth example.

TABLE 4. Exact, numerical and proposed APM solutions for the sixth example.

x	Exact	The Fourth-order Runge-Kutta	Method in [28]	Proposed APM
1/8	3.7390	3.7389	3.7521	3.7389
2/8	4.7657	4.7655	4.8223	4.7647
3/8	6.1972	6.1968	6.3546	6.1943
4/8	8.2000	8.1991	8.5617	8.1948
5/8	11.0112	11.0096	11.7579	11.0040
6/8	14.9694	14.9667	16.4098	14.9611
7/8	20.5592	20.5545	23.2102	20.5510
1	28.4746	28.4669	33.1910	28.4689

However, the advantage of the proposed APM is that it provides both analytical and numerical solutions, while the Runge-Kutta method provides only the numerical solution. Moreover, the CPU time is only 2.3906 s.

IV. APPLICATION OF APM TO THE THIN FILM FLOW PROBLEM

In this section, the APM performance is tested on a well-known application in engineering and physics (Thin Film Flow). The problem of thin film flow of a liquid on a certain solid surface has received a lot of attention [29], [30].

Commonly, most discussions related to this matter are about a viscous fluid flowing over a certain solid surface. Several parameters are taken into consideration, such as gravity, viscosity, and tension.

The flow of thin films of viscous fluid over a free surface can be modeled as a third-order ODE, which describes the shape of the free surface of the fluid and takes into consideration the surface tension effect.

One of the ODEs that deal with a fluid dynamic problem can be expressed as follows:

$$S''' = f(X) \tag{35}$$

where

$$f_1(X) = -1 + S^{-2}$$
$$f_2(X) = -1 + (1 + \delta + \delta^2)S^{-2} - (\delta + \delta^2)S^{-3}$$

