Piecewise-Truncated Parametric Iteration Method: a Promising Analytical Method for Solving Abel Differential Equations

Jajar Saberi-Nadjafi and Asghar Ghorbani

Department of Applied Mathematics, School of Mathematical Sciences,
Ferdowsi University of Mashhad, Mashhad, Iran

Reprint requests to S. N.; Fax: (+ 98) 511 8828606; E-mail: najafi141@gmail.com or
A. G.; E-mail: as_gh56@yahoo.com

Z. Naturforsch. **65a,** 529 – 539 (2010); received March 31, 2009 / revised October 26, 2009

This paper deals with the analytical approximate solution of Abel differential equations (ADEs) of the type $\mathrm{d}u/\mathrm{d}t = \sum_{k=0}^{m} g_k(t)u^k$ by proposing a new modified version of the parametric iteration method (PIM). The modified algorithm analytically approximates the solution of ADEs in a sequence of subintervals which is continuous everywhere. The local convergence and the stability of the algorithm are discussed in details. Also we show how to characterize the stability function and the region on which the algorithm is presented. Some examples are given to demonstrate the efficiency and accuracy of this algorithm. Comparison with numerical Runge-Kutta methods (RK) shows that the modified algorithm presented in this paper has the advantage of giving an analytical form of the solution, which is not possible in the purely numerical RK techniques. Moreover, the approximations obtained by the new algorithm converge faster than the numerical RK4 methods, as will be shown. The obtained results reveal that the present algorithm is a promising iterative analytical method for solving ADEs. Furthermore, the proposed algorithm provides us with an easy way to modify the rate and region convergence of the solution. Most promising, however, it seems that the newly developed technique can be further implemented easily to solve other nonlinear ordinary differential equations (ODEs) of physical interests.

Key words: Piecewise-Truncated Parametric Iteration Method; Truncated Parametric Iteration Method; Parametric Iteration Method; Abel Differential Equations; Runge-Kutta Methods
PACS numbers: 02.30.Hq, 02.60.Lj

1. Introduction

It is well known that many phenomena in scientific fields can be described by nonlinear ordinary differential equations. These nonlinear models of real-life problems are still difficult to reliably solve either numerically or theoretically. And even if an exact solution is obtainable, the required calculations may be too complicated to be practical, or the resulting solution may be difficult to interpret. Recently, there has been much attention devoted to search for better and more efficient solution methods for determining a solution, approximate or exact, analytical or numerical, to these nonlinear models (see [1 – 10] and the references cited therein). In this direction, recently, the new analytical approximate methods such as the Adomian decomposition method (ADM) [3], the homotopy analysis method (HAM) [5], the variational iteration method (VIM) [7], the homotopy perturbation method (HPM) [9], the parametric iteration method (or fractional iteration method) (PIM) [1] have been proposed to effectively handle nonlinear problems. In this work, we consider the Abel differential equation of the first kind of the following form:

$$u'(t) = \sum_{k=0}^{m} g_k(t)u^k(t), \quad u(t_0) = c, \tag{1}$$

where $g_k(t)$ $(k = 0, 1, \ldots, m)$ are given analytic functions, c is a constant, and u is the solution to be determined later. These differential equations arose in the context of the studies of N. H. Abel [11] on the theory of elliptic functions. An interesting area of this type of equation also arises in biological systems, see [12], and also in physics, engineering, ecology, and economics (see [13]). Thus, methods to be able to solve these equations are of great importance to engineers and scientists. It represents a natural generalization of the well-known Riccati equation. The solution of this equation can be obtained using numerical integra-

0932–0784 / 10 / 0600–0529 $ 06.00 © 2010 Verlag der Zeitschrift für Naturforschung, Tübingen · http://znaturforsch.com

tion methods. Although these methods are more flexible than any other analytical approach, they have their drawbacks. These methods will provide solutions in discretized form, only at two ends of the time interval, thereby making it complicated in achieving a continuous representation. Furthermore, they respond sensitively on the selection of time-step size to be dependable [14]. In contrast, analytical techniques do offer some options in acquiring solutions but they most habitually require some sort of linearization techniques for a successful effort. For those reasons, approximate analytical methods have been basic for the study. It should be emphasized that one of the main advantages of the approximate analytical methods is its ability in providing us a continuous representation of the approximate solution, which allows better information of the solution over the time interval.

Therefore, the strategy that will be pursued in this work rests mainly on establishing a reliable algorithm based on PIM for finding highly accurate analytical approximate solutions of ADEs of the type (1), which provides us with a convenient way to modify the rate and region convergence of the solution. The ease of using this method shows its power and its efficiency. The analyzed examples reveal that the newly developed algorithm is very simple, effective, and much more accurate to solve ADEs. Moreover, it can further be performed readily to accurately simulate solutions of nonlinear equations and systems of ODEs of physical interests.

2. The Basic Idea of PIM

In this section, we describe PIM for solving the ADEs of (1). The PIM method gives rapidly convergent successive approximations of the exact solution if such a solution exists, otherwise approximations can be used for numerical purposes.

2.1. Parametric Iteration Method

The idea of PIM is very simple and straightforward. For example, consider a differential equation as follows:

$$A[u(t)] = 0, \tag{2}$$

where A is a nonlinear operator, t denotes the time, and $u(t)$ is an unknown variable. To explain the basic idea of PIM, we first consider (2) as below:

$$L[u(t)] + R[u(t)] + N[u(t)] = f(t), \tag{3}$$

where L with the property $Lf \equiv 0$ when $f \equiv 0$ denotes the so-called auxiliary linear operator with respect to u, R the linear differential operator with respect to u, N is a nonlinear operator with respect to u, and $f(t)$ is the source term. We then construct a family of iterative processes for (2) as [1]:

$$\begin{aligned} u_{n+1}(t) = u_n(t) - h\int_0^t \Lambda_{t,s}(H(s)\{L[u_n(s)] \\ + R[u_n(s)] + N[u_n(s)] - f(s)\})\mathrm{d}s, \end{aligned} \tag{4}$$

with the property

$$u_n(t_0) = u(t_0), \quad \forall n, \tag{5}$$

where $u_0(t)$ is the initial guess (which can be freely chosen with possible unknown constants, or it can also be solved from its corresponding linear homogeneous equation $L[u_0(t)] = 0$), the subscript n denotes the nth iteration, and $h \neq 0$, $H(t) \neq 0$, and $\Lambda_{(t,s)} \neq 0$ denote the so-called auxiliary parameter, auxiliary function, and a general multiplier, respectively, which can be identified easily and efficiently by the techniques proposed in this paper below. It should be emphasized that though we have the great freedom to choose the auxiliary linear operator L, the auxiliary parameter h, the auxiliary function $H(t)$, the multiplier $\Lambda_{(t,s)}$, and the initial approximation $u_0(t)$, which is fundamental to the validity and flexibility of PIM, we can also assume that all of them are properly chosen so that solution of (4) exists, as will be shown in this paper later. Accordingly, the successive approximations $u_n(t), n \geq 1$, of PIM in the auxiliary parameter h will be readily obtained by choosing the zeroth component. Consequently, the exact solution may be obtained by using

$$u(t) = \lim_{n\to\infty} u_n(t). \tag{6}$$

It is important to note that for linear problems, its exact solution can be obtained easily by only one iteration step due to the fact that the multiplier can be suitably identified.

