Hindawi
International Journal of Differential Equations
Volume 2018, Article ID 6043936, 10 pages
https://doi.org/10.1155/2018/6043936

Research Article

Convergent Power Series of sech(x) and Solutions to Nonlinear Differential Equations

U. Al Khawaja [1] **and Qasem M. Al-Mdallal** [2]

[1] *Physics Department, United Arab Emirates University, P.O. Box 15551, Al Ain, UAE*
[2] *Department of Mathematical Sciences, United Arab Emirates University, P.O. Box 15551, Al Ain, UAE*

Correspondence should be addressed to U. Al Khawaja; u.alkhawaja@uaeu.ac.ae

Received 25 September 2017; Revised 7 December 2017; Accepted 8 January 2018; Published 13 February 2018

Academic Editor: Jaume Giné

It is known that power series expansion of certain functions such as sech(x) diverges beyond a finite radius of convergence. We present here an iterative power series expansion (IPS) to obtain a power series representation of sech(x) that is convergent for all x. The convergent series is a sum of the Taylor series of sech(x) and a complementary series that cancels the divergence of the Taylor series for $x \geq \pi/2$. The method is general and can be applied to other functions known to have finite radius of convergence, such as $1/(1 + x^2)$. A straightforward application of this method is to solve analytically nonlinear differential equations, which we also illustrate here. The method provides also a robust and very efficient numerical algorithm for solving nonlinear differential equations numerically. A detailed comparison with the fourth-order Runge-Kutta method and extensive analysis of the behavior of the error and CPU time are performed.

1. Introduction

It is well-known that the Taylor series of some functions diverge beyond a finite radius of convergence [1]. For instance, by way of example not exhaustive enumeration, the Taylor series of sech(x) and $1/(1 + x^2)$ diverge for $x \geq \pi/2$ and $x \geq 1$, respectively. Increasing the number of terms in the power series does not increase the radius of convergence; it only makes the divergence sharper. The radius of convergence can be increased only slightly via some functional transforms [2]. Among the many different methods of solving nonlinear differential equations [3–9], the power series is the most straightforward and efficient [10]. It has been used as a powerful numerical scheme for many problems [11–19] including chaotic systems [20–23]. Many numerical algorithms and codes have been developed based on this method [10–12, 20–24]. However, the above-mentioned finiteness of radius of convergence is a serious problem that hinders the use of this method to wide class of differential equations, in particular the nonlinear ones. For instance, the nonlinear Schrödinger equation (NLSE) with cubic nonlinearity has the sech(x) as a solution. Using the power series method to solve this equation produces the power series of a sech(x), which is valid only for $x < \pi/2$.

A review of the literature reveals that the power series expansion was exploited by several researchers [10–12, 20–24] to develop powerful numerical methods for solving nonlinear differential equations. Therefore, this paper is motivated by a desire to extend these attempts to a develop a numerical scheme with systematic control on the accuracy and error. Specifically, two main advances are presented in this paper: (1) a method of constructing a convergent power series representation of a given function with an arbitrarily large radius of convergence and (2) a method of obtaining analytic power series solution of a given nonlinear differential equation that is free from the finite radius of convergence. Through this paper, we show robustness and efficiency of the method via a number of examples including the chaotic Lorenz system [25] and the NLSE. Therefore, solving the problem of finite radius of convergence will open the door wide for applying the power series method to much larger class of differential equations, particularly the nonlinear ones.

It is worth mentioning that the literature includes several semianalytical methods for solving nonlinear differential

equations; such as homotopy analysis method (HAM), homotopy perturbation method (HPM), and Adomian decomposition method (ADM); for more details see [26–29] and the references therein. Essentially, these methods generate iteratively a series solution for the nonlinear systems where we have to solve a linear differential equation at each iteration. Although these methods prove to be effective in solving *most* of nonlinear differential equations and in obtaining a convergent series solution, they have few disadvantages such as the large number of terms in the solution as the number of iterations increases. One of the most important advantages of the present technique is the simplicity in transforming the nonlinear differential equation into a set of simple algebraic difference equations which can be easily solved.