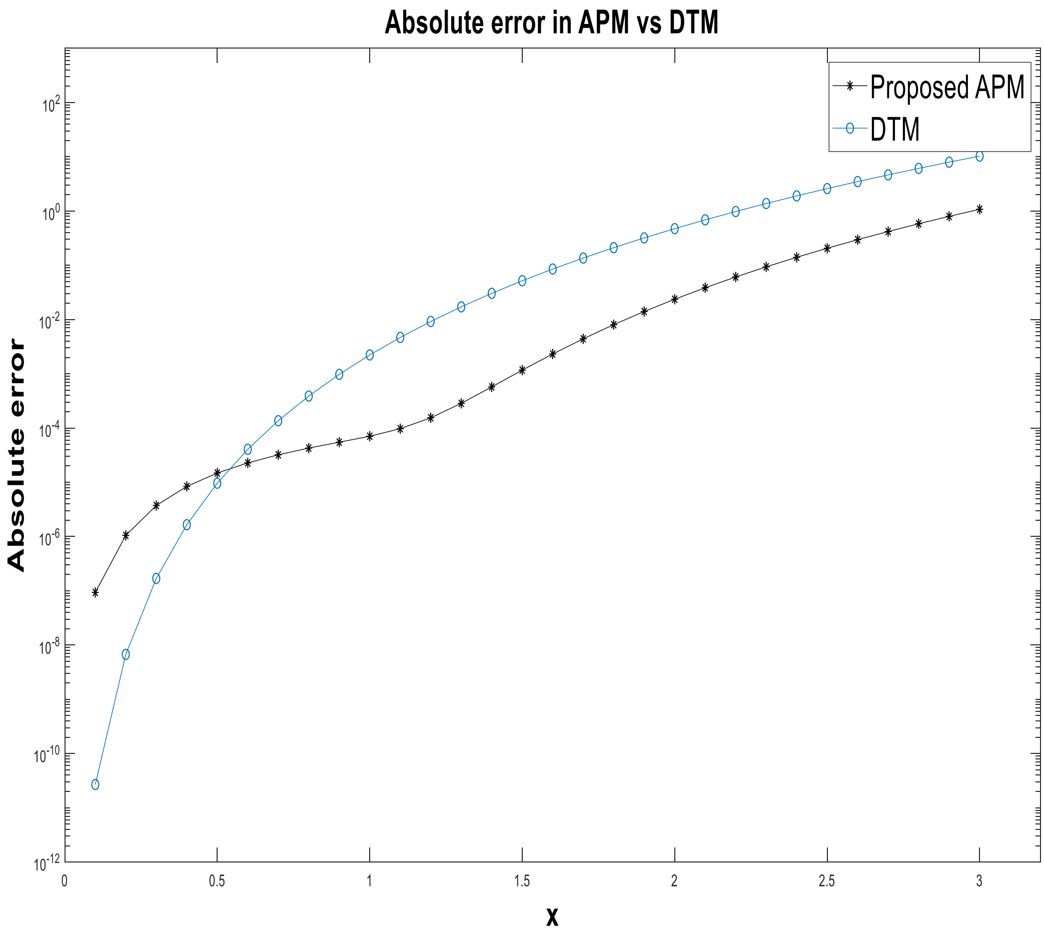

FIGURE 9. Absolute error values using the APM and the DTMELZ method [27] compared to the exact solution for the fifth example.

TABLE 5. Exact, proposed APM and other numerical solutions for the case of $k = 3$.

x	Exact	Solution 1 in [29]	Solution 2 in [29]	Solution 3 in [29]	Solution 4 in [30]	Proposed APM Solution
0	1.000000000	1.0000000000	1.000000000	1.000000000	1.000000000	1.000000000
0.2	1.221211030	1.2211551423	1.2211551424	1.2211551424	1.221162945	1.2211714901
0.4	1.488834893	1.4881052838	1.4881052842	1.4881052842	1.488166464	1.4882636134
0.6	1.807361404	1.8042625471	1.8042625481	1.8042625482	1.804459968	1.8047588269
0.8	2.179819234	2.1715227960	2.1715227981	2.1715227982	2.171968706	2.1725551662
1	2.608275822	2.5909582556	2.5909582591	2.5909582592	2.591788995	2.5926669956

TABLE 6. Exact, proposed APM, and domian decomposition method solutions [16] for the case of $k = 2$.

x	Exact Solution	Analytical ADM Solution [16]	Proposed APM Solution
0	1.000000000	1.0000000000	1.0000000000
0.2	1.221211030	1.2212104177	1.2212110341
0.4	1.488834893	1.4888655644	1.4888445970
0.6	1.807361404	1.8077660800	1.8073920350
0.8	2.179819234	2.1824139377	2.1798828162
1	2.608275822	2.6194444444	2.6083819601

$$f_3(X) = S^{-2} - S^{-3}$$
$$f_4(X) = S^{-2} \qquad (36)$$

Generally, (36) represents the draining of a thin film down on a dry wall, while $f_1(X)$ represents the drainage dry surface, and $f_2(X)$ is a prewetted surface by a thin film with a very small thickness $\delta > 0$.

Various numerical methods have been devoted to solve the thin film flow problem as a third-order ODE in a famous form as follows:

$$S''' = S^{-k} \qquad (37)$$
$$\text{with} \quad S(0) = x_1, \quad S'(0) = x_2, \quad S''(0) = x_3$$

where the initial conditions x_1, x_2 and x_3 are constants. In a thin fluid layer, the importance of such constants is that they