2.2. A Simple Technique for Determining the Multiplier

In this subsection, we give an easy and practical way to identify the multiplier $\Lambda_{(t,s)}$ of PIM for (4) in general. For this purpose, we first consider the auxiliary linear operator L, linear differential operator R, and nonlinear operator N as

$$L[u(t)] = \alpha_1(t)u' + \alpha_0(t)u, \quad \alpha_1(t) \neq 0, \tag{7}$$

$$R[u(t)] = [1-\alpha_1(t)]u' - [g_1(t)+\alpha_0(t)]u, \tag{8}$$

$$N[u(t)] = -\sum_{k=2}^{m} g_k(t)u^k, \tag{9}$$

which gives us

$$L[u(t)]+R[u(t)]+N[u(t)] = g_0(t). \tag{10}$$

According to (4), its iterative relation can be written in the following form:

$$\begin{aligned} u_{n+1}(t) = u_n(t) - h\int_0^t \Lambda_{(t,s)}(H(s)\{\alpha_1(s)u_n'(s) \\ + \alpha_0(s)u_n(s)+R[u_n(s)]+N[u_n(s)]-g_0(s)\})\mathrm{d}s. \end{aligned} \tag{11}$$

Using a simple integration by parts, the relation (11) becomes

$$\begin{aligned} u_{n+1}(t) &= u_n(t) - h[-\alpha_1(t_0)\Lambda_{(t,t_0)}H(t_0)u_n(t_0)] \\ &- h\underbrace{[\alpha_1(t)\Lambda_{(t,t)}H(t)]}_{-1}u_n(t) \\ &+ h\int_0^t \underbrace{\left[\frac{\partial}{\partial s}\alpha_1(s)\Lambda_{(t,s)}H(s) - \alpha_0(s)\Lambda_{(t,s)}H(s)\right]}_{0}u_n(s)\mathrm{d}s \\ &- h\int_0^t \Lambda_{(t,s)}(H(s)\{R[u_n(s)]+N[u_n(s)]-g_0(s)\})\mathrm{d}s. \end{aligned} \tag{12}$$

Due to the fact that the following iterative formula also be the other one for solving (1)

$$\begin{aligned} u_{n+1}(t) = (1+h)u_n(t) - h\underbrace{[-\alpha_1(t_0)\Lambda_{(t,t_0)}H(t_0)u_n(t_0)]}_{u_0(t)} \\ -h\int_0^t \Lambda_{(t,s)}(H(s)\{R[u_n(s)]+N[u_n(s)]-g_0(s)\})\mathrm{d}s, \end{aligned} \tag{13}$$

the following simple linear conditions to identify the multiplier must be satisfied:

$$\begin{aligned} &\frac{\partial}{\partial s}(\alpha_1(s)\Lambda_{(t,s)}H(s)) - \alpha_0(s)\Lambda_{(t,s)}H(s) = 0, \\ &\alpha_1(t)\Lambda_{(t,t)}H(t) = -1, \end{aligned} \tag{14}$$

which, for simplicity of the solving procedure, we always put $H(t) = 1$. So, the multiplier can be determined easily and efficiently by solving the conditions (14) as follows:

$$\Lambda_{(t,s)} = \frac{-1}{\alpha_1(t)}\exp\left(\int_s^t \frac{\alpha_1'(\xi)-\alpha_0(\xi)}{\alpha_1(\xi)}\mathrm{d}\xi\right). \tag{15}$$

As a result, according to (4), we will have the following original parametric iteration formula:

$$\begin{aligned} u_{n+1}(t) = u_n(t) + \frac{h}{\alpha_1(t)}\int_0^t \exp\left(\int_s^t \frac{\alpha_1'(\xi)-\alpha_0(\xi)}{\alpha_1(\xi)}\mathrm{d}\xi\right) \\ \cdot\left(H(s)\{u_n'(s) - \sum_{k=0}^{m} g_k(s)u_n^k(s)\}\right)\mathrm{d}s. \end{aligned} \tag{16}$$

Remark 2.2.1. In general for a nth-order ordinary differential equations with the following auxiliary linear operator

$$\begin{aligned} L = \alpha_n(t)\frac{\mathrm{d}^n}{\mathrm{d}t^n} + \alpha_{n-1}(t)\frac{\mathrm{d}^{n-1}}{\mathrm{d}t^{n-1}} + \cdots \\ + \alpha_1(t)\frac{\mathrm{d}}{\mathrm{d}t} + \alpha_0(t), \quad \alpha_n(t) \neq 0, \end{aligned} \tag{17}$$

the multiplier in a similar way can be easily identified by solving the following homogeneous linear equation:

$$\begin{aligned} &(-1)^n\frac{\partial^n}{\partial s^n}(\alpha_n(s)\Lambda_{(t,s)}) \\ &+ (-1)^{n-1}\frac{\partial^{n-1}}{\partial s^{n-1}}(\alpha_{n-1}(s)\Lambda_{(t,s)}) + \cdots \\ &- \frac{\partial}{\partial s}(\alpha_n(s)\Lambda_{(t,s)}) + \alpha_n(s)\Lambda_{(t,s)} = 0, \end{aligned} \tag{18}$$

subject to the conditions

$$\begin{aligned} &\sum_{j=1}^{i}(-1)^{i+j}\frac{\partial^{i-j}}{\partial s^{i-j}}(\alpha_{n-j+1}(s)\Lambda_{(t,s)})_{s=t} = \\ &\chi_{n-i+1} - 1, \quad i = 1,\cdots,n, \end{aligned} \tag{19}$$

where $\chi_n = 0$ when $n \leq 0$ and $\chi_n = 1$ when $n > 1$.

