

## Research Article

# A Novel Representation of the Exact Solution for Differential Algebraic Equations System Using Residual Power-Series Method

Khaled Moaddy,<sup>1</sup> Mohammed AL-Smadi,<sup>2</sup> and Ishak Hashim<sup>3,4</sup>

<sup>1</sup>Department of Mathematics, Faculty of Science and Arts, Shaqra University, Shaqra 11691, Saudi Arabia

<sup>2</sup>Applied Science Department, Ajloun College, Al-Balqa Applied University, Ajloun 26816, Jordan

<sup>3</sup>School of Mathematical Sciences, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

<sup>4</sup>Research Institute, Center for Modeling & Computer Simulation (RI/CM&CS), King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia

Correspondence should be addressed to Khaled Moaddy; moaddy@yahoo.com

Received 16 October 2014; Accepted 7 December 2014

Academic Editor: Zhengqiu Zhang

Copyright © 2015 Khaled Moaddy et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We implement a relatively new analytic iterative technique to get approximate solutions of differential algebraic equations system based on generalized Taylor series formula. The solution methodology is based on generating the residual power series expansion solution in the form of a rapidly convergent series with easily computable components. The residual power series method (RPSM) can be used as an alternative scheme to obtain analytical approximate solution of different types of differential algebraic equations system applied in mathematics. Simulations and test problems were analyzed to demonstrate the procedure and confirm the performance of the proposed method, as well as to show its potentiality, generality, viability, and simplicity. The results reveal that the proposed method is very effective, straightforward, and convenient for solving different forms of such systems.

## 1. Introduction

Differential algebraic equations (DAEs) have gained in the recent years considerable importance and popularity partly due to its powerful potential applications. The applications of the DAEs can be found in many fields of scientific and engineering including electrical networks, circuit analysis, control theory, robotics, power systems, and chemical process simulations [1–5], which can be successfully modeled by coupled differential and algebraic equation. In this regard, the DAEs are type of differential equations, in which the unknown functions are satisfying additional algebraic equations, whereas the derivatives are not in general expressed explicitly and typically derivatives of some of the dependent variables may not appear in the equations at all. In other words, the DAEs consist of a set of differential equations together with additional algebraic constraints.

Many mathematicians, engineers, and physicists have devoted considerable effort to studying the nature of the DAEs involving explicit and approximate solutions where a

complicated system of DAEs can be described quite precisely with mathematical expressions. Generally, many explicit solutions have been found to the linear DAEs, but there exists no method that yields an explicit solution for the nonlinear DAEs. Most often, these systems are typically modeled as complex nonlinear DAEs. Thus, a purely mathematical analysis is often not possible to obtain a closed-form solution. Approximations based on numerical methods are therefore needed.

In science literature, a lot of studies about the DAEs have appeared according to [6–9] and references therein. In this point, it is usually difficult to solve these kinds of DAEs analytically. Therefore, there are many powerful numerical methods in literature that can be used to approximate solutions to the DAEs system. To mention but a few, the backward differentiation formula (BDF) has been the first numerical method that was used to obtain the solutions for a class of DAEs [10]. In [11], the implicit Runge-Kutta method has been applied to solve the DAEs numerically. Çelik et al., in [12], have presented approximate solutions for chemical DAEs

using the Padé series method. Also, Çelik and Yeloglu [13] have used the Chebyshev series approximation for solving other kinds of DAEs. In contrast, Hosseini has employed the Adomian decomposition method (ADM) for solving linear and nonlinear forms of the DAEs [14, 15]. In addition, the variational iteration method (VIM) [16] and the homotopy perturbation method (HPM) [17] have been introduced for dealing with different kinds of DAEs and their engineering applications. Lately, the authors [18] have developed a new modified method to further investigate for solving linear and nonlinear system of DAEs of higher index without index reduction based on the HPM. However, none of previous studies propose a methodical way to solve systems of DAEs. Moreover, previous studies require more effort to achieve the results and usually they are suited for linear form of DAEs system. The applications of other versions for linear and nonlinear problems can be found in [19–26] and references therein.

Furthermore, the RPSM has been developed as an efficient numerical as well as analytical method to determine the coefficients of power series solutions for a class of fuzzy differential equation by Abu Arqub [27]. Besides, the RPSM has been successfully applied to get numerical solutions for many other problems, for instance, generalized Lane-Emden equation which is a highly nonlinear singular differential equation [28], regular initial value problems [29], composite and noncomposite fractional differential equations [30], and boundary value problems of fractional order [31]. This method is effective to construct power series expansion solution for strongly linear and nonlinear equations without linearization, perturbation, or discretization [32], which computes the coefficients of the power series solutions by chain of linear equations of one variable. The RPSM is an alternative procedure for obtaining analytical Taylor series solution for system of DAEs. Consequently, using the residual error concept, we get a series solution in practice as well as a truncated series solution. For linear problems, the exact solution can be obtained by few terms of the RPS solution.

The RPSM is an approximate analytical method for solving large amount of problems of ordinary, partial, integral, and integrodifferential equations. The biggest advantage over the existing standard analytical techniques is that it overcomes the difficulty arising in calculating complicated terms. However, the method provides the solution in terms of a rapidly convergent series with easily computable components. The analytical approximate solution should be constructed in the form of a polynomial which does not exhibit the real behaviors of the problem but gives a good approximation to the true solution in the given interval. Further, this approach is different from the traditional higher order Taylor series method. The Taylor series method is computationally expensive for large orders and is suited for the linear problems, whilst the RPS techniques need small computational requirements together with high precision.

In this paper, we consider the following system of differential algebraic equations:

$$f_i(t, x_1(t), x_2(t), \dots, x_n(t), x'_1(t), x'_2(t), \dots, x'_n(t)) = 0, \\ i = 1, 2, \dots, m,$$

$$f_i(t, x_1(t), x_2(t), \dots, x_n(t)) = 0, \quad i = m + 1, 2, \dots, n, \quad (1)$$

subject to the initial conditions

$$x_1(t_0) = \alpha_1, \quad x_2(t_0) = \alpha_2, \dots, x_n(t_0) = \alpha_n, \quad (2)$$

where  $t_0 \leq t \leq T$ ,  $t_0$ ,  $\alpha_i$ , and  $a$  are real finite constants with  $a > 0$ ,  $f_i : [t_0, T] \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  are nonlinear continuous functions of  $t$ ,  $x_i$ ,  $x'_i$ ,  $i = 1, 2, \dots, n$ , which satisfy all necessary requirements of the existence of a unique solution, and  $x_i(t)$  are unknown functions of independent variable  $t$  to be determined. Throughout this paper, we assume that  $f_i$ ,  $x_i$  are analytical functions on the given interval.

For several reasons, we consider (1) directly rather than to try to reformulate it as an initial value problem of the form

$$x'_i(t) = f_i(t, x_1(t), x_2(t), \dots, x_n(t)), \quad i = 1, 2, \dots, n. \quad (3)$$

Some often, via simulation of physical phenomena as a matter of fact, the model takes form of system of DAEs depicting a collection of relationships between the variables with interest and some of their derivatives. These relationships may even be generated automatically by modeling or simulation program. The variables usually have a physical significance; changing the model to (2) may produce less meaningful variables. In the case of computer-generated or nonlinear models, especially complex (nonlinear) models, it may be time consuming or impossible to obtain an explicit model. Parameters are present in many applications where any change in the parameter values can alter the relationships between variables, which require different explicit models with solution manifolds of different dimensions. On the other hand, the index of DAEs systems is a measure of the degree of singularity of the system and also widely regarded as an indication of certain difficulties for numerical methods. However, if the original DAEs can be solved directly, then it becomes easier to explore the effect of modeling changes and parameter variation and also to interface the modeling software directly with design software. These advantages enable researchers to focus their attention on the physical phenomena of interest. The desirability of working directly with DAEs has been recognized for over twenty years by scientists and engineers in several areas. Depending on the area, DAEs have been called singular, implicit, differential-algebraic, descriptor, generalized state space, noncanonic, noncausal, degenerate, semistate, constrained, reduced order model, and nonstandard systems.

The basic motivation of this paper is to apply the RPSM to develop an approach for obtaining the representation of exact and approximate solutions to strongly linear and nonlinear system of DAEs. This approach is simple and needs less effort to achieve the results. It does not require any converting while switching from first to higher order in which the solutions and all its derivatives are applicable for each arbitrary point in the given interval. Thus, the RPSM can be applied directly to the given problems by choosing an appropriate value for the initial guess approximation.

The remainder of this paper is organized as follows. In Section 2, basic idea of the residual power series method (RPSM) together with analysis of the method is presented. In the meantime, the RPSM is extended to provide symbolic approximate series solutions for system of differential algebraic equations (1) and (2). Based on the above, numerical examples are given to illustrate the capability of the proposed method in Section 3. Results reveal that only few terms are required to deduce the approximate solutions which are found to be accurate and efficient. Finally, some conclusions are summarized in the last section.