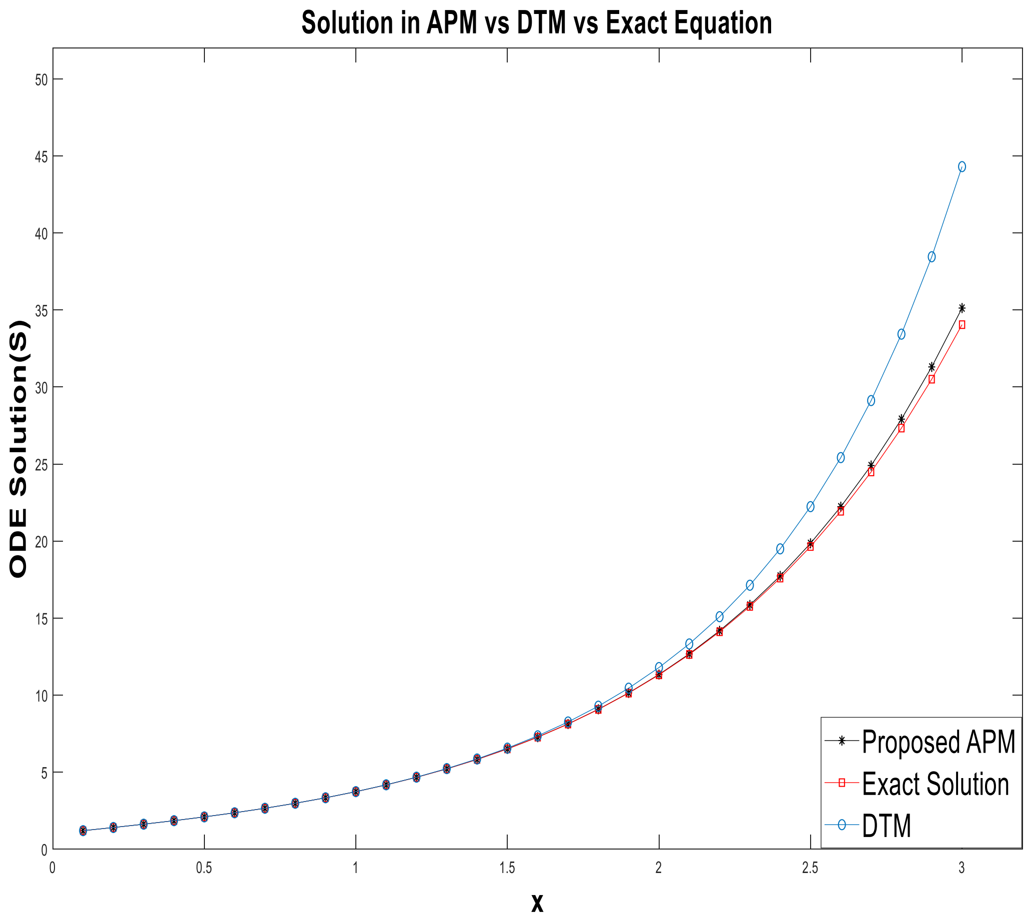

FIGURE 10. Exact and estimated solutions using APM and DTMELZ [27].

describe the dynamic balance between viscous forces and surface in the absence of gravity.

Considering the cases, where $x_1 = x_2 = x_3 = 1$, $k = 2$, and $k = 3$, the proposed APM is also applied to analytically solve (37). The results are summarized in Table 5 and Table 6 and compared with the results of all other numerical methods in [29], [30], for the case of $k = 3$, and the analytical method in [16] for the case of $k = 2$.

From the aforementioned comparison results, it can be noticed that the APM provides a better and closer solution to the exact one. In addition, it can produce an analytical solution valid for all values of x.

For the case of $k = 3$, the analytical solution by the proposed APM can be found as:

$$S_{APM}(x) = 1 + x + \frac{1}{2}x^2 + 0.166x^3 - 0.108317x^4 + 0.035086x^5 - 0.00482x^6 - 0.00558x^7 \quad (38)$$

The absolute values of the computed errors between the best-estimated numerical solutions [29], [30] and those of the proposed APM, at the discretized points of interest, compared to the closed-form solution are shown in Fig. 12. The CPU time is only 0.6527 s.

For the second case of $k = 2$, an analytical solution was found to solve the thin film flow problem in terms of the initial values using the ADM [16]. The comparison between the APM and the ADM confirms the efficiency of the proposed APM compared to both numerical and analytical methods.

The absolute values of the estimated errors of the proposed APM solution and the estimated analytical solution in [16], at the discretized points of interest, compared to the closed-form solution are shown in Fig. 13.

It can be noticed that the proposed APM produces relatively smaller absolute errors compared to the ADM. The CPU time is only 1.4843 s.

After convergence, the analytical solution by the proposed APM, for the case of $k = 2$, can be found as:

$$S_{APM}(x) = 1 + x + \frac{1}{2}x^2 + 0.166x^3 - 0.08224x^4 + 0.03045x^5 - 0.00725x^6 + 0.000762x^7 \quad (39)$$

Using the seventh-order ADM in [16] with the same initial conditions, the solution can be expressed as:

$$S_{ADM}(x) = 1 + x + \frac{1}{2}x^2 + 0.1667x^3 - 0.0833x^4 + 0.0333x^5 - 0.00694x^6 + 0.00833x^7 \quad (40)$$