2.3. Convergence Theorem

The parametric iteration formula of (16) produces a recurrence sequence $\{u_n(t)\}$. Obviously, the limit of the sequence will be the solution of (1) if the sequence is convergent. Here, $C^n[0,T]$ denotes the class of all real valued functions defined on $[0,T]$ which have continuous nth order derivative.

Lemma 2.3.1. *If for every n, $u_n \in C^1[0,T]$, then the parametric iteration formula (16) is equivalent to the following iterative relation:*

$$L[u_{n+1}(t) - u_n(t)] = hH(t)A[u_n(t)], \tag{20}$$

where L is as noted above, (7), and $A[u_n(t)] = u_n'(t) - \sum_{k=0}^{m} g_k(t)u_n^k(t)$.

Proof. Suppose u_n and u_{n+1} satisfy the parametric iteration formula (16). Applying $\frac{\mathrm{d}}{\mathrm{d}t}$ to both sides of (16) results in

$$\frac{\mathrm{d}}{\mathrm{d}t}[u_{n+1}(t)u_n(t)] = -h\int_0^t \frac{\partial \Lambda_{(t,s)}}{\partial t} H(s)A[u_n(s)]\mathrm{d}s \\ -hH(t)\Lambda_{(t,t)}A[u_n(t)]. \tag{21}$$

Now, using the conditions (14) and $\partial\Lambda_{(t,s)}/\partial t = -\frac{\alpha_0(t)}{\alpha_1(t)}\Lambda_{(t,s)}$ in (21), we will have

$$\alpha_1(t)\frac{\mathrm{d}}{\mathrm{d}t}[u_{n+1}(t)-u_n(t)] \\ +\alpha_0(t)[u_{n+1}(t)-u_n(t)] = hH(t)A[u_n(t)]. \tag{22}$$

From the definition (7) of L, we obtain

$$L[u_{n+1}(t)-u_n(t)] = hH(t)A[u_n(t)]. \tag{23}$$

Conversely, suppose u_n and u_{n+1} satisfy (20). Multiplying (20) by $\Lambda_{(t,s)}$, in view of the definition of L and $\Lambda_{(t,s)} \neq 0$, and then integrating on both sides of the resulted term yields

$$\int_0^t \Lambda_{(t,s)}\alpha_1(s)[u'_{n+1}(s)-u'_n(s)]\mathrm{d}s \\ +\int_0^t \Lambda_{(t,s)}\alpha_0(s)[u_{n+1}(s)-u_n(s)]\mathrm{d}s \\ = h\int_0^t \Lambda_{(t,s)}H(s)A[u_n(s)]\mathrm{d}s. \tag{24}$$

Using simple integration by parts, the expression (24) becomes

$$\alpha_1(t)\Lambda_{(t,t)}[u_{n+1}(t)-u_n(t)] \\ -\int_0^t \left(\frac{\partial}{\partial s}(\alpha_1(s)\Lambda_{(t,s)}) - \alpha_0(s)\Lambda_{(t,s)}\right) \\ [u_{n+1}(s)-u_n(s)]\mathrm{d}s = h\int_0^t \Lambda_{(t,s)}H(s)A[u_n(s)]\mathrm{d}s, \tag{25}$$

which exactly results in (16) upon imposing the conditions (14), i.e.,

$$u_{n+1}(t) = u_n(t) - h\int_0^t \Lambda_{(t,s)}H(s)A[u_n(s)]\mathrm{d}s, \tag{26}$$

and this ends the proof. □

Theorem 2.3.2. *If the sequence (6) converges, where $u_n(t)$ is produced by the iterative relationship of (16), then it is the exact solution of (1).*

Proof. If the sequence $u_n(t)$ converges, we can write

$$v(t) = \lim_{n\to\infty} u_n(t) \tag{27}$$

or

$$v(t) = \lim_{n\to\infty} u_{n+1}(t). \tag{28}$$

Using (27) and (28), and the definition (7) of L, we have

$$\lim_{n\to\infty} L[u_{n+1}(t)-u_n(t)] = L\lim_{n\to\infty}[u_{n+1}(t)-u_n(t)] \\ = 0. \tag{29}$$

From the expression (29) and according to Lemma 2.3.1, we obtain

$$L\lim_{n\to\infty}[u_{n+1}(t)-u_n(t)] = hH(t)\lim_{n\to\infty} A[u_n(t)] \\ = 0, \tag{30}$$

since $h \neq 0$ and $H(t) \neq 0$, this gives us

$$\lim_{n\to\infty} A[u_n(t)] = 0. \tag{31}$$

From (31) and the definition of A, it holds that

$$\lim_{n\to\infty} A[u_n(t)] = \lim_{n\to\infty}\left[u'_n(t) - \sum_{k=0}^m g_k(t)u_n^k(t)\right] \\ = (\lim_{n\to\infty} u_n(t))' - \sum_{k=0}^m g_k(t)(\lim_{n\to\infty} u_n(t))^k \\ = v(t) - \sum_{k=0}^m g_k(t)v^k(t). \tag{32}$$

From (31) and (32), we have

$$v(t) - \sum_{k=0}^m g_k(t)v^k(t) = 0, \quad t \geq t_0. \tag{33}$$

On the other hand, in view of (27), (6), and (5), it holds

$$v(t_0) = \lim_{n\to\infty} u_n(t_0) = u(t_0) = c. \tag{34}$$

Therefore, according to the above two expressions, $v(t)$ must be the exact solution of (1) and this ends the proof. □

It is clear that the convergence of the sequence (6) depends on the auxiliary linear operator L, the auxiliary parameter h, the auxiliary function $H(t)$, the multiplier $\Lambda_{(t,s)}$, and the initial guess $u_0(t)$. Fortunately, PIM provides us with great freedom to choose all of them. Thus, as long as L, h, $H(t)$, $\Lambda_{(t,s)}$, and $u_0(t)$ are so properly chosen that the sequence (6) converges in a region $t_0 \leq t \leq T$, it should converge to the exact solution in this region. Therefore, the combination of the convergence theorem and the freedom of the choice of the auxiliary linear operator L, the auxiliary parameter h, the auxiliary function $H(t)$, the multiplier $\Lambda_{(t,s)}$, and the initial guess $u_0(t)$ establishes the foundation of the validity and flexibility of PIM.