## 2. Adaptation of Residual Power Series Method (RPSM)

In this section, we present a brief description and preliminary results of the standard RPSM that will be used in the remainder of this paper. In the meantime, we employ our technique for finding out a series solution of system (1) and (2) by formulating and analyzing the method in solving such systems. Afterward, a convergence theorem is presented in order to capture the behavior of the solution.

The RPSM consists of expressing the solutions of system (1) and (2) as a power series expansion about the initial point  $t = t_0$ . To achieve our goal, we suppose that these solutions take the following form:

$$x_i(t) = \sum_{m=0}^{\infty} x_{i,m}(t), \quad i = 1, 2, \dots, n, \quad (4)$$

where  $x_{i,m}(t)$  are the terms of approximations such that  $x_{i,m}(t) = c_{i,m}(t - t_0)^m$ .

For  $m = 0$ , substituting the initial guesses  $x_{i,0}(t_0) = c_{i,0} = x_i^{(m)}(t_0)/m! = x_i(t_0)$ , which are known from initial conditions (2), into  $x_i(t)$ ,  $i = 1, 2, \dots, n$ , leads to the approximate solutions for the DAEs system such that  $x_i(t) = x_{i,0}(t_0) + \sum_{m=1}^{\infty} x_{i,m}(t)$ ,  $i = 1, 2, \dots, n$ , whereas  $x_{i,m}(t)$ , for  $m = 1, 2, \dots, k$ , can be calculated by the following  $k$ th-truncated series:

$$x_i^k(t) = \sum_{m=0}^k c_{i,m}(t - t_0)^m, \quad i = 1, 2, \dots, n. \quad (5)$$

Regarding applying the RPSM, the  $k$ th-residual functions and the  $\infty$ th residual functions are given, respectively, by

$$\begin{aligned} \text{Res}_i^k(t) &= f_i\left(t, x_1^k(t), x_2^k(t), \dots, x_n^k(t), \right. \\ &\quad \left. \frac{d}{dt}x_1^k(t), \frac{d}{dt}x_2^k(t), \dots, \frac{d}{dt}x_n^k(t)\right), \\ &\quad i = 1, 2, \dots, m, \end{aligned}$$

$$\begin{aligned} \text{Res}_i^k(t) &= \frac{d}{dt}f_i\left(t, x_1^k(t), x_2^k(t), \dots, x_n^k(t)\right) = 0, \\ &\quad i = m + 1, \dots, n, \end{aligned} \quad (6)$$

$$\begin{aligned} \text{Res}_i^{\infty}(t) &= \lim_{k \rightarrow \infty} \text{Res}_i^k(t) \\ &= f_i\left(t, x_1(t), x_2(t), \dots, x_n(t), \right. \\ &\quad \left. x_1'(t), x_2'(t), \dots, x_n'(t)\right), \quad i = 1, 2, \dots, m, \end{aligned}$$

$$\begin{aligned} \text{Res}_i^{\infty}(t) &= \lim_{k \rightarrow \infty} \text{Res}_i^k(t) \\ &= \frac{d}{dt}f_i\left(t, x_1(t), x_2(t), \dots, x_n(t)\right) = 0, \\ &\quad i = m + 1, \dots, n. \end{aligned} \quad (7)$$

Obviously,  $\text{Res}_i^{\infty}(t) = 0$  for each  $t \in [t_0, T]$ , which are infinitely differentiable functions at  $t = t_0$ . Furthermore,  $(d^m/dt^m)\text{Res}_i^{\infty}(t_0) = (d^m/dt^m)\text{Res}_i^k(t_0) = 0$ ,  $m = 0, 1, 2, \dots, k$ , and these relations are fundamental rules in the RPSM and its applications. Particularly,  $(d^{k-1}/dt^{k-1})\text{Res}_i^{\infty}(t_0) = (d^{k-1}/dt^{k-1})\text{Res}_i^k(t_0) = 0$ ,  $i = 1, 2, \dots, n$ ,  $k = 1, 2, \dots$

Now, substituting the  $k$ th-truncated series  $x_i^k(t)$  into (6) yields

$$\begin{aligned} \text{Res}_i^k(t) &= f_i\left(t, \sum_{m=0}^k c_{1,m}(t - t_0)^m, \sum_{m=0}^k c_{2,m}(t - t_0)^m, \dots, \right. \\ &\quad \sum_{m=0}^k c_{n,m}(t - t_0)^m, \sum_{m=1}^k m c_{1,m}(t - t_0)^{m-1}, \\ &\quad \sum_{m=1}^k m c_{2,m}(t - t_0)^{m-1}, \dots, \\ &\quad \left. \sum_{m=1}^k m c_{n,m}(t - t_0)^{m-1}\right), \quad i = 1, 2, \dots, m, \\ \text{Res}_i^k(t) &= \frac{d}{dt}f_i\left(t, \sum_{m=0}^k c_{1,m}(t - t_0)^m, \sum_{m=0}^k c_{2,m}(t - t_0)^m, \dots, \right. \\ &\quad \left. \sum_{m=0}^k c_{n,m}(t - t_0)^m\right) = 0, \quad i = m + 1, \dots, n. \end{aligned} \quad (8)$$

Consequently, based on  $\text{Res}_i^{\infty}(t_0) = \text{Res}_i^1(t_0) = 0$ ,  $i = 1, 2, \dots, n$ , setting  $t = t_0$  and  $k = 1$  in (8) leads to

$$\begin{aligned} f_i(t_0, \alpha_1, \alpha_2, \dots, \alpha_n, c_{1,1}, c_{2,1}, \dots, c_{n,1}) &= 0, \\ &\quad i = 1, 2, \dots, m, \end{aligned}$$

$$\begin{aligned} & \frac{\partial}{\partial t} f_i(t_0, \alpha_1, \alpha_2, \dots, \alpha_n) \\ & + \sum_{j=1}^n c_{j,1} \frac{\partial}{\partial x_j} f_i(t_0, \alpha_1, \alpha_2, \dots, \alpha_n) = 0, \\ & i = m + 1, 2, \dots, n, \end{aligned} \tag{9}$$

which contained  $n$  equations in  $n$  variables that can be solved directly by using the following Mathcad commands:

$$\begin{pmatrix} f_1(t_0, \alpha_1, \alpha_1, \dots, \alpha_1, c_{1,1}, c_{1,2}, \dots, c_{n,1}) \\ f_2(t_0, \alpha_1, \alpha_1, \dots, \alpha_1, c_{1,1}, c_{1,2}, \dots, c_{n,1}) \\ \vdots \\ f_m(t_0, \alpha_1, \alpha_1, \dots, \alpha_1, c_{1,1}, c_{1,2}, \dots, c_{n,1}) \\ \frac{\partial}{\partial t} f_m(t_0, \alpha_1, \alpha_2, \dots, \alpha_n) + \sum_{j=1}^n c_{j,1} \frac{\partial}{\partial x_j} f_{m+1}(t_0, \alpha_1, \alpha_2, \dots, \alpha_n) \\ \frac{\partial}{\partial t} f_{m+1}(t_0, \alpha_1, \alpha_2, \dots, \alpha_n) + \sum_{j=1}^n c_{j,1} \frac{\partial}{\partial x_j} f_{m+2}(t_0, \alpha_1, \alpha_2, \dots, \alpha_n) \\ \vdots \\ \frac{\partial}{\partial t} f_n(t_0, \alpha_1, \alpha_2, \dots, \alpha_n) + \sum_{j=1}^n c_{j,1} \frac{\partial}{\partial x_j} f_n(t_0, \alpha_1, \alpha_2, \dots, \alpha_n) \end{pmatrix} \text{solve,}$$

$$\begin{pmatrix} c_{1,1} \\ c_{2,1} \\ \vdots \\ c_{n,1} \end{pmatrix} \rightarrow (\beta_1^{(1)} \ \beta_2^{(1)} \ \dots \ \beta_n^{(1)}).$$

(10)

Thus, by using (5), the first approximate solutions of system (1) and (2) can be written as follows:

$$x_i^1(t) = \alpha_i + \beta_i^{(1)}(t - t_0), \quad i = 1, 2, \dots, n. \tag{11}$$