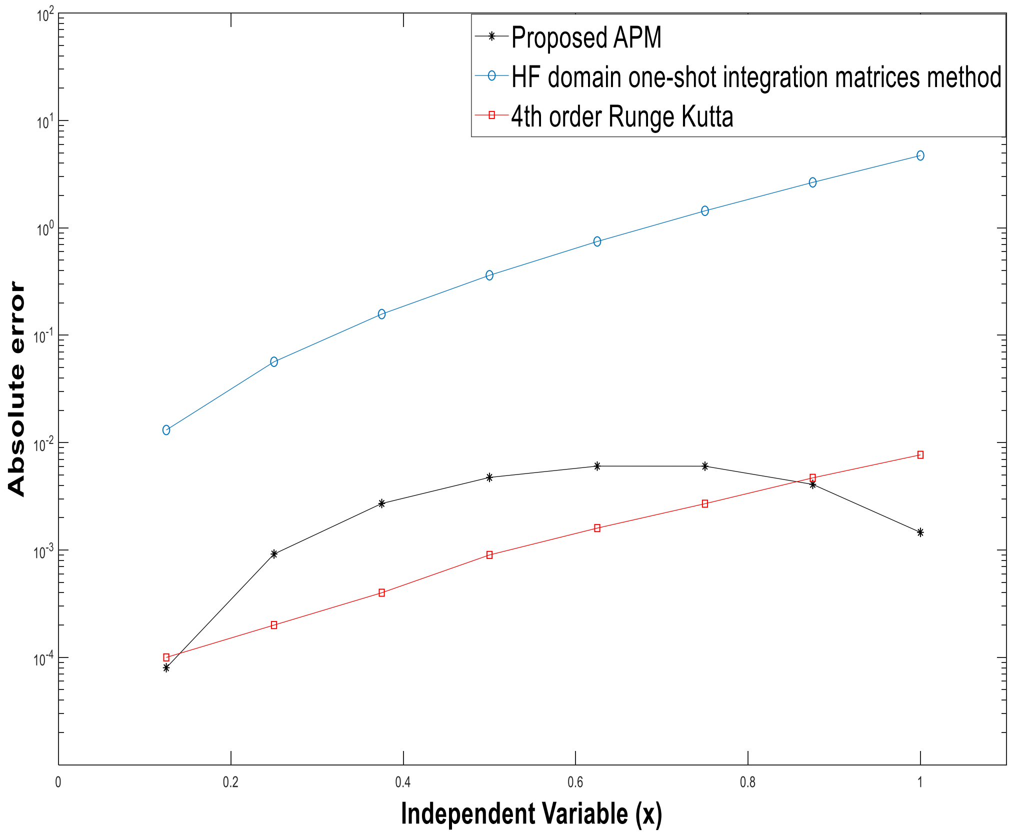

FIGURE 11. Absolute errors for the APM, the fourth-order Runge-Kutta and the hybrid domain one-shot integration matrices method solutions.