2.4. One Way of Choosing the Auxiliary Parameter

It is important to ensure that a solution series obtained using PIM, which is always as a family of solution expressions in the auxiliary parameter h, is convergent in a large enough region whereby the rate and region convergence of solution series are dependent upon the auxiliary parameter h and thus can be enlarged by choosing a proper value for h. Most important, however, it is how to choose the value of h to make sure that the solution series converges fast enough in a large enough region. Since we have a family of solution expressions in the auxiliary parameter h, hence, regarding h as an independent variable, a simple and practical way of selecting h is to plot the curve of solution's derivatives with respect to h in the initial point [5]. So, if the solution is unique, all of them converge to the same value and hence there exists a horizontal line segment in its figure that corresponds to a region of h. For brevity, we call such a region *the valid region of h*. Thus, if we set h any value in the so-called valid region of h, we are quite sure that the corresponding solution series converges. It is found that, for given initial approximation and the auxiliary function, the valid regions are often nearly the same for a given problem, although up to now we cannot give a mathematical proof in general. In most cases, we can find a proper value of h to ensure that the solution series converges in the whole spatial/temporal regions. Therefore, the curves h provide us with an easy way to show the influence of h on the rate and region convergence of solution series.

2.5. A Truncated PIM

The successive iterations of PIM may be very complex so that the resulting integrals in the relation (16) may not be performed analytically. Also, in general, the application of PIM to ADEs leads to a calculation of unneeded and repeated terms. The unneeded and repeated calculations may or may not lead to faster convergence. In order to completely eliminate these repeated calculations, provided that the integrand of (16) in each of the iterations is expanded in the multivariate Taylor series around $t = t_0$, the following modified version of PIM, called the truncated PIM (TP), is introduced in solving ADEs of (1):

$$u_{n+1}(t) = u_n(t) + h\int_0^t \Phi_n(t,s)\mathrm{d}s, \qquad (35)$$

where $u_0(t) = c$ is the initial guess and $\Phi_n(t,s)$ $(n = 0,1,\ldots)$ are obtained from the following:

$$\begin{aligned} &-\Lambda_{(t,s)}H(s)\left(u_n'(s) - \sum_{k=0}^{m} g_k(s)u_n^k(s)\right) \\ &\quad = \Phi_n(t,s) + O((s-t_0)^{n+1}) + O((t-t_0)^{n+1}). \end{aligned} \qquad (36)$$

It is noteworthy to point out that the TP formula (35) can cancel all the repeated calculations and terms that are not needed as will be shown below. Also, it is capable of solving nonlinear Abel differential equations with the complicated variable coefficients. Furthermore, it can reduce the size of calculations. Bisides, the TP algorithm (35) may solve a nonlinear Abel differential equation exactly if its solution is an algebraic polynomial up to some degree.

2.6. A Piecewise-Truncated PIM

In general, by using the TP algorithm (35), we obtain a series solution, which in practice is a truncated series solution. This series solution gives a good approximation to the exact solution in a small region of t. An easy and reliable way of ensuring validity of the approximations (35) for large t (i. e., $[t_0,T]$) is to determine the solution in a sequence of equal subintervals of t, i. e. $I_i = [t_i,t_{i+1}]$, where $\Delta t = t_{i+1} - t_i$, $i = 0,1,\ldots,N-1$, with $t_N = T$. Therefore, on $[t_i,t_{i+1}]$ according to the relation (35), we can construct the following piecewise approximations of TP, which is called the piecewise TP (PTP):

$$\begin{aligned} u_{i+1,j+1}(t) &= u_{i+1,j}(t) + h\int_i^t \Phi_{i+1,j}(t,s)\mathrm{d}s \\ &= u_{i+1,n_{i+1}}(t;t_i,c_i,h), \\ u_{i+1,0}(t) &= u_{i,n_i}(t_i) = c_i, \quad j = 0,1,\ldots,n_{i+1}-1, \\ &-\Lambda_{(t,s)}H(s)\left(u_{i+1,j}'(s) - \sum_{k=0}^{m} g_k(s)u_{i+1,j}^k(s)\right) \\ &= \Phi_{i+1,j}(t,s) + O((s-t_i)^{j+1}) + O((t-t_i)^{j+1}), \end{aligned} \qquad (37)$$

where $u_{0,n_0}(t_0) = u(t_0) = c = c_0$. Now, we can obtain the n_{i+1}th-order PTP approximation $u_{i+1,n_{i+1}}(t)$ on $[t_i, t_{i+1}]$. Thus, in the light of (37) for $i = 0, 1, \cdots, N-1$, the approximate analytical solution of (1) on the entire interval $[t_0, T]$ can easily be obtained. It should be emphasized that the PIM and TP algorithms provide analytical solutions in $[t_0, T]$, while the PTP technique provides analytical solutions in $[t_i, t_{i+1}]$, which are continuous at the end points of each interval, i. e., $u_{i,n_i}(t_i) = c_i = u_{i+1,n_{i+1}}(t_i), i = 0, 1, \cdots, N-1$.

2.7. RK4 and PTP Solutions

In this subsection, for the sake of comparative purposes, we introduce the basic idea of the fourth-order Runge-Kutta algorithm. In numerical analysis, the Runge-Kutta methods are an important family of implicit and explicit iterative methods for the approximation of solutions of differential equations. One type of the family of Runge-Kutta methods is very commonly used: it is the fourth-order Runge-Kutta method which is often referred as 'RK4'. We will explain the idea behind the numerical RK4 method by considering (1) in the following form:

$$\begin{aligned} &u' = f(t,u), \quad f(t,u) = A(t) + B(t)u + C(t)u^2, \\ &u(t_0) = u_0. \end{aligned} \tag{38}$$

Then, the numerical RK4 method for this problem is given by the following:

$$u_{n+1} = u_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \tag{39}$$

where

$$\begin{aligned} k_1 &= sf(t_n, u_n), \\ k_2 &= sf(t_n + \frac{s}{2}, u_n + \frac{k_1}{2}), \\ k_3 &= sf(t_n + \frac{s}{2}, u_n + \frac{k_2}{2}), \\ k_4 &= sf(t_n + s, u_n + k_3). \end{aligned} \tag{40}$$

The RK4 method has the error per step which is of order s^5, while the total accumulated error is of order s^4.