Next, in order to obtain the second approximate solutions, we set  $k = 2$  such that  $x_i^2(t) = \sum_{m=0}^2 c_{i,m}(t - t_0)^m$ . Then, we differentiate both sides of (8) with respect to  $t$  and substitute  $t = t_0$  to obtain

$$\begin{aligned} & \left( \frac{d}{dt} \text{Res}_i^2 \right) (t_0) \\ & = \frac{\partial}{\partial t} f_i(t_0, \alpha_1, \alpha_2, \dots, \alpha_n, \beta_1^{(1)}, \beta_2^{(1)}, \dots, \beta_n^{(1)}) \\ & + \sum_{j=1}^n \beta_j^{(1)} \frac{\partial}{\partial x_j} f_i(t_0, \alpha_1, \alpha_2, \dots, \alpha_n, \beta_1^{(1)}, \beta_2^{(1)}, \dots, \beta_n^{(1)}) \\ & + 2 \sum_{j=1}^n c_{j,2} \frac{\partial}{\partial x_j} f_i(t_0, \alpha_1, \alpha_2, \dots, \alpha_n, \beta_1^{(1)}, \beta_2^{(1)}, \dots, \beta_n^{(1)}) \\ & := F_i(c_{1,2}, c_{2,2}, \dots, c_{n,2}), \quad i = 1, 2, \dots, m, \end{aligned}$$

$$\begin{aligned} & \left( \frac{d}{dt} \text{Res}_i^2 \right) (t_0) \\ & = \frac{\partial^2}{\partial t^2} f_i(t_0, \alpha_1, \alpha_2, \dots, \alpha_n) \\ & + 2 \sum_{j=1}^n c_{j,2} \frac{\partial}{\partial x_j} f_i(t_0, \alpha_1, \alpha_2, \dots, \alpha_n) \\ & + \sum_{j=1}^n \left[ \frac{\partial}{\partial t} \frac{\partial}{\partial x_j} f_i(t_0, \alpha_1, \alpha_2, \dots, \alpha_n) \right. \\ & \quad + \beta_1^{(1)} \frac{\partial^2}{\partial x_j \partial x_1} f_i(t_0, \alpha_1, \alpha_2, \dots, \alpha_n) \\ & \quad + \beta_2^{(1)} \frac{\partial^2}{\partial x_j \partial x_2} f_i(t_0, \alpha_1, \alpha_2, \dots, \alpha_n) \\ & \quad \left. + \dots + \beta_n^{(1)} \frac{\partial^2}{\partial x_j \partial x_n} f_i(t_0, \alpha_1, \alpha_2, \dots, \alpha_n) \right] \beta_j^{(1)} \\ & := G_i(c_{1,2}, c_{2,2}, \dots, c_{n,2}), \quad i = m + 1, 2, \dots, n. \end{aligned} \tag{12}$$

According to the fact that  $(d/dt)\text{Res}_i^2(t_0) = (d/dt)\text{Res}_i^\infty(t_0) = 0, i = 1, 2, \dots, n$ , the values of  $c_{i,2}$  can be found easily by using the following Mathcad commands:

$$\begin{pmatrix} F_1(c_{1,2}, c_{2,2}, \dots, c_{n,2}) \\ F_2(c_{1,2}, c_{2,2}, \dots, c_{n,2}) \\ \vdots \\ F_m(c_{1,2}, c_{2,2}, \dots, c_{n,2}) \\ G_{m+1}(c_{1,2}, c_{2,2}, \dots, c_{n,2}) \\ G_{m+2}(c_{1,2}, c_{2,2}, \dots, c_{n,2}) \\ \vdots \\ G_n(c_{1,2}, c_{2,2}, \dots, c_{n,2}) \end{pmatrix} \text{solve,}$$

$$\begin{pmatrix} c_{1,2} \\ c_{2,2} \\ \vdots \\ c_{n,2} \end{pmatrix} \rightarrow (\beta_1^{(2)} \ \beta_2^{(2)} \ \dots \ \beta_n^{(2)}).$$

(13)

Hence, the second approximate solutions of system (1) and (2) can be written as follows:

$$x_i^2(t) = \alpha_i + \beta_1^{(1)}(t - t_0) + \beta_1^{(2)}(t - t_0)^2, \tag{14}$$

$$i = 1, 2, \dots, n.$$

By the same technique, the process can be repeated to generate a sequence of approximate solutions  $x_i^k(t)$  for the system (1) and (2). Moreover, higher accuracy can be achieved by evaluating more components of the solution.

It will be convenient to have a notation for the error in the approximation  $x_i(t) \approx x_i^k(t)$ . Accordingly, we will let

$\text{Rem}_i^k(t)$  denote the difference between  $x_i(t)$  and its  $k$ th Taylor polynomial; that is,

$$\begin{aligned} \text{Rem}_i^k(t) &= x_i(t) - x_i^k(t) \\ &= \sum_{m=k+1}^{\infty} x_i^{(m)}(t_0)(t-t_0)^m, \quad i = 1, 2, \dots, n, \end{aligned} \tag{15}$$

where the functions  $\text{Rem}_i^k(t)$  are called the  $k$ th remainder for the Taylor series of  $x_i(t)$ . In fact, it often happens that the remainder  $\text{Rem}_i^k(t)$  become smaller and smaller, approaching zero, as  $k$  gets large.

Next, we present a convergence theorem of the RPSM to capture the behavior of these solutions. Afterwards, we introduce the error functions to study the accuracy and efficiency of the method. Actually, continuous approximations to the solution will be obtained. Taylor's theorem allows us to represent fairly general functions exactly in terms of polynomials with a known, specified, and bounded error. The next theorem will guarantee convergence to the exact analytic solution of (1) and (2).

**Theorem 1.** *Suppose that  $x_i(t)$ ,  $i = 1, 2, \dots, n$ , is the exact solution for system (1) and (2). Then, the approximate solution obtained by the RPSM is in fact the Taylor expansion of  $x_i(t)$ .*

*Proof.* The proof of Theorem 1 is similar to proof of Theorem 1 in [28].  $\square$

**Corollary 2.** *Let  $x_i(t)$ ,  $i = 1, 2, \dots, n$ , be a polynomial for some  $i$ ; then the RPSM will obtain the exact solution.*

To show the accuracy of the present method for our problems, we report four types of error. The first one is the residual error,  $\text{Res}_i^k(t)$ , and is defined as

$$\text{Res}_i^k(t) := \left| \frac{d}{dt} x_i^k(t) - f_i(t, x_1^k(t), x_2^k(t), \dots, x_n^k(t)) \right|, \tag{16}$$

while the exact, Ext, relative, Rel, and consecutive, Con, errors are defined, respectively, by

$$\begin{aligned} \text{Ext}_i^k(t) &:= |x_{i,\text{exact}}(t) - x_i^k(t)|, \\ \text{Rel}_i^k(t) &:= \frac{|x_{i,\text{exact}}(t) - x_i^k(t)|}{|x_{i,\text{exact}}(t)|}, \\ \text{Con}_i^k(t) &:= |x_i^{k+1}(t) - x_i^k(t)|, \end{aligned} \tag{17}$$

for  $i = 1, 2, \dots, n$ , where  $t \in [t_0, T]$ ,  $x_i^k$  are the  $k$ th-order approximation of  $x_{i,\text{exact}}(t)$  obtained by the RPSM and  $x_{i,\text{exact}}(t)$  are the exact solution. An excellent account of the study of error analysis, which includes its definitions, varieties, applications, and method of derivations, can be found in [34].

### 3. Simulations and Test Problems

To give a clear overview of the content of this work, we consider some examples to demonstrate the performance and

efficiency of the present technique. The proposed method provides an analytical approximate solution in terms of an infinite power series. However, there is a practical need to evaluate this solution and to obtain numeric values from the infinite power series. The consequent series truncation and the practical procedure are conducted to accomplish this task and transform the otherwise analytical results into an exact solution, which is evaluated to a finite degree of accuracy. In most real life situations, the differential equation that models the problem is too complicated to solve exactly, and there is a practical need to approximate the solution.

*Example 1.* Consider the following linear system of DAEs [33]:

$$\begin{aligned} x_1'(t) - tx_2'(t) + t^2x_3'(t) + x_1(t) \\ - (t+1)x_2(t) + (t^2+2t)x_3(t) &= 0, \\ x_2'(t) - tx_3'(t) - x_2(t) + (t-1)x_3(t) &= 0, \\ x_3(t) &= \sin(t), \end{aligned} \tag{18}$$

subject to the initial conditions

$$x_1(0) = 1, \quad x_2(0) = 1, \quad x_3(0) = 0. \tag{19}$$

The exact solution is  $x_1(t) = e^{-t} + te^t$ ,  $x_2(t) = e^t + t \sin(t)$ ,  $x_3(t) = \sin(t)$ .