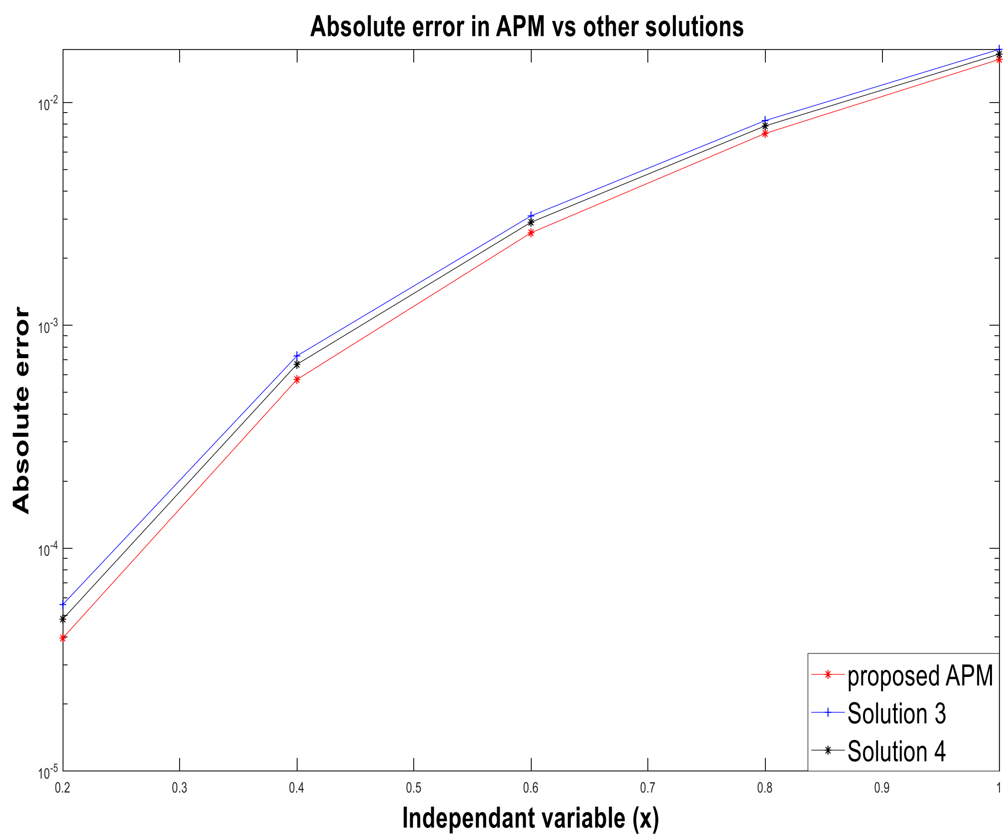

FIGURE 12. Absolute errors using the APM compared to solutions 3 and 4, for $k = 3$.

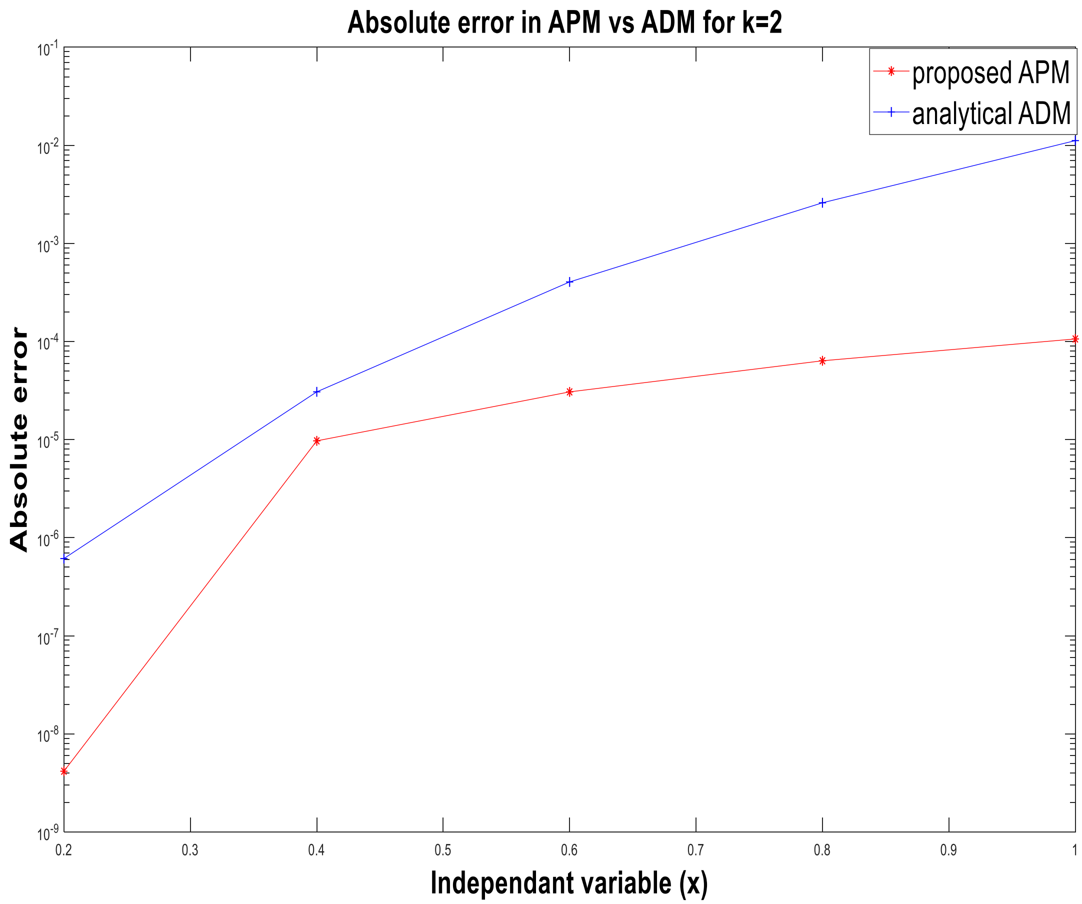

FIGURE 13. Absolute errors using the APM compared to the ADM, for $k = 2$.