Following the Subsection 2.6, the n_{i+1}th-order explicit approximate analytical solution via the PTP method for (1) can be written as

$$\begin{aligned} u_{i+1,n_{i+1}}(t) = &\sum_{j=0}^{n_{i+1}} \frac{\gamma_{i,j}(t_i, c_i, h)}{j!}(t - t_i)^j \\ &+ O((t-t_i)^{n_{i+1}+1}), \quad t \in [t_i, t_{i+1}], \end{aligned} \tag{41}$$

where $\gamma_{i,j}(t_i, c_i, h)$ is a coefficient dependent on t_i, c_i, and h. The expression (41) demonstrates that the numerical results of the n_{i+1}th-order PTP method has the error per step which is of order $(\Delta t)^{n_{i+1}+1}$, while the total accumulated error is of order $(\Delta t)^{n_{i+1}}$.

2.8. Analysis of Stability

Applying the PTP method $(h, \Delta t, n_m)$ to the test problem $u' = Lu$, $u(t_0) = u_0$, we get the following iteration scheme:

$$u_{m,n_m} = [R_{h,\Delta t}(h, \omega)]^m u_0, \quad \omega = L\Delta t, \tag{42}$$

where

$$R_{h,\Delta t}(h, \omega) = \sum_{k=0}^{n_m} \mu_0^{n_m,k}(h) \frac{\omega^k}{k!}, \tag{43}$$

with

$$\begin{aligned} &\mu_0^{n_m,k}(h) = (-h)^k \sum_{i=0}^{n_m-k} \binom{k-1+i}{i} (1+h)^i, \\ &k \leq n_m, \end{aligned} \tag{44}$$

$$\mu_0^{n_m,k}(-1) = 1, \quad n \leq m, \tag{45}$$

is the well-known stability function (see, e. g., [15]), which in our case depends on Δt and h. What us interest is the set of all Δt and h such that (42) is numerically stable. We would say that the proof of (45) can be observed in [5].

Definition 2.8.1. *The stability region of a PTP method for given* Δt *and* h *is defined as*

$$\Sigma_{h,\Delta t} := \{h, \omega \in X : |R_{h,\Delta t}(h, \omega)| \leq 1\}. \tag{46}$$

The stability region usually is difficult to characterize in general, except for some special cases [16]. Here, we suggest a geometric approach to discover the stability region. Since the numerical results of the PTP algorithm (41) are always as a finite series in the auxiliary parameter h and the fixed step size $\Delta = \Delta t$, thus, regarding h and Δ as independent variables, a simple way of selecting h with respect to Δ is to plot the curves of the resulting series with respect to h and Δ. So, if the series is convergent, there exists a horizontal segment in its figure called the $h\Delta$-curves that corresponds to a region of h and Δ, i. e. the stability region $\Sigma_{h,\Delta}$. Therefore, if we set h and Δ values in $\Sigma_{h,\Delta}$, we are sure that the PTP method is numerically stable, as will be shown in this paper below. It seems that the stability regions are often nearly the same for a given problem, although up to now we cannot give a mathematical proof in general. In most of the cases, we can find proper values of h and Δ to ensure that the numerical results of the PTP algorithm are stable. Therefore, the $h\Delta$-curves provide us with a simple way to find the stability regions.

In general, by means of the curve of numerical results versus h and Δ, it is easy to find out the corresponding stability region of the PTP algorithm.

3. Implementations

To give a clear overview of the content of this study, several ADEs will be investigated. These equations will be examined by the PTP algorithm, which will ultimately show the efficacy and accuracy of this method. Moreover, the numerical results specify that the approach is easy to implement and accurate whenever applied to ADEs of the first order. All the results given here are calculated by using the symbolic calculus software Maple 11.

3.1. Application to an ADE with Constant Coefficients

As a first example, let us consider the following ADE with constant coefficients.

Example 1. Consider

$$u' = 1 + u - u^3, \tag{47}$$

with the initial condition $u(0) = 0$. In order to solve (47) by using the PTP algorithm, for simplicity, we choose $\Lambda_{(t,s)} = -1$ (i. e., $\alpha_1(t) = 1$ and $\alpha_0(t) = 0$) and $H(t) = 1$. Therefore, in view of (37) and by taking $n_{i+1} = 2(n = 0, 1, \cdots, N-1)$, we can obtain a second order PTP approximation in the subintervals I_i as follows:

$$\begin{aligned} & u_{i+1,2}(t; t_i, c_i, h) \\ & = c_i + [h + h(1+h)](-1 - c_i + c_i^3)(t - t_i) \\ & \quad + \frac{1}{2} h^2 (3c_i^2 - 1)(-1 - c_i + c_i^3)(t - t_i)^2, \end{aligned} \tag{48}$$

where $i = 0, 1, \cdots, N-1$, $t_0 = 0$ and $c_0 = u(0) = 0$. The numerical results of the PTP algorithm, i. e., c_i, $1 \le i \le N$ can be easily obtained by the following:

$$c_{i+1} = u_{i+1,2}(t_{i+1}; t_i, c_i, h), \quad 0 \le i \le N-1. \tag{49}$$

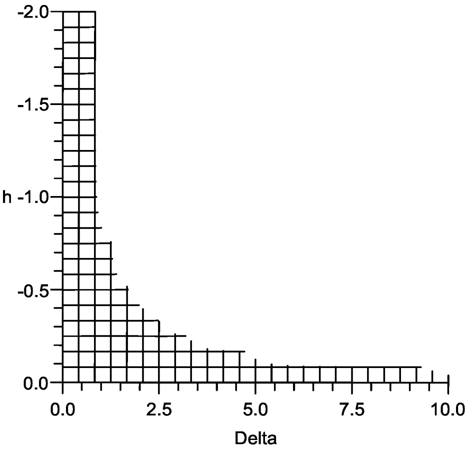

Fig. 1. (Plotting $h\Delta$-curves) Valid region of h with relation to the step size Δ (Delta) for Example 1 by using the numerical result c_2 of the second-order PTP solution.

To investigate the stability region of the solution obtained in (48) via the second-order PTP algorithm, here we plot the curve of c_2 with respect to h and Δ, as shown in Figure 1. According to this curve, it is easy to discover the stability region of (48). We point out that the stability region becomes more accurate as the number i increases. It is usually convenient to investigate the stability region of the PTP method by means of such kinds of curves.

Now, according to (48) and (49), by taking $N = 500$ and $T = 100$, we can obtain the approximations of (47) on $[0, 100]$. Figure 2a shows the approximate solution obtained for (47) by using the second-order PTP algorithm for $h = -0.91$ versus the numerical RK78 solution for a sample interval of t, i. e. $0 \le t \le 100$. It must be emphasized that only the second-order term of our algorithm was used in evaluating the approximate solution for Figure 2a. Also, the difference between the second-order PTP solution (48) for $h = -0.91$ and the numerical RK78 solution has been given in Figure 2b.