To apply the RPS approach in solving system (18) and (19), we start with selecting initial guesses of the approximations such that  $x_{1,0}(t) = 1$ ,  $x_{2,0}(t) = 1$ , and  $x_{3,0}(t) = 0$ ; then the  $k$ th-truncated series solutions  $x_i^k(t)$ ,  $i = 1, 2, 3$ , for system (18) and (19) are given by

$$\begin{aligned} x_1^k(t) &= \sum_{m=0}^k c_{1,m}t^m = 1 + c_{1,1}t + c_{1,2}t^2 + \dots + c_{1,k}t^k, \\ x_2^k(t) &= \sum_{m=0}^k c_{2,m}t^m = 1 + c_{2,1}t + c_{2,2}t^2 + \dots + c_{2,k}t^k, \\ x_3^k(t) &= \sum_{m=1}^k c_{3,m}t^m = c_{3,1}t + c_{3,2}t^2 + c_{3,3}t^3 + \dots + c_{3,k}t^k. \end{aligned} \tag{20}$$

Accordingly, the unknown coefficients  $c_{i,m}$ ,  $m = 1, 2, \dots, k$ ,  $i = 1, 2, 3$ , can be found by constructing the following  $k$ th residual functions  $\text{Res}_i^k(t)$ ,  $i = 1, 2, 3$ , such that

$$\begin{aligned} \text{Res}_1^k(t) &= \sum_{m=1}^k mc_{1,m}t^{m-1} - \sum_{m=1}^k mc_{2,m}t^m \\ &+ \sum_{m=1}^k mc_{3,m}t^{m+1} + \sum_{m=0}^k c_{1,m}t^m - (t+1) \\ &\cdot \sum_{m=0}^k c_{2,m}t^m + (t^2+2t) \sum_{m=1}^k c_{3,m}t^m, \end{aligned}$$



$$\begin{aligned} \text{Res}_2^k(t) &= \sum_{m=1}^k mc_{2,m}t^{m-1} - \sum_{m=1}^k mc_{3,m}t^m \\ &\quad - \sum_{m=0}^k c_{2,m}t^m + (t-1) \sum_{m=1}^k c_{3,m}t^m, \\ \text{Res}_3^k(t) &= \frac{d}{dt} \left[ \sum_{m=1}^k c_{3,m}t^m - \sin t \right]. \end{aligned} \quad (21)$$

Now, in order to find the 1st-approximate solutions, put  $k = 1$  through 1 to get that

$$\begin{aligned} \text{Res}_1^1(t) &= c_{1,1} - (1 - c_{1,1} + 2c_{2,1})t \\ &\quad - (c_{2,1} - 3c_{3,1})t^2 + c_{3,1}t^3, \\ \text{Res}_2^1(t) &= (1-t)c_{2,1} - 2c_{3,1}t + c_{3,1}t^2 - 1, \\ \text{Res}_3^1(t) &= \frac{d}{dt} [c_{3,1}t - \sin t]. \end{aligned} \quad (22)$$

Using the facts that  $\text{Res}_1^1(0) = \text{Res}_2^1(0) = \text{Res}_3^1(0) = 0$ , the values of  $c_{i,1}$ ,  $i = 1, 2, 3$ , are given by  $c_{1,1} = 0$ ,  $c_{2,1} = 1$ , and  $c_{3,1} = 1$ . Based on this, the first approximations  $x_i^1(t)$ ,  $i = 1, 2, 3$ , of the RPS solution for (18) and (19) can be written by  $x_1^1(t) = 1$ ,  $x_2^1(t) = 1 + t$ , and  $x_3^1(t) = t$ . In contrast, the second approximations  $x_i^2(t)$ ,  $i = 1, 2, 3$ , of the RPS solutions have the forms  $x_1^2(t) = 1 + c_{1,2}t^2$ ,  $x_2^2(t) = 1 + t + c_{2,2}t^2$ , and  $x_3^2(t) = t + c_{3,2}t^2$ , in which the coefficients  $c_{1,2}$ ,  $c_{2,2}$ , and  $c_{3,2}$  can be found by putting  $k = 2$  through 1 and differentiating both sides of 1 with respect to  $t$  such that

$$\begin{aligned} \left( \frac{d}{dt} \text{Res}_1^2 \right) (t) &= 2c_{1,2} + 2(2 + c_{1,2} - 3c_{2,2})t \\ &\quad + 3(1 - c_{2,2} + 4c_{3,2})t^2 + 4c_{3,2}t^3 - 3, \\ \left( \frac{d}{dt} \text{Res}_2^2 \right) (t) &= 2c_{2,2} + 2(1 - c_{2,2} - 3c_{3,2})t + 3c_{3,2}t^2 - 3, \\ \left( \frac{d}{dt} \text{Res}_3^2 \right) (t) &= \frac{d}{dt} [1 + 2c_{3,2}t - \cos t]. \end{aligned} \quad (23)$$

Consequently,  $c_{1,2} = 3/2$ ,  $c_{2,2} = 3/2$ , and  $c_{3,1} = 0$  as soon as  $(d/dt)\text{Res}_1^2(0) = (d/dt)\text{Res}_2^2(0) = (d/dt)\text{Res}_3^2(0) = 0$ . Therefore, the second approximations for (18) and (19) are  $x_1^2(t) = 1 + (3/2)t^2$ ,  $x_2^2(t) = 1 + t + (3/2)t^2$ , and  $x_3^2(t) = t$ . On the other hand, the fourth approximations of the RPS solution for (18) and (19) according to  $(d^3/dt^3)\text{Res}_i^4(0) =$

$(d^3/dt^3)\text{Res}_i^4(0) = (d^3/dt^3)\text{Res}_i^4(0) = 0$  of the  $k$ th residual function 1 are given by

$$\begin{aligned} x_1^4(t) &= 1 + \frac{3}{2}t^2 + \frac{1}{3}t^3 + \frac{5}{24}t^4, \\ x_2^4(t) &= 1 + t + \frac{3}{2}t^2 + \frac{1}{6}t^3 - \frac{1}{8}t^4, \\ x_3^4(t) &= t + \frac{1}{6}t^3. \end{aligned} \quad (24)$$

Likewise, the 10th truncated series  $x_i^{10}(t)$ ,  $i = 1, 2, 3$ , of the RPS solution are given by

$$\begin{aligned} x_1^{10}(t) &= 1 + \frac{3t^2}{2} + \frac{t^3}{3} + \frac{5t^4}{24} + \frac{t^5}{30} + \frac{7t^6}{720} \\ &\quad + \frac{t^7}{840} + \frac{t^8}{4480} + \frac{t^9}{45360} + \frac{11t^{10}}{3628800} \\ &= \left( 1 - t + \frac{t^2}{2!} - \frac{t^3}{3!} + \frac{t^4}{4!} - \frac{t^5}{5!} \right. \\ &\quad \left. + \frac{t^6}{6!} - \frac{t^7}{7!} + \frac{t^8}{8!} - \frac{t^9}{9!} + \frac{t^{10}}{10!} \right) \\ &\quad + \left( t + t^2 + \frac{t^3}{2!} + \frac{t^4}{3!} + \frac{t^5}{4!} \right. \\ &\quad \left. + \frac{t^6}{5!} + \frac{t^7}{6!} + \frac{t^8}{7!} + \frac{t^9}{8!} + \frac{t^{10}}{9!} \right) \\ &= \sum_{j=0}^{10} (-1)^j \frac{t^j}{j!} + \sum_{j=0}^9 \frac{t^{j+1}}{j!}, \\ x_2^{10}(t) &= 1 + t + \frac{3t^2}{2} + \frac{t^3}{6} - \frac{t^4}{8} + \frac{t^5}{120} + \frac{7t^6}{720} \\ &\quad + \frac{t^7}{5040} - \frac{t^8}{5760} + \frac{t^9}{362880} + \frac{11t^{10}}{3628800} \\ &= \left( 1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} + \frac{t^4}{4!} + \frac{t^5}{5!} \right. \\ &\quad \left. + \frac{t^6}{6!} + \frac{t^7}{7!} + \frac{t^8}{8!} + \frac{t^9}{9!} + \frac{t^{10}}{10!} \right) \\ &\quad + \left( t^2 - \frac{t^4}{3!} + \frac{t^6}{5!} - \frac{t^8}{7!} + \frac{t^{10}}{9!} \right) \\ &= \sum_{j=0}^{10} \frac{t^j}{j!} + \sum_{j=0}^4 (-1)^j \frac{t^{(2j+2)}}{(2j+1)!}, \\ x_3^{10}(t) &= t - \frac{t^3}{6} + \frac{t^5}{120} - \frac{t^7}{5040} \\ &\quad + \frac{t^9}{362880} = \sum_{j=0}^4 (-1)^j \frac{t^{2j+1}}{(2j+1)!}. \end{aligned} \quad (25)$$

Thus, the approximate solutions  $x_i^k(t)$ ,  $i = 1, 2, 3$ , of system (18) and (19) have general forms that coincide with the exact solutions such that

$$\begin{aligned}
 x_1(t) &= \sum_{m=0}^{\infty} c_{1,m} t^m = \sum_{j=0}^{\infty} (-1)^j \frac{t^j}{j!} \\
 &\quad + t \sum_{j=0}^{\infty} \frac{t^j}{j!} = e^{-t} + t e^t, \\
 x_2(t) &= \sum_{m=0}^{\infty} c_{2,m} t^m = \sum_{j=0}^{\infty} \frac{t^j}{j!} \\
 &\quad + t \sum_{j=0}^{\infty} (-1)^j \frac{t^{(2j+1)}}{(2j+1)!} = e^t + t \sin(t), \\
 x_3(t) &= \sum_{m=0}^{\infty} c_{3,m} t^m = \sum_{j=0}^{\infty} (-1)^j \frac{t^{(2j+1)}}{(2j+1)!} = \sin(t).
 \end{aligned} \tag{26}$$

To show the accuracy of the method, numerical results at some selected grid points together with comparison between the absolute errors of RPSM for 8th-order approximations and the ADM [33] are given in Tables 1 and 2. From the tables, it can be seen that the present method provides us with an accurate approximate solution to system (18) and (19). Indeed, the results reported in these tables confirm the effectiveness of the RPS method.