The efficiency of the proposed APM in the thin film flow problem has been emphasized with both numerical and analytical solutions.

V. CONCLUSION

An efficient adaptive polynomial method (APM), which can provide an analytical solution for linear and nonlinear third-order ODEs, has been presented. By employing this APM, analytical solutions could be derived, even when only numerical solutions may be available. The APM adapts the coefficients of polynomials using the efficient adaptive SLMS algorithm. Low computational complexity can be achieved by employing the adaptive SLMS algorithm due to the simple update equation. Moreover, regarding the APM analytical solution, the initial conditions can be easily found. The improved performance of the APM is demonstrated through the comparison with some powerful conventional analytical and numerical methods through six famous examples, and tested in an application of a thin film flow problem as well.

In the first example, the proposed APM outperforms the LS-SVM method, where the mean square error has dramatically dropped one order of magnitude, from 4.3564×10^{-7} to 2.531×10^{-8}. The obtained results for the second example are similar; the overall mean square error has dropped one order of magnitude, from 4.65853×10^{-8} to 6.35539×10^{-9}.

The third example has definitely confirmed the high efficiency of the proposed APM by severely dropping mean square error twelve orders of magnitude, from 1.5833125 to 2.69154×10^{-12}.

Regarding the fourth and the fifth examples, although the obtained results of the proposed APM show higher absolute error values only at the few epochs where the weights are freezed, lower absolute error values are obtained everywhere else, which indicates a better accuracy of the proposed APM.

Regarding the sixth example, although the proposed APM shows the same order of magnitude of the fourth-order Runge-Kutta method, its results are superior, where the order of magnitude dropps, approximately five orders of magnitude, from 4.007939979 to 3.2115×10^{-5}. Moreover, all utilized methods for computing the results of the sixth example show a semi-identical mean square error that has an average of 0.004852099 and a standard deviation of 0.000128587, while the proposed APM shows a slightly smaller mean square error of 0.004347715.

Similarly, in the application to the thin film flow problem, the proposed APM shows one order of magnitude smaller mean square error of -3.50099×10^{-5} compared to that of the solution using the analytical ADM of 0.002366344 for the case of $k = 2$. For $k = 3$, the proposed APM shows slightly better results for all epochs.

The detailed explanation of all six examples and the application to the thin film flow problem and their corresponding results, definitely, illustrate and confirm the superiority of the proposed APM in terms of accuracy, simplicity and processing time.

Future work of this research could involve testing the performance of the APM in 2-dimensional nonlinear problems, differential algebraic equations and fractional-order ODEs. Different adaptive algorithms could be tested to increase the accuracy of the results. Comparison between the proposed method and other methods in solving higher-order ODEs will be studied. Finally, more practical applications could be used to test the performance of the APM.