The paper is thus divided into two, seemingly separated, but actually connected main parts. In the first (Section 2), we show, for a given function, how a convergent power series is constructed out of the nonconverging one. In the second part (Section 3.1), we essentially use this idea to solve nonlinear differential equations. In Section 3.2, we investigate the robustness and efficiency of the method by studying the behavior of its error and CPU time versus the parameters of the method. We summarise our results in Section 4.

2. Iterative Power Series Method

This section describes how to obtain a convergent power series for a given function that is otherwise not converging for all x. In brief, the method is described as follows. We expand the function $f(x)$ in a power series as usual, say around $x = 0$. Then we reexpress the coefficients, $f^{(n)}(x)$, in terms of $f(x)$. This establishes a recursion relation between the higher-order coefficients, $f^{(n)}(0)$, and the lowest order ones, $f^{(0)}(0)$ and $f^{(1)}(0)$, and thus the power series is written in terms of only these two coefficients. Then the series and its derivative are calculated at $x = \Delta$, where Δ is much less than the radius of convergence of the power series. A new power series expansion of $f(x)$ is then performed at $x = \Delta$. Similarly, the higher-order coefficients are reexpressed in terms of the lowest order coefficients $f^{(0)}(\Delta)$ and $f^{(1)}(\Delta)$. The value of the previous series and its derivative calculated at $x = \Delta$ are then given to $f^{(0)}(\Delta)$ and $f^{(1)}(\Delta)$, respectively. Then a new expansion around 2Δ is performed with the lowest order coefficients being taken from the previous series, and so on. This iterative process is repeated N times. The final series will correspond to a convergent series at $x = N\Delta$.

Here is a detailed description of the method. The function $f(x)$ is expanded in a Taylor series, $T^0(x)$, around $x = 0$. The infinite Taylor series is an exact representation of $f(x)$ for $x < R$ where R is the radius of convergence. For $x \geq R$ the series diverges. We assume that x is divided into N small intervals $\Delta = x/N$ such that $\Delta < R$. Expanding $f(x)$ around the beginning of each interval we obtain N convergent Taylor series representing $f(x)$ in each interval

$$T^j(y) = \sum_{n=0}^{\infty} \frac{1}{n!} f^{(n)}(j\Delta)(y - j\Delta)^n, \quad j\Delta \leq y < (j+1)\Delta, \ j = 0, 1, 2, \ldots, N, \tag{1}$$

where $T^j(y)$ denotes the Taylor series expansion of $f(y)$ around $y = j\Delta$ and $f^{(n)}(j\Delta)$ is the nth derivative of $f(y)$ calculated at $y = j\Delta$. It is noted that we use $y \in [(j-1)\Delta, j\Delta]$ as the independent variable for the nth Taylor series expansion to distinguish it from $x = N\Delta$. However, these series can not be combined in a single series since their ranges of applicability are different and do not overlap. To obtain a single convergent power series out of the set of series T^j, we put forward two new ideas, which constitute the basis of our method; namely:

(1) Reexpress $f^{(n)}(y)$ in terms of $f(y)$ as $f^{(n)}(y) = F_n[f(y)]$, where the functional $F_n[f(y)]$ is determined by direct differentiation of $f(y)$ for n times and then reexpressing the result in terms of $f(y)$ only. We conjecture that this is possible for a wide class of functions if not all. At least for the two specific functions considered here, this turned out to be possible. Equation (1) then takes the form

$$T^j(y) = \sum_{n=0}^{\infty} a_n \left(a_0^j\right)(y - j\Delta)^n, \tag{2}$$

where we have renamed $f(j\Delta)$ by a_0^j and $F_n[f(j\Delta)]/n!$ by $a_n(a_0^j)$ for a reason to be obvious in the next section. Thus, the coefficients a_n for all n are determined only by a_0^j.