*Example 2.* Consider the following nonlinear system of DAEs [18]:

$$\begin{aligned}
 x_1'(t) &= x_1(t) - x_2(t) x_3(t) + \sin(t) + t \cos(t), \\
 x_2'(t) &= t x_3(t) + x_1^2(t) + \sec^2(t) \\
 &\quad - t^2 (\cos(t) + \sin^2(t)),
 \end{aligned} \tag{27}$$

$$x_1(t) - x_3(t) + t (\cos(t) - \sin(t)) = 0,$$

for  $0 \leq t \leq 1$ , subject to initial conditions

$$x_1(0) = x_2(0) = x_3(0) = 0. \tag{28}$$

The exact solution is  $x_1(t) = t \sin(t)$ ,  $x_2(t) = \tan(t)$ , and  $x_3(t) = t \cos(t)$ .

As we mentioned earlier, if we select the first terms as initial guesses of the approximations as  $x_{1,0}(t) = 0$ ,  $x_{2,0}(t) = 0$ , and  $x_{3,0}(t) = 0$ , then the  $k$ th-truncated series solutions  $x_i^k(t)$ ,  $i = 1, 2, 3$ , for (27) and (28) have the following form:

$$\begin{aligned}
 x_1^k(t) &= \sum_{m=1}^k c_{1,m} t^m = c_{1,1} t + c_{1,2} t^2 + \dots + c_{1,k} t^k, \\
 x_2^k(t) &= \sum_{m=1}^k c_{2,m} t^m = c_{2,1} t + c_{2,2} t^2 + \dots + c_{2,k} t^k, \\
 x_3^k(t) &= \sum_{m=1}^k c_{3,m} t^m = c_{3,1} t + c_{3,2} t^2 + \dots + c_{3,k} t^k,
 \end{aligned} \tag{29}$$

in which the values of the coefficients  $c_{i,m}$ ,  $m = 1, 2, \dots, k$ ,  $i = 1, 2, 3$ , can be found by constructing the  $k$ th residual functions  $\text{Res}_i^k(t)$ ,  $i = 1, 2, 3$ , as follows:

$$\begin{aligned}
 \text{Res}_1^k(t) &= \sum_{m=1}^k m c_{1,m} t^{m-1} - \sum_{m=1}^k c_{1,m} t^m \\
 &\quad + \left( \sum_{m=1}^k c_{2,m} t^m \right) \left( \sum_{m=1}^k c_{3,m} t^m \right) \\
 &\quad - \sin(t) - t \cos(t), \\
 \text{Res}_2^k(t) &= \sum_{m=1}^k m c_{2,m} t^{m-1} - \sum_{m=1}^k c_{3,m} t^{m+1} - \left( \sum_{m=1}^k c_{1,m} t^m \right)^2 \\
 &\quad - \sec^2(t) + t^2 (\cos(t) + \sin^2(t)), \\
 \text{Res}_3^k(t) &= \frac{d}{dt} \left[ \sum_{m=1}^k c_{1,m} t^m - \sum_{m=1}^k c_{3,m} t^m \right. \\
 &\quad \left. + t (\cos(t) + \sin(t)) \right].
 \end{aligned} \tag{30}$$

Using the facts  $\text{Res}_1^1(0) = \text{Res}_2^1(0) = \text{Res}_3^1(0) = 0$  and  $(d/dt)\text{Res}_1^2(0) = (d/dt)\text{Res}_2^2(0) = (d/dt)\text{Res}_3^2(0) = 0$  through (30), we can obtain  $x_1^2(t) = t^2$ ,  $x_2^2(t) = t$  and  $x_3^2(t) = t$  as soon as  $c_{1,1} = 0$ ,  $c_{2,1} = 1$ ,  $c_{3,1} = 1$ ,  $c_{1,2} = 1$ ,  $c_{2,2} = 0$ , and  $c_{3,1} = 0$ . Furthermore, based on  $(d^{k-1}/dt^{k-1})\text{Res}_i^k(0) = 0$ ,  $k = 3, 4, \dots, 10$ ,  $i = 1, 2, 3$ , the 10th truncated series  $x_i^{10}(t)$ ,  $i = 1, 2, 3$ , of the RPS solution for system (27) and (28) are given by

$$\begin{aligned}
 x_1^{10}(t) &= t^2 - \frac{t^4}{6} + \frac{t^6}{120} - \frac{t^8}{7!} + \frac{t^{10}}{9!}, \\
 x_2^{10}(t) &= t + \frac{t^3}{3} + \frac{2t^5}{15} + \frac{17t^7}{315} + \frac{62t^9}{2835}, \\
 x_3^{10}(t) &= t - \frac{t^3}{2} + \frac{t^5}{24} - \frac{t^7}{720} + \frac{t^9}{8!}.
 \end{aligned} \tag{31}$$

Therefore, the approximate solutions of system (27) and (28) can be expressed as

$$\begin{aligned}
 x_1(t) &= \lim_{k \rightarrow \infty} x_1^k(t) = t^2 - \frac{t^4}{6} + \frac{t^6}{120} - \frac{t^8}{7!} + \frac{t^{10}}{9!} + \dots \\
 &= t \left( t - \frac{t^3}{3!} + \frac{t^5}{5!} - \frac{t^7}{7!} + \frac{t^9}{9!} + \dots \right) = t \sin(t), \\
 x_2(t) &= \lim_{k \rightarrow \infty} x_2^k(t) = t + \frac{t^3}{3} + \frac{2t^5}{15} \\
 &\quad + \frac{17t^7}{315} + \frac{62t^9}{2835} + \dots = \tan(t),
 \end{aligned}$$

TABLE 1: Numerical comparison of the exact error function  $\text{Ext}^8(t)$  for  $x_1(t)$  to Example 1 in  $[0, 1]$ .

$t_i$	Exact solution	Approximate solution	ADM [33]	Present method
0.1	1.0153545098435244	1.0153545098435020	$0.1100000 \times 10^{-7}$	$2.24265 \times 10^{-14}$
0.2	1.0630113047100158	1.0630113046984127	$0.1050000 \times 10^{-6}$	$1.16032 \times 10^{-11}$
0.3	1.1457758629545190	1.1457758625022320	$0.1100000 \times 10^{-5}$	$4.52287 \times 10^{-10}$
0.4	1.2670499250921474	1.2670499189841269	$0.6300000 \times 10^{-5}$	$6.10802 \times 10^{-9}$
0.5	1.4308912950626975	1.4308912489149306	$0.2495100 \times 10^{-4}$	$4.61478 \times 10^{-8}$
0.6	1.6420829163283317	1.6420826748571429	$0.7771800 \times 10^{-4}$	$2.41471 \times 10^{-7}$
0.7	1.9062121990207430	1.9062112184149302	$0.2052320 \times 10^{-3}$	$9.80606 \times 10^{-7}$
0.8	2.2297617069111960	2.2297583989841270	$0.4805960 \times 10^{-3}$	$3.30793 \times 10^{-6}$
0.9	2.6202124597818544	2.6202027750022320	$0.1027034 \times 10^{-2}$	$9.68478 \times 10^{-6}$
1.0	3.0861612696304874	3.0861359126984130	$0.2042608 \times 10^{-2}$	$2.53569 \times 10^{-5}$

TABLE 2: Numerical comparison of the exact error function  $\text{Ext}^8(t)$  for  $x_2(t)$  to Example 1 in  $[0, 1]$ .