REFERENCES

[1] M. P. Beccar-Varela, A. M. Bhuiyan, M. C. Mariani, and O. K. Tweneboah, "Analytic methods for solving higher order ordinary differential equations," *Mathematics*, vol. 7, no. 9, pp. 1–17, 2019.
[2] H. Singh and H. M. Srivastava, "Numerical simulation for fractional-order Bloch equation arising in nuclear magnetic resonance by using the Jacobi polynomials," *Appl. Sci.*, vol. 10, no. 8, p. 2850, Apr. 2020.
[3] H. Singh and H. M. Srivastava, "Numerical investigation of the fractional-order Liénard and duffing equations arising in oscillating circuit theory," *Frontiers Phys.*, vol. 8, p. 120, Apr. 2020.
[4] Z. Sabir, H. Günerhan, and J. L. G. Guirao, "On a new model based on third-order nonlinear multisingular functional differential equations," *Math. Problems Eng.*, vol. 2020, pp. 1–9, Feb. 2020.
[5] H. Singh, F. Akhavan Ghassabzadeh, E. Tohidi, and C. Cattani, "Legendre spectral method for the fractional Bratu problem," *Math. Methods Appl. Sci.*, vol. 43, no. 9, pp. 5941–5952, Jun. 2020.
[6] H. Singh and H. M. Srivastava, "Jacobi collocation method for the approximate solution of some fractional-order Riccati differential equations with variable coefficients," *Phys. A, Stat. Mech. Appl.*, vol. 523, pp. 1130–1149, Jun. 2019.
[7] D. Rani and V. Mishra, "Numerical inverse Laplace transform based on Bernoulli polynomials operational matrix for solving nonlinear differential equations," *Results Phys.*, vol. 16, Mar. 2020, Art. no. 102836.
[8] J. G. Oghonyon, S. A. Okunuga, and S. Bishop, "A 5-step block predictor and 4-step corrector methods for solving general second order ordinary differential equations," *Global J. Pure Appl. Math.*, vol. 11, no. 5, pp. 3847–3862, 2015.
[9] B. S. H. Kashkari and S. Alqarni, "Optimization of two-step block method with three hybrid points for solving third order initial value problems," *J. Nonlinear Sci. Appl.*, vol. 12, no. 7, pp. 450–469, Mar. 2019.
[10] M. K. Duromola and A. L. Momoh, "Hybrid numerical method with block extension for direct solution of third order ordinary differential equations," *Amer. J. Comput. Math.*, vol. 9, no. 2, pp. 68–80, 2019.
[11] M. Alkasassbeh and Z. Omar, "Hybrid one-step block fourth derivative method for the direct solution of third order initial value problems of ordinary differential equations," *Int. J. Pure Appl. Math.*, vol. 119, no. 1, pp. 207–224, 2018.
[12] D. N. Butusov, A. V. Tutueva, and E. S. Homitskaya, "Extrapolation semi-implicit ODE solvers with adaptive timestep," in *Proc. 19th IEEE Int. Conf. Soft Comput. Meas. (SCM)*, Saint-Petersburg, Russia, May 2016, pp. 137–140.
[13] R. I. McLachlan, G. Reinout, and W. Quispel, "Geometric integrators for ODEs," *J. Phys. A, Math. Gen.*, vol. 39, no. 19, p. 5251, 2006.
[14] F. Casas and A. Escorihuela-Tomàs, "Composition methods for dynamical systems separable into three parts," *Mathematics*, vol. 8, no. 4, p. 533, Apr. 2020.
[15] M. Saqlain, W. Irshad, U. Bashir, and G. Murtaza, "Numerical solutions for third order ordinary differential equation by differential transform method & Elzaki transform method," *Sci. Inquiry Rev.*, vol. 2, no. 4, pp. 1–11, Oct. 2018.
[16] E. Momoniat, T. A. Selway, and K. Jina, "Analysis of Adomian decomposition applied to a third-order ordinary differential equation from thin film flow," *Nonlinear Anal., Theory, Methods Appl.*, vol. 66, no. 10, pp. 2315–2324, May 2007.
[17] S. Chakraverty and S. Mall, "Single layer Chebyshev neural network model with regression-based weights for solving nonlinear ordinary differential equations," *Evol. Intell.*, vol. 13, pp. 1–8, Mar. 2020.
[18] Y. Yang, M. Hou, H. Sun, T. Zhang, F. Weng, and J. Luo, "Neural network algorithm based on legendre improved extreme learning machine for solving elliptic partial differential equations," *Soft Comput.*, vol. 24, no. 2, pp. 1083–1096, Jan. 2020.
[19] K. Moaddy, M. Al-Smadi, and I. Hashim, "A novel representation of the exact solution for differential algebraic equations system using residual power-series method," *Discrete Dyn. Nature Soc.*, vol. 2015, pp. 1–12, Jan. 2015.
[20] M. T. Haweel, F. E. A. El-Samie, and O. Zahran, "Polynomial series FLANN for nonlinear equalization," *Menoufia J. Electron. Eng. Res.*, vol. 28, no. 1, pp. 78–82, Dec. 2019.
[21] H. H. Ali and M. T. Haweel, "Legendre neural networks with multi input multi output system equations," in *Proc. 7th Int. Conf. Comput. Eng. Syst. (ICCES)*, Cairo, Egypt, Nov. 2012, pp. 92–97.
[22] H. H. Ali and M. T. Haweel, "Legendre based equalization for nonlinear wireless communication channels," in *Proc. Saudi Int. Electron., Commun. Photon. Conf. (SIECPC)*, Riyadh, Saudi Arabia, Apr. 2013, pp. 1–4.
[23] Y. Lu, Q. Yin, H. Li, H. Sun, Y. Yang, and M. Hou, "The LS-SVM algorithms for boundary value problems of high-order ordinary differential equations," *Adv. Difference Equ.*, vol. 2019, no. 1, pp. 1–22, Dec. 2019.
[24] K. A. Hussain, F. Ismail, N. Senu, and F. Rabiei, "Fourth-order improved Runge–Kutta method for directly solving special third-order ordinary differential equations," *Iranian J. Sci. Technol., Trans. A, Sci.*, vol. 41, no. 2, pp. 429–437, 2017.
[25] K. Sj and O. Fo, "Symmetric 2-step 4-point hybrid method for the solution of general third order differential equations," *J. Appl. Comput. Math.*, vol. 06, no. 02, p. 348, 2017.
[26] O. O. Agboola, A. A. Opanuga, and J. A. Gbadeyan, "Solution of third order ordinary differential equations using differential transform method," *Global J. Pure Appl. Math.*, vol. 11, no. 4, pp. 2511–2516, 2015.
[27] R. B. Ogunrinde, "Comparative study of differential transformation method (DTM) and adomian decomposition method (ADM) for solving ordinary differential equations," *J. Contemp. Appl. Math.*, vol. 9, no. 1, pp. 1–25, Jul. 2019.
[28] K. Batiha and B. Batiha, "A new algorithm for solving linear ordinary differential equations," *World Appl. Sci. J.*, vol. 15, no. 12, pp. 1774–1779, 2011.
[29] N. Ghawadri, N. Senu, F. Ismail, and Z. B. Ibrahim, "Exponentially fitted and trigonometrically fitted explicit modified Runge–Kutta type methods for solving y''' = f(x,y,y')," *J. Appl. Math.*, vol. 2018, pp. 1–19, Jul. 2018.
[30] E. Momoniat and F. M. Mahomed, "Symmetry reduction and numerical solution of a third-order ODE from thin film flow," *Math. Comput. Appl.*, vol. 15, no. 4, pp. 709–719, Dec. 2010.