(2) Calculate a_0^j from T^{j-1} at $j\Delta$

$$a_0^j = T^{j-1}(j\Delta) = \sum_{n=0}^{\infty} a_n \left(a_0^{j-1}\right)\Delta^n, \quad j = 1, 2, \ldots, N, \tag{3}$$

which amounts to assigning the value of the Taylor series at the end of an interval to a_0^j of the consecutive one. Equation (3) captures the essence of the recursive feature of our method; a_0^N is calculated recursively from a_0^0 by repeated action of the right-hand-side on a_0^0. While T^j represents the function f within an interval of width Δ, the sequence a_0^j corresponds to the values of the function at the end of the intervals. In the limit $N \to \infty$, or equivalently $\Delta \to 0$, the discrete set of a_0^j values and $j\Delta$ render to the continuous function $f(x)$ and its independent variable x, respectively. Formally, the convergent power series expansion of $f(x)$ around $x = 0$ will thus be given by

$$f(x) = \lim_{N\to\infty} S^N, \tag{4}$$

where S^N denotes the N^{th} iteration of

$$S[f(0)] = \sum_{n=0}^{\infty} a_n (f(0)) \left(\frac{x}{N}\right)^n. \tag{5}$$

As an illustrative example, we apply the method to $f(x) = \operatorname{sech}(x)$. The infinite Taylor series expansion of this function diverges sharply to infinity at $x = \pi/2$. The first step is to

determine the coefficients a_n^j, which are the coefficients of the T^j series

$$\begin{aligned} T^j(y) = {} & \operatorname{sech}(y)\big|_{y=j\Delta} + [-\operatorname{sech}(y)\tanh(y)]_{y=j\Delta}(y \\ & - j\Delta) + \frac{1}{2!}\left[-\operatorname{sech}^3(y) + \operatorname{sech}(y)\tanh^2(y)\right]_{y=j\Delta} \\ & \cdot (y - j\Delta)^2 + \frac{1}{3!}\left[\left(5\operatorname{sech}^2(y) - \tanh^2(y)\right)\right. \\ & \left.\cdot \operatorname{sech}(y)\tanh(y)\right]_{y=j\Delta}(y - j\Delta)^3 + \frac{1}{4!}\left[5\operatorname{sech}^5(y)\right. \\ & \left. - 18\operatorname{sech}^3(y)\tanh^2(y) + \operatorname{sech}(y)\tanh^4(y)\right]_{y=j\Delta} \\ & \cdot (y - j\Delta)^4 + \cdots. \end{aligned} \tag{6}$$

The next step is to reexpress the higher-order coefficients, a_n^j, in terms of the zeroth-order coefficient $a_0^j = \operatorname{sech}(y)|_{y=j\Delta}$. The property $\operatorname{sech}^2(y) + \tanh^2(y) = 1$ is used to that end. It is noticed, however, that while it is possible to express the even-n coefficients in terms of a_0^j only, the odd-n coefficients can only be expressed terms of both a_0^j and $a_1^j = -\operatorname{sech}(y)\tanh(y)|_{y=j\Delta}$. In the context of solving differential equations using the power series method, this reflects the fact that the solution is expressed in terms of two independent parameters (initial conditions). The sech function is indeed a solution of a second-order differential equation, which is solved using this method in the next section. Equation (6) then takes the form

$$\begin{aligned} T^j(y) = {} & a_0^j + a_1^j(y - j\Delta) + \frac{1}{2!}\left[1 - 2\left(a_0^j\right)^2\right] \\ & \cdot a_0^j(y - j\Delta)^2 + \frac{1}{3!}\left[1 - 6\left(a_0^j\right)^2\right] \\ & \cdot a_1^j(y - j\Delta)^3 \\ & + \frac{1}{4!}\left[1 - 8\left(a_0^j\right)^2 + 8\left(a_0^j\right)^4 - 12\left(a_1^j\right)^2\right] \\ & \cdot a_0^j(y - j\Delta)^4 + \cdots. \end{aligned} \tag{7}$$

Calculating $T^j(y)$ series at the end of its interval of applicability, $y = (j+1)\Delta$, we get

$$\begin{aligned} T^j((j+1)\Delta) = {} & a_0^j + a_1^j\Delta + a_2^j\left(a_0^j\right)\Delta^2 + a_3^j\left(a_0^j, a_1^j\right)\Delta^3 \\ & + a_4^j\left(a_0^j\right)\Delta^4 + \cdots, \end{aligned} \tag{8}$$

where the "recursion" coefficients are given by

$$\begin{aligned} a_2^j\left(a_0^j\right) &= \frac{1}{2!}\left[1 - 2\left(a_0^j\right)^2\right]a_0^j, \\ a_3^j\left(a_0^j, a_1^j\right) &= \frac{1}{3!}\left[1 - 6\left(a_0^j\right)^2\right]a_1^j, \\ a_4^j\left(a_0^j\right) &= \frac{1}{4!}\left[1 - 8\left(a_0^j\right)^2 + 8\left(a_0^j\right)^4 - 12\left(a_1^j\right)^2\right]a_0^j. \end{aligned} \tag{9}$$