$t_i$	Exact solution	Approximate solution	ADM [33]	Present method
0.1	1.1151542597403306	1.1151542597403274	$0.2000000 \times 10^{-8}$	$3.10862 \times 10^{-15}$
0.2	1.2611366243191822	1.2611366243174604	$0.9200000 \times 10^{-7}$	$1.72173 \times 10^{-12}$
0.3	1.4385148695744050	1.4385148695022323	$0.1061000 \times 10^{-5}$	$7.21727 \times 10^{-11}$
0.4	1.6475920345647306	1.6475920335238095	$0.6065000 \times 10^{-5}$	$1.04092 \times 10^{-9}$
0.5	1.8884340400022297	1.8884340316530257	$0.2355100 \times 10^{-4}$	$8.34920 \times 10^{-9}$
0.6	2.1609042844275300	2.1609042382857147	$0.7165000 \times 10^{-4}$	$4.61418 \times 10^{-8}$
0.7	2.4647050885368604	2.4647048915260410	$0.1842280 \times 10^{-3}$	$1.97011 \times 10^{-7}$
0.8	2.7994258012120863	2.7994251052698410	$0.4188780 \times 10^{-3}$	$6.95942 \times 10^{-7}$
0.9	3.1645973298216850	3.1645952037879463	$0.8671680 \times 10^{-3}$	$2.12603 \times 10^{-6}$
1.0	3.5597528132669414	3.5597470238095240	$0.1667496 \times 10^{-2}$	$5.78946 \times 10^{-6}$

$$\begin{aligned}
 x_3(t) &= \lim_{k \rightarrow \infty} x_3^k(t) = t - \frac{t^3}{2} + \frac{t^5}{24} - \frac{t^7}{720} + \frac{t^9}{8!} + \dots \\
 &= t \left( 1 - \frac{t^2}{2!} + \frac{t^4}{4!} - \frac{t^6}{6!} + \frac{t^8}{8!} + \dots \right) = t \cos(t),
 \end{aligned}
 \tag{32}$$

subject to the initial conditions

$$x_1(0) = 1, \quad x_2(0) = 1, \quad x_3(0) = 1. \tag{34}$$

which are compatible with the exact solutions.

In order to show how the  $k$ th values affect the approximate solutions, we calculate the maximum error functions of  $x_i^k(t)$ ,  $i = 1, 2, 3$ ,  $k = 5, 10, 15, 20$ , for various values of  $t$  in  $[0, 1]$  with step size of 0.1 that are listed in Table 3 by using the  $k$ th order RPS approximate solution. In Table 4, the residual error functions  $\text{Res}_i^k(t)$ ,  $i = 1, 2, 3$ ,  $k = 15, 30$ , have been listed for  $t_i = i/4$ ,  $i = 0, 1, 2, 3, 4$ , to demonstrate the rapid convergence of the present method by increasing the order of RPS approximation. As a result, it is clear from these tables that the approximate solutions are found to be in good agreement with the exact solutions for all values of  $t$  in  $[0, 1]$ .

*Example 3.* Consider the following nonlinear system of DAEs:

$$\begin{aligned}
 x_1'(t) - x_1(t) + x_1(t)x_3(t) + x_3(t) + x_3'(t) &= 1, \\
 x_3'(t) - x_2(t) + x_1^2(t) + x_3(t) &= 0, \\
 2x_2(t) - 2x_1^2(t) &= 0,
 \end{aligned}
 \tag{33}$$

The exact solution is  $x_1(t) = e^t$ ,  $x_2(t) = e^{2t}$ ,  $x_3(t) = e^{-t}$ .

To apply the RPS approach for solving system (33) and (34), if we start with selecting the initial guesses of the approximations such that  $x_{1,0}(t) = 1$ ,  $x_{2,0}(t) = 1$ , and  $x_{3,0}(t) = 1$ , then the  $k$ th-truncated series solutions  $x_i^k(t)$ ,  $i = 1, 2, 3$ , can be given by

$$\begin{aligned}
 x_1^k(t) &= \sum_{m=0}^k c_{1,m} t^m = 1 + c_{1,1}t + c_{1,2}t^2 + \dots + c_{1,k}t^k, \\
 x_2^k(t) &= \sum_{m=0}^k c_{2,m} t^m = 1 + c_{2,1}t + c_{2,2}t^2 + \dots + c_{2,k}t^k, \\
 x_3^k(t) &= \sum_{m=0}^k c_{3,m} t^m = 1 + c_{3,1}t + c_{3,2}t^2 + \dots + c_{3,k}t^k.
 \end{aligned}
 \tag{35}$$



TABLE 3: Maximum error functions of  $x_i^k(t)$ ,  $i = 1, 2, 3$ , for  $k = 5, 10, 15$ , and  $20$  in  $[0, 1]$ .

Description	$k = 5$	$k = 10$	$k = 15$	$k = 20$
$\max \{ \text{Ext}_1^k(t) \}$	$8.13765 \times 10^{-3}$	$2.48923 \times 10^{-8}$	$7.61946 \times 10^{-13}$	$1.11022 \times 10^{-16}$
$\max \{ \text{Ext}_2^k(t) \}$	$9.07411 \times 10^{-2}$	$4.81568 \times 10^{-6}$	$5.00908 \times 10^{-9}$	$5.14231 \times 10^{-11}$
$\max \{ \text{Ext}_3^k(t) \}$	$1.36436 \times 10^{-3}$	$2.73497 \times 10^{-7}$	$4.77396 \times 10^{-14}$	$5.55112 \times 10^{-17}$
$\max \{ \text{Rel}_1^k(t) \}$	$9.67075 \times 10^{-3}$	$2.95819 \times 10^{-8}$	$9.05493 \times 10^{-13}$	$1.57480 \times 10^{-16}$
$\max \{ \text{Rel}_2^k(t) \}$	$5.82642 \times 10^{-2}$	$8.81505 \times 10^{-6}$	$9.16906 \times 10^{-9}$	$9.41293 \times 10^{-11}$
$\max \{ \text{Rel}_3^k(t) \}$	$2.52518 \times 10^{-3}$	$5.06192 \times 10^{-7}$	$8.83572 \times 10^{-14}$	$1.26509 \times 10^{-16}$
$\max \{ \text{Con}_1^k(t) \}$	$8.33333 \times 10^{-3}$	$4.22653 \times 10^{-8}$	$7.64722 \times 10^{-13}$	$1.23098 \times 10^{-16}$
$\max \{ \text{Con}_2^k(t) \}$	$5.39683 \times 10^{-2}$	$6.32775 \times 10^{-6}$	$4.50155 \times 10^{-9}$	$4.62128 \times 10^{-11}$
$\max \{ \text{Con}_3^k(t) \}$	$1.38889 \times 10^{-3}$	$2.75573 \times 10^{-7}$	$4.77396 \times 10^{-14}$	$5.55112 \times 10^{-17}$

TABLE 4: The values of residual error functions  $\text{Res}_i^k(t)$ ,  $i = 1, 2, 3$ , to Example 2 when  $k = 15, 30$ .

$t_i$	$\text{Res}_n^{15}(t_i)$			$\text{Res}_n^{30}(t_i)$		
	$n = 1$	$n = 2$	$n = 3$	$n = 1$	$n = 2$	$n = 3$
0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.25	$8.60423 \times 10^{-15}$	$2.40341 \times 10^{-12}$	$2.77556 \times 10^{-17}$	$5.55112 \times 10^{-17}$	0.00	$2.77556 \times 10^{-17}$
0.50	$2.19794 \times 10^{-9}$	$1.72568 \times 10^{-7}$	$5.55112 \times 10^{-17}$	$4.44089 \times 10^{-16}$	$3.39728 \times 10^{-14}$	$2.77556 \times 10^{-17}$
0.75	$3.15260 \times 10^{-6}$	$1.34741 \times 10^{-4}$	$8.04912 \times 10^{-15}$	$1.00890 \times 10^{-10}$	$7.74385 \times 10^{-9}$	$4.85723 \times 10^{-17}$
1.00	$5.36043 \times 10^{-4}$	$1.82182 \times 10^{-2}$	$8.09630 \times 10^{-13}$	$9.62768 \times 10^{-7}$	$5.76677 \times 10^{-5}$	$5.55112 \times 10^{-17}$

Accordingly, the unknown coefficients  $c_{i,m}$ ,  $m = 1, 2, \dots$ ,  $k, i = 1, 2, 3$ , can be found by constructing the following  $k$ th residual functions  $\text{Res}_i^k(t)$ ,  $i = 1, 2, 3$ , such that

$$\begin{aligned}
 \text{Res}_1^k(t) &= \sum_{m=1}^k m c_{1,m} t^{m-1} - \sum_{m=0}^k c_{1,m} t^m \\
 &\quad + \left( \sum_{m=0}^k c_{1,m} t^m + 1 \right) \sum_{m=0}^k c_{3,m} t^m \\
 &\quad + \sum_{m=1}^k m c_{3,m} t^{m-1} - 1, \\
 \text{Res}_2^k(t) &= \sum_{m=1}^k m c_{3,m} t^{m-1} - \sum_{m=0}^k c_{2,m} t^m \\
 &\quad + \left( \sum_{m=0}^k c_{1,m} t^m \right)^2 + \sum_{m=0}^k c_{3,m} t^m, \\
 \text{Res}_3^k(t) &= \frac{d}{dt} \left[ \sum_{m=0}^k 2c_{2,m} t^m - 2 \left( \sum_{m=0}^k c_{1,m} t^m \right)^2 \right].
 \end{aligned} \tag{36}$$