MOHAMMAD T. HAWEEL was born in Chicago, Illinois, USA, in 1988. He received the B.S. degree in electronics and communications engineering from the Shorouk Academy, Cairo, Egypt, in 2010, and the M.S. degree in electrical and electronic engineering from the Arab Academy for Science Technology and Maritime Transport, Cairo, in 2014. He is currently pursuing the Ph.D. degree with the Department of Electronics and Electrical Communication Engineering, Menoufia University, Cairo. From 2010 to 2014, he was a Teaching Assistant with Egyptian Russian University (ERU), Cairo, and after receiving the M.S. degree, he became a full-time Lecturer with ERU, from 2014 to 2017. He is also working as a full-time Lecturer with Shaqra University, Riyadh, Saudi Arabia. He published two journals and nine IEEE conference papers in the field of signal processing, neural networks and its applications, and two journals. His research interests include signal processing and their applications in communications, mathematics, biomedical engineering, system modeling, adaptive filter design, neural networks and learning algorithms, image processing, and engineering education and quality control.

O. ZAHRAN received the B.Sc. degree (Hons.) from the Faculty of Electronic Engineering, Menoufia University, Egypt, in 1997, and the Ph.D. degree from the University of Liverpool, U.K., in 2006. In 2007, he joined as a Teac hing Staff with the Department of Electronics and Electrical Communications Engineering, Faculty of Electronic Engineering, Menoufia University. His current research interests include artificial intelligent, image enhancement, image restoration, image interpolation, super-resolution reconstruction of images, data hiding, multimedia communications, medical image processing, optical signal processing, digital communications, and non-destructive testing applications.

FATHI E. ABD EL-SAMIE (Member, IEEE) received the B.Sc. (Hons.), M.Sc., and Ph.D. degrees from Menoufia University, Menouf, Egypt, in 1998, 2001, and 2005, respectively. Since 2005, he has been a Teaching Staff Member with the Department of Electronics and Electrical Communications, Faculty of Electronic Engineering, Menoufia University. His current research interests include image enhancement, image restoration, image interpolation, super-resolution reconstruction of images, data hiding, multimedia communications, medical image processing, optical signal processing, and digital communications. He has received the most cited paper award from the *Digital Signal Processing* journal in 27 August 2008.