Finally, we assign $T^j((j+1)\Delta)$ to a_0^{j+1}

$$\begin{aligned} a_0^{j+1} = {} & a_0^j + a_1^j\Delta + a_2^j\left(a_0^j\right)\Delta^2 + a_3^j\left(a_0^j, a_1^j\right)\Delta^3 \\ & + a_4^j\left(a_0^j\right)\Delta^4 + \cdots. \end{aligned} \tag{10}$$

The second independent coefficient a_1^{j+1} is determined by the derivative of $T^j(y)$ calculated at $y = (j+1)\Delta$

$$\begin{aligned} a_1^{j+1} = {} & a_1^j + 2a_2^j\left(a_0^j\right)\Delta + 3a_3^j\left(a_0^j, a_1^j\right)\Delta^2 + 4a_4^j\left(a_0^j\right)\Delta^3 \\ & + \cdots. \end{aligned} \tag{11}$$

While, in the limit $N \to \infty$, a_0^j corresponds to the function $f(x)$, the sequence a_1^j corresponds to $f'(x)$. Therefore, the power series expansion of $\operatorname{sech}(x)$ and its first derivative are given by

$$\begin{pmatrix} f(x) \\ f'(x) \end{pmatrix} = \lim_{N\to\infty} \begin{pmatrix} a_0 + a_1\left(\frac{x}{N}\right) + a_2(a_0)\left(\frac{x}{N}\right)^2 + a_3(a_0, a_1)\left(\frac{x}{N}\right)^3 + a_4(a_0)\left(\frac{x}{N}\right)^4 + \cdots \\ a_1 + 2a_2(a_0)\left(\frac{x}{N}\right) + 3a_3(a_0, a_1)\left(\frac{x}{N}\right)^2 + 4a_4(a_0)\left(\frac{x}{N}\right)^3 + \cdots \end{pmatrix}^N, \tag{12}$$

where the superscript of the matrix on the right-hand-side, N, denotes the Nth iteration of the matrix. The superscript j has been removed since the functional form of the recursion coefficients does not depend on j. The procedure of calculating the power series recursively is described as follows. First, $a_0 = \operatorname{sech}(0) = 1$ and $a_1 = \operatorname{sech}'(0) = 0$ are substituted in the right-hand-side of the last equation. Then the result of the upper element is taken as the updated value of a_0, and, similarly, the lower element updates a_1. The two updated values are then resubstituted back in the right-hand-side. The process is repeated N times. To obtain an explicit form of the series we truncate the Taylor series at $n_{\max} = 4$ and use $N = 4$ iterations. The resulting expansion takes the form

$$\begin{aligned}\operatorname{sech}(x) = 1 - \frac{1}{2}x^2 + \frac{5}{24}x^4 - 0.0806681x^6 \\ + 0.0302048x^8 + \cdots + 1.4434798 \\ \times 10^{-461}x^{624}.\end{aligned} \tag{13}$$

It is noted that the higher-order coefficients, which correspond to ratios of large integers, are represented in real numbers for convenience. Already with such a small number of iterations, $N = 4$, the number of terms equals 313. By inspection, we find that the number of terms equals $((n_{\max} + 1)^N + 1)/2$. Here, $n_{\max}$ is even due to the fact that sech(x) is an even function.

It is also noted that the series (13) is composed of the Taylor expansion of sech(x) around zero, represented by the first three terms, and a series of higher-order terms generated from the nonlinearity in the recursion relations of a_n. In fact, we prove in the next section that this property holds for any $n_{\max}$, N, and function $f(x)$, provided that the Taylor series of the later exists. Therefore, the power series expansion of sech(x), given by (12), can be put in the suggestive form

$$\operatorname{sech}(x) = T + \lim_{N\to\infty} C(N), \tag{14}$$

where T is the infinite Taylor series of $f(x)$ about $x = 0$ and $C(N)$ is a complementary series. It turns out that the complementary series increases the radius of convergence of T for $x \geq \