Now, in order to find the 1st-approximate solutions, put  $k = 1$  through (36) to get that

$$\begin{aligned}
 \text{Res}_1^1(t) &= c_{1,1} - (1 + c_{1,1}t) \\
 &\quad + (2 + c_{1,1}t)(1 + c_{3,1}t) + c_{3,1} - 1,
 \end{aligned}$$

$$\text{Res}_2^1(t) = c_{3,1} - (1 + c_{2,1}t) + (1 + c_{1,1}t)^2 + 1 + c_{3,1}t,$$

$$\text{Res}_3^1(t) = \frac{d}{dt} [2(1 + c_{2,1}t) - 2(1 + c_{1,1}t)^2]. \tag{37}$$

Use the fact that  $\text{Res}_1^1(0) = \text{Res}_2^1(0) = \text{Res}_3^1(0) = 0$ , to get  $c_{1,1} = 1$ ,  $c_{2,1} = 2$ , and  $c_{3,1} = -1$ . Based upon this, the first approximations  $x_i^1(t)$ ,  $i = 1, 2, 3$ , of the RPS solution can be written by  $x_1^1(t) = 1 + t$ ,  $x_2^1(t) = 1 + 2t$ , and  $x_3^1(t) = 1 - t$ . Furthermore, the second approximations  $x_i^2(t)$ ,  $i = 1, 2, 3$ , of the RPS solution for system (33) and (34) are given by

$$\begin{aligned}
 x_1^2(t) &= 1 + t + c_{1,2}t^2, \\
 x_2^2(t) &= 1 + 2t + c_{2,2}t^2, \\
 x_3^2(t) &= 1 - t + c_{3,2}t^2,
 \end{aligned} \tag{38}$$

where the coefficients  $c_{1,2}$ ,  $c_{2,2}$ , and  $c_{3,2}$  will be found by putting  $k = 2$  through (36) and using the facts that  $(d/dt)\text{Res}_1^2(0) = (d/dt)\text{Res}_2^2(0) = (d/dt)\text{Res}_3^2(0) = 0$  such that  $c_{1,2} = 1/2$ ,  $c_{2,2} = 2$ , and  $c_{3,2} = 1/2$ . Therefore, the second approximations for the system are  $x_1^2(t) = 1 + t + (1/2)t^2$ ,  $x_2^2(t) = 1 + 2t + 2t^2$ , and  $x_3^2(t) = 1 - t + (1/2)t^2$ . By continuing with the similar fashion, the 10th-order approximations

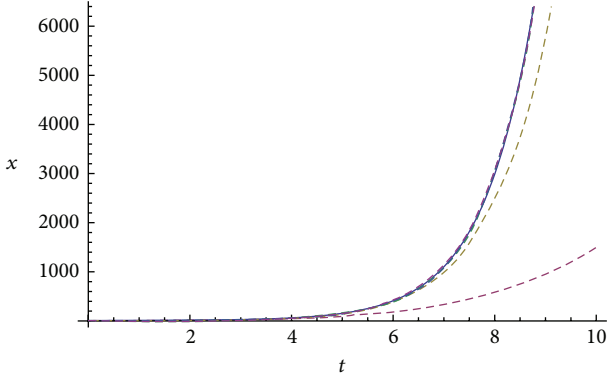


FIGURE 1: Plots of the exact solution  $x_1(t)$  and the RPS solutions  $x_1^k(t)$ ,  $k = 5, 10, 15, 20$ , for (33) and (34) on  $[0, 10]$ , where  $x_1(t)$  and  $x_1^k(t)$  are represented, respectively, by straight and dashed lines.

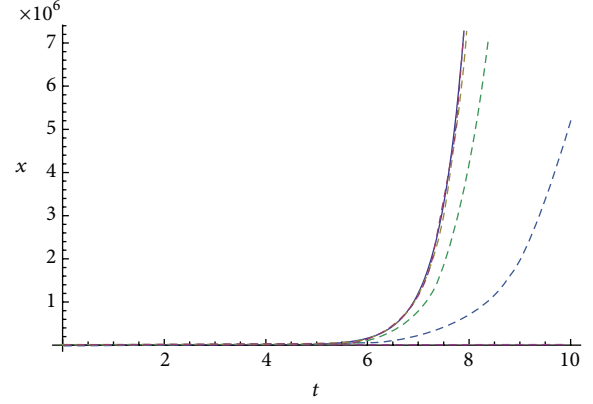


FIGURE 2: Plots of the exact solution  $x_2(t)$  and the RPS solutions  $x_2^k(t)$ ,  $k = 5, 10, 15, 20$ , for (33) and (34) on  $[0, 10]$ , where  $x_2(t)$  and  $x_2^k(t)$  are represented, respectively, by straight and dashed lines.

$x_i^{10}(t)$ ,  $i = 1, 2, 3$ , of the RPS solution lead to the following results:

$$\begin{aligned}
 x_1^{10}(t) &= 1 + t + \frac{1}{2}t^2 + \frac{1}{6}t^3 + \frac{1}{24}t^4 + \frac{1}{120}t^5 \\
 &\quad + \frac{1}{720}t^6 + \frac{1}{5040}t^7 + \frac{1}{40320}t^8 \\
 &\quad + \frac{1}{362880}t^9 + \frac{1}{3628800}t^{10}, \\
 x_2^{10}(t) &= 1 + 2t + 2t^2 + \frac{4}{3}t^3 + \frac{2}{3}t^4 \\
 &\quad + \frac{4}{15}t^5 + \frac{4}{45}t^6 + \frac{8}{315}t^7 + \frac{2}{315}t^8 \\
 &\quad + \frac{4}{2835}t^9 + \frac{4}{14175}t^{10}, \\
 x_3^{10}(t) &= 1 - t + \frac{1}{2}t^2 - \frac{1}{6}t^3 + \frac{1}{24}t^4 \\
 &\quad - \frac{1}{120}t^5 + \frac{1}{720}t^6 - \frac{1}{5040}t^7 + \frac{1}{40320}t^8 \\
 &\quad - \frac{1}{362880}t^9 + \frac{1}{3628800}t^{10}.
 \end{aligned} \tag{39}$$

Correspondingly, the general forms of the solutions  $x_1(t)$ ,  $x_2(t)$ , and  $x_3(t)$  for (33) and (34) are given by

$$\begin{aligned}
 x_1(t) &= \sum_{m=0}^{\infty} c_{1,m} t^m = \sum_{j=0}^{\infty} \frac{1}{(j)!} t^j = e^t, \\
 x_2(t) &= \sum_{m=0}^{\infty} c_{2,m} t^m = \sum_{j=0}^{\infty} \frac{1}{(j)!} (2t)^j = e^{2t}, \\
 x_3(t) &= \sum_{m=0}^{\infty} c_{3,m} t^m = \sum_{j=0}^{\infty} \frac{(-1)^j}{(j)!} t^j = e^{-t},
 \end{aligned} \tag{40}$$

which are coinciding with the exact solutions.

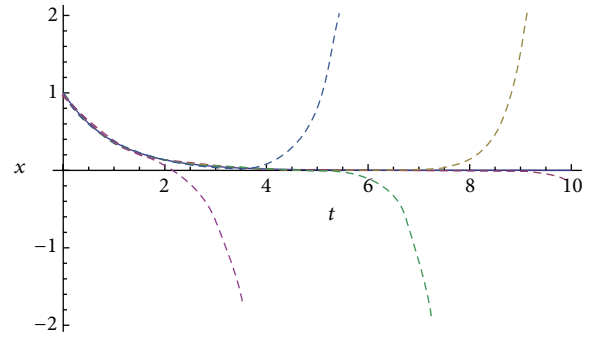


FIGURE 3: Plots of the exact solution  $x_3(t)$  and the RPS solutions  $x_3^k(t)$ ,  $k = 5, 10, 15, 20$ , for (33) and (34) on  $[0, 10]$ , where  $x_3(t)$  and  $x_3^k(t)$  are represented, respectively, by straight and dashed lines.