3.2. Application to an ADE with Uncomplicated Variable Coefficients

Now, we take a nonlinear Abel differential equation with uncomplicated variable coefficients as follows.

Example 2. Suppose

$$u' = t^4 - 4t^3 u + 6t^2 u^2 - 4tu^3 + u^4, \tag{50}$$

with the initial condition $u(0) = 0$. Proceeding as before, to obtain an acceptable solution of (50)

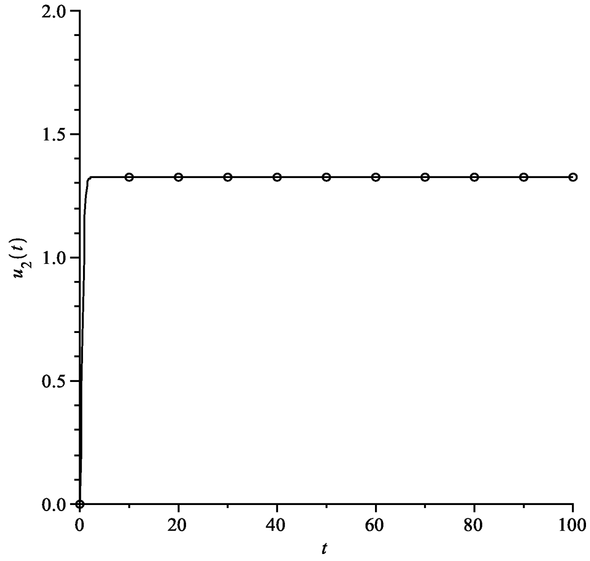

Fig. 2a. Approximate solution for Example 1 using the second-order PTP algorithm when $h = -0.91$ and $\Delta t = 0.2$; second-order PTP solution (solid line) and the numerical RK78 solution (circle symbols).

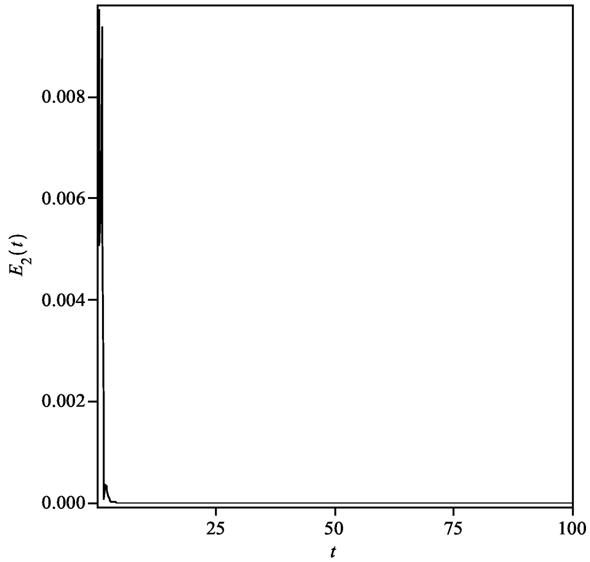

Fig. 2b. Difference between the second-order PTP solution when $h = -0.91$ and $\Delta t = 0.2$, and the numerical RK78 solution for Example 1.

via the PTP algorithm (with $\Lambda_{(t,s)} = -1$ and $H(t) = 1$), we consider a second-order PTP approximation in the subintervals I_i. According to the PTP method (37), by taking $N = 500$ and $T = 100$, we can obtain the approximations of (50) on $[0,100]$.

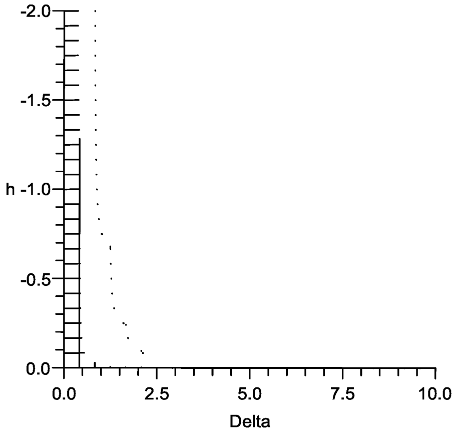

Fig. 3. (Plotting $h\Delta$-curves) Valid region of h with relation to the step size Δ (Delta) for Example 2 by using the numerical result c_2 of the second-order PTP solution.

To investigate the stability region of the solution obtained via the second-order PTP algorithm, we have plotted the curve of c_2 with respect to h and Δ, as shown in Figure 3.

Figure 4a shows the approximate solution obtained for (50) using the second-order PTP algorithm for $h = -1$ versus the numerical RK4 ($s = 0.001$) solution for a sample interval. It is to be noted that only the second-order term of the PTP algorithm was used in evaluating the approximate solutions for Figure 4a. Also our approximate solution applying the new algorithm is in excellent agreement with the numerical values in the large interval of t. Moreover, the difference between the second-order PTP solution for $h = -1$ and the numerical RK4 solution ($s = 0.001$) has been given in Figure 4b.

3.3. Application to an ADE with Complicated Variable Coefficients

The following example corresponds to an ADE with complicated variable coefficients.

Example 3. Assume that

$$u' = \mathrm{e}^{-2\sin t} - 10\,\mathrm{argsin}(\tanh t)u^2 - \mathrm{e}^{-\frac{t^2}{4}}u^4, \quad (51)$$

with the initial condition $u(0) = 0$.

To solve this equation through the PTP algorithm (with $\Lambda_{(t,s)} = -1$ and $H(t) = 1$), we consider a third-order PTP approximation in the subintervals I_i. According to the PTP method (37), by taking $N = 500$ and $T = 100$, we can obtain the approximations of (51)

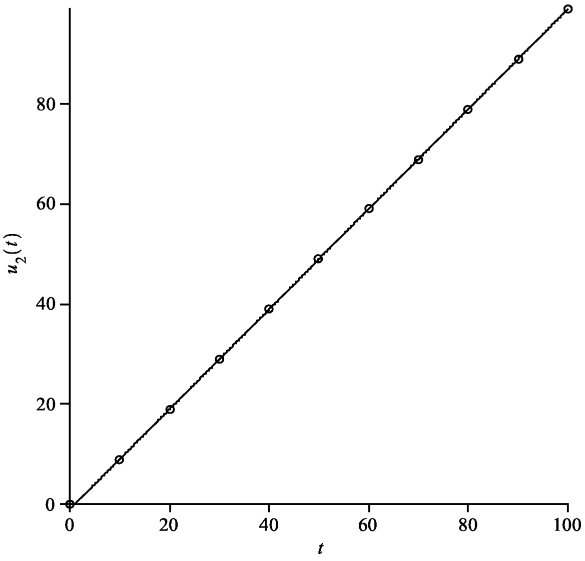

Fig. 4a. Approximate solution for Example 2 using the second-order PTP algorithm when $h = -1$ and $\Delta t = 0.2$; second-order PTP solution (solid line) and the numerical RK4 solution with $s = 0.001$ (circle symbols).