To illustrate the convergence of the approximate solutions  $x_i(t)$ ,  $i = 1, 2, 3$ , to the exact solutions with respect to the  $k$ th-order of solutions, we present numerical results of Example 3 graphically by Figures 1, 2, and 3, which show the exact solution  $x_i(t)$ ,  $i = 1, 2, 3$ , and some iterated approximations  $x_i^k(t)$ ,  $i = 1, 2, 3$ ,  $k = 5, 10, 15, 20, 25$ , respectively. These graphs reveal that the proposed method is an effective and convenient method for solving nonlinear DAEs systems with less computational and iteration steps.

#### 4. Conclusion

In this paper, a new efficient and accurate numerical technique is proposed and applied to handle the linear and non-linear system of DAEs based on the use of the residual power series method (RPSM). The present technique is performed based mainly on the generation of residual error function and then applying the generalized Taylor series formula. The numerical solutions obtained are an infinite series for appropriate initial conditions. In addition, the RPSM does not require linearization, perturbation, or discretization of the variables, as well as avoiding the round-off errors. The results reveal that the RPSM is a powerful and straightforward tool

and is convenient to handle a various range of engineering problems. The approximate solutions in this analysis are found in the closed form of a convergent series.

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

### Acknowledgment

The authors would like to express their thanks to the unknown referees for their careful reading and helpful comments.

### References

- [1] E. Çelik and M. Bayram, "The numerical solution of physical problems modeled as a system of differential-algebraic equations (DAEs)," *Journal of the Franklin Institute*, vol. 342, no. 1, pp. 1–6, 2005.
- [2] E. Çelik, E. Karaduman, and M. Bayram, "Numerical solutions of chemical differential-algebraic equations," *Applied Mathematics and Computation*, vol. 139, no. 2-3, pp. 259–264, 2003.
- [3] E. Çelik, "On the numerical solution of chemical differential-algebraic equations by Pade series," *Applied Mathematics and Computation*, vol. 153, no. 1, pp. 13–17, 2004.
- [4] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Problems in Differential Algebraic Equations*, Elsevier, New York, NY, USA, 1989.
- [5] E. Çelik and M. Bayram, "Numerical solution of differential-algebraic equation systems and applications," *Applied Mathematics and Computation*, vol. 154, no. 2, pp. 405–413, 2004.
- [6] E. Çelik and M. Bayram, "On the numerical solution of differential-algebraic equations by Padé series," *Applied Mathematics and Computation*, vol. 137, no. 1, pp. 151–160, 2003.
- [7] E. Hairer, C. Lubich, and M. Roche, *The Numerical Solution of DAE Systems by Runge-Kutta*, Springer, Berlin, Germany, 1989.
- [8] M. Saravi, E. Babolian, R. England, and M. Bromilow, "System of linear ordinary differential and differential-algebraic equations and pseudo-spectral method," *Computers & Mathematics with Applications*, vol. 59, no. 4, pp. 1524–1531, 2010.
- [9] E. Çelik and M. Bayram, "Arbitrary order numerical method for solving differential-algebraic equation by Padé series," *Applied Mathematics and Computation*, vol. 137, no. 1, pp. 57–65, 2003.
- [10] C. W. Gear, "The simultaneous numerical solution of differential-algebraic equations," *IEEE Transactions on Circuit Theory*, vol. 18, no. 1, pp. 89–95, 1971.
- [11] U. M. Ascher and L. R. Petzold, "Projected implicit Runge-Kutta methods for differential-algebraic equations," *SIAM Journal on Numerical Analysis*, vol. 28, no. 4, pp. 1097–1120, 1991.
- [12] E. Çelik, E. Karaduman, and M. Bayram, "Numerical method to solve chemical differential-algebraic equations," *International Journal of Quantum Chemistry*, vol. 89, no. 5, pp. 447–451, 2002.
- [13] E. Çelik and T. Yeloglu, "Chebyshev series approximation for solving differential-algebraic equations (DAEs)," *International Journal of Computer Mathematics*, vol. 83, no. 8-9, pp. 651–662, 2006.
- [14] M. M. Hosseini, "Adomian decomposition method for solution of nonlinear differential algebraic equations," *Applied Mathematics and Computation*, vol. 181, no. 2, pp. 1737–1744, 2006.
- [15] M. M. Hosseini, "Adomian decomposition method for solution of differential-algebraic equations," *Journal of Computational and Applied Mathematics*, vol. 197, no. 2, pp. 495–501, 2006.
- [16] F. Soltanian, S. M. Karbassi, and M. M. Hosseini, "Application of He's variational iteration method for solution of differential-algebraic equations," *Chaos, Solitons and Fractals*, vol. 41, no. 1, pp. 436–445, 2009.
- [17] F. Soltanian, M. Dehghan, and S. M. Karbassi, "Solution of the differential algebraic equations via homotopy perturbation method and their engineering applications," *International Journal of Computer Mathematics*, vol. 87, no. 9, pp. 1950–1960, 2010.
- [18] F. Salehi, M. A. Asadi, and M. M. Hosseini, "Solving system of DAEs by modified homotopy perturbation method," *Journal of Computer Science & Computational Mathematics*, vol. 2, no. 6, pp. 1–5, 2012.
- [19] M. Al-Smadi, O. Abu Arqub, and A. El-Ajou, "A numerical iterative method for solving systems of first-order periodic boundary value problems," *Journal of Applied Mathematics*, vol. 2014, Article ID 135465, 10 pages, 2014.
- [20] I. Komashynska and M. AL-Smadi, "Iterative reproducing kernel method for solving second-order integrodifferential equations of Fredholm type," *Journal of Applied Mathematics*, vol. 2014, Article ID 459509, 11 pages, 2014.
- [21] O. Abu Arqub, M. Al-Smadi, and S. Momani, "Application of reproducing kernel method for solving nonlinear Fredholm-Volterra integrodifferential equations," *Abstract and Applied Analysis*, vol. 2012, Article ID 839836, 16 pages, 2012.
- [22] M. Al-Smadi, O. Abu Arqub, and S. Momani, "A computational method for two-point boundary value problems of fourth-order mixed integrodifferential equations," *Mathematical Problems in Engineering*, vol. 2013, Article ID 832074, 10 pages, 2013.
- [23] O. A. Arqub, M. Al-Smadi, and N. Shawagfeh, "Solving Fredholm integro-differential equations using reproducing kernel Hilbert space method," *Applied Mathematics and Computation*, vol. 219, no. 17, pp. 8938–8948, 2013.
- [24] M. Al-Smadi, O. Abu Arqub, and N. Shawagfeh, "Approximate solution of BVPs for 4th-order IDEs by using RKHS method," *Applied Mathematical Sciences*, vol. 6, no. 50, pp. 2453–2464, 2012.
- [25] S. Momani, A. Freihat, and M. AL-Smadi, "Analytical study of fractional-order multiple chaotic FitzHugh-Nagumo neurons model using multistep generalized differential transform method," *Abstract and Applied Analysis*, vol. 2014, Article ID 276279, 10 pages, 2014.
- [26] M. Al-Smadi, A. Freihat, O. Abu Arqub, and N. Shawagfeh, "A novel multistep generalized differential transform method for solving fractional-order Lü chaotic and hyperchaotic systems," *Journal of Computational Analysis and Applications*. In press.
- [27] O. Abu Arqub, "Series solution of fuzzy differential equations under strongly generalized differentiability," *Journal of Advanced Research in Applied Mathematics*, vol. 5, no. 1, pp. 31–52, 2013.
- [28] O. Abu Arqub, A. El-Ajou, A. S. Bataineh, and I. Hashim, "A representation of the exact solution of generalized Lane-Emden equations using a new analytical method," *Abstract and Applied Analysis*, vol. 2013, Article ID 378593, 10 pages, 2013.
- [29] M. Al-Smadi, "Solving initial value problems by residual power series method," *Theoretical Mathematics & Applications*, vol. 3, no. 1, pp. 199–210, 2013.
- [30] A. El-Ajou, O. Abu Arqub, Z. Al Zhour, and S. Momani, "New results on fractional power series: theories and applications," *Entropy*, vol. 15, no. 12, pp. 5305–5323, 2013.

- [31] O. A. Arqub, A. El-Ajou, Z. A. Zhou, and S. Momani, "Multiple solutions of nonlinear boundary value problems of fractional order: a new analytic iterative technique," *Entropy*, vol. 16, no. 1, pp. 471–493, 2014.
- [32] O. Abu Arqub, Z. Abo-Hammour, R. Al-Badarneh, and S. Momani, "A reliable analytical method for solving higher-order initial value problems," *Discrete Dynamics in Nature and Society*, vol. 2013, Article ID 673829, 12 pages, 2013.
- [33] E. Çelik, M. Bayram, and T. Yeloglu, "Solution of differential-algebraic equations (DAEs) by adomian decomposition method," *International Journal Pure & Applied Mathematical Sciences*, vol. 3, pp. 93–100, 2006.
- [34] R. V. Dukkupati, *Numerical Methods*, New Age International, New Delhi, India, 2010.





# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