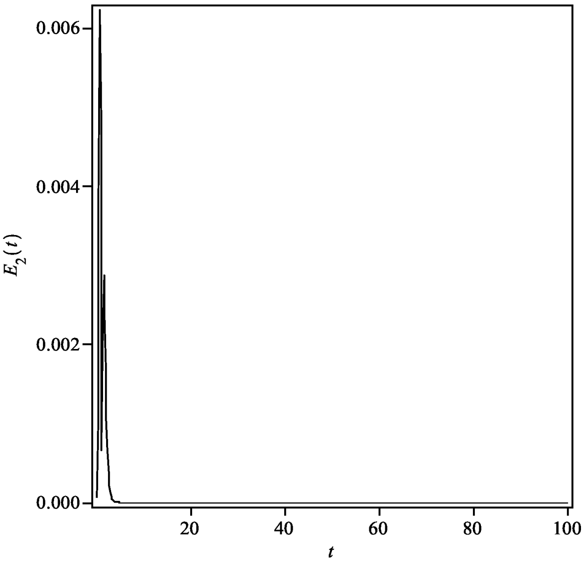

Fig. 4b. Difference between the second-order PTP solution when $h = -1$ and $\Delta t = 0.2$, and the numerical RK4 solution ($s = 0.001$) for Example 1.

on $[0, 100]$. The $h\Delta$-curves (the stability region), the approximate solution for $h = -0.8$, and the absolute error (the difference between the third-order PTP solution for $h = -0.8$ and the numerical RK78 solution) of the third-order PTP algorithm have been given in Figures 5, 6a, and 6b, respectively. From the numerical results of this example, it is easy to conclude that the PTP algorithm is very effective to solve ADEs.

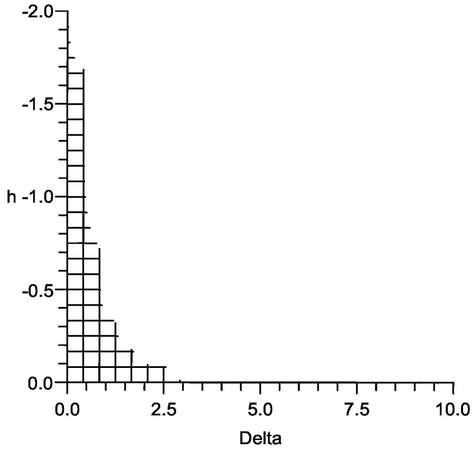

Fig. 5. (Plotting $h\Delta$-curves) Valid region of h with relation to the step size Δ (Delta) for Example 3 by using the numerical result c_2 of the third-order PTP solution.

4. The Comparison with the RK4 Method

It should be emphasized that the convergence of the RK4 method for solving (1) depends mainly on choosing the step size. For example, the numerical RK4 method with the step size $s = 0.2$ for solving (51) of Example 3 is divergent. While, the present PTP algorithm is rather free of this shortcoming, see Figure 6b. This is due to the role of the auxiliary parameter h.

Two important points can be considered here. Firstly, the PTP algorithm has the advantage of providing an analytical form of the solution, which is not possible in the purely numerical RK4 method. Secondly, the PTP algorithm when $n_{i+1} = 4$, $\forall i$ gives an approximate analytical solution which is of comparable accuracy to the RK4 method. In view of (41), in particular, the analytic approximations obtained by using the PTP algorithm when $n_{i+1} > 4$, $\forall i$ converge faster than the numerical RK4 method, e. g., observe Figures 7a and 7b.

The results presented in Examples 1 – 3 indicate that the accuracy of the PTP algorithm, which here use a constant number of iterations and a fixed time step, depends on the characteristics of RDEs to be solved, e. g., variable or constant coefficients, nonlinearities, and other characteristics. These examples also show that the PTP algorithm with $n_{i+1} = 2$ yields results in good accordance with those obtained with MAPLE dverk78 solver. On the other hand, the PTP method

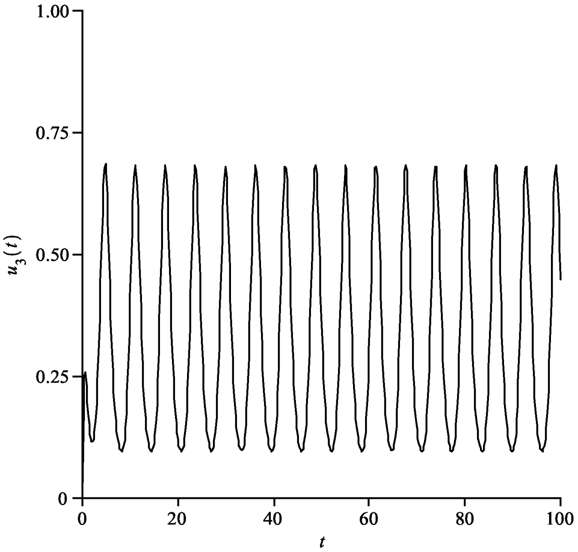

Fig. 6a. Approximate solution for Example 3 using the third-order PTP algorithm when $h = -0.8$ and $\Delta t = 0.2$.

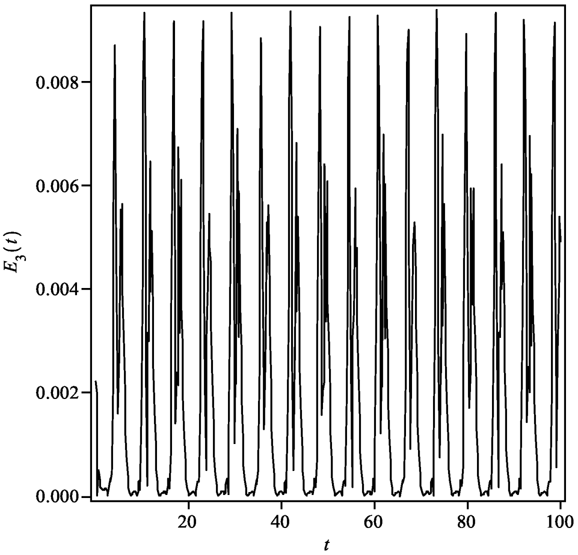

Fig. 6b. Difference between the third-order PTP solution when $h = -0.8$ and $\Delta t = 0.2$, and the numerical RK78 solution for Example 3.

provides piecewise series solutions which are continuous everywhere, whereas the RK4 and dverk78 techniques provide discrete solutions at discrete times.

From the above comparative results, it is easy to wrap up that the PTP algorithm proposed in this paper could lead to a promising analytical method for solving the nonlinear ordinary differential equations and systems.

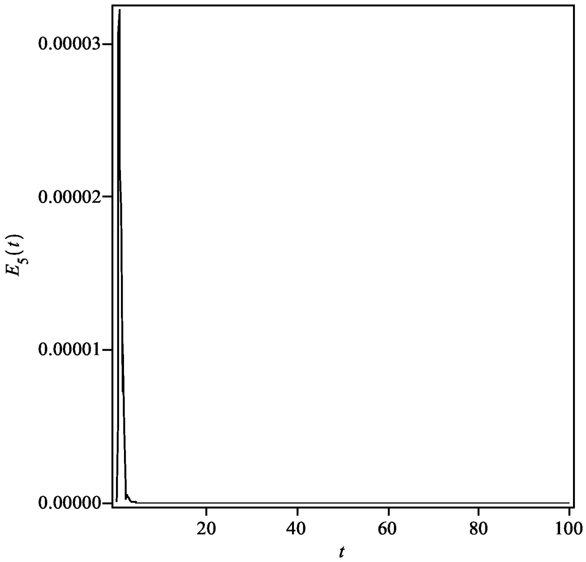

Fig. 7a. Absolute error (the difference between the 5th-order PTP solution and the numerical RK4 solution when $s = 0.001$) of the 5th-order PTP solution when $h = -1$ and $\Delta t = 0.2$ for Example 2.

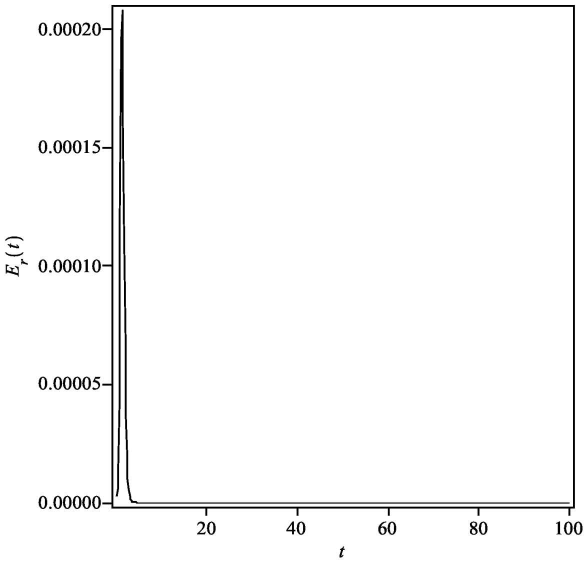

Fig. 7b. Absolute error (the difference between the numerical RK4 solution and the numerical RK4 solution when $s = 0.001$) of the numerical RK4 solution when $s = 0.2$ for Example 2.

5. Concluding Comments

In this paper, the authors proposed a new algorithm called the piecewise-truncated PIM (PTP) for solving the nonlinear Abel differential equations of the first kind. We tested few modelling equations by using the PTP algorithm presented in this work, and the obtained results have shown excellent performance. Different from the numerical RK4 method, the PTP algorithm in solving all the examples given here had stable numerical outcomes. The presented algorithm has the advantage of providing an analytical form of the solution, which is not possible in the purely numerical RK4 method. Also the approximations obtained by the PTP algorithm converge faster than the RK4 method. Moreover, we gave a geometrical scheme for characterizing the stability region of the algorithm. Therefore, the combination of the algorithm and the graphic scheme provides us with a simple way to modify the rate and region convergence of the solution. The results obtained in this work demonstrate that the PTP algorithm is easy to implement, accurate when applied to nonlinear ADEs, and avoids tedious computational works. Generally speaking, the proposed approach can be further carried out readily to solve other more complex nonlinear ordinary differential equations of physical interests.

Acknowledgements

This research is supported by grant No. MA87125 SAB from Center for Research in Modeling and Computation of Linear and Nonlinear Systems, (CRMCS), Ferdowsi University of Mashhad, Mashhad, Iran. The authors would like to express their deep appreciation to the referees for their comments and suggestions.

[1] A. Ghorbani, Comput. Methods Appl. Mech. Eng. **197**, 4173 (2008).

[2] V. Daftardar-Gejji and H. Jafari, J. Math. Anal. Appl. **316**, 753 (2006).

[3] G. Adomian, Solving Frontier Problems of Physics: The Decomposition Method, Kluwer Academic Publishers, Boston 1994.

[4] G. Adomian, J. Math. Anal. Appl. **135**, 501 (1988).

[5] Sh. J. Liao, Beyond perturbation: introduction to the homotopy analysis method. Chapman Hall/CRC Press, Boca Raton 2003.

[6] J. Cang, Y. Tan, H. Xu, and Sh. J. Liao, Chaos, Solitons, and Fractals, **40**, 1 (2009).

[7] J. H. He, Commun. Nonlinear Sci. Numer. Simul. **2**, 230 (1997).

[8] J. H. He, Comput. Methods Appl. Mech. Eng. **167**, 57 (1998).

[9] J. H. He, Comput. Methods Appl. Mech. Eng. **178**, 257 (1999).

[10] J. H. He, Int. J. Nonlinear Mech. **35**, 37 (2000).

[11] N. H. Abel, J. Reine Angew. Math. **4**, 309 (1829).

[12] G. Adomian, G. E. Adomian, and R. E. Bellman, Proc. Natl. Acad. Sci. USA **81**, 2938 (1984).

[13] C. M. Bender and S. A. Orszag, Sec.1.6, in Advanced Mathematical Methods for Scientists and Engineers, Wiley, New York 1986.

[14] M. Yamaguti and S. Ushiki, Physica D **3**, 618 (1981).

[15] K. Burrage, Parallel and Sequential Methods for Ordinary Differential Equations, Oxford University Press, Oxford 1995.

[16] N. S. Hoang, R. B. Sidje, and N. H. Cong, J. Comput. Appl. Math. **198**, 187 (2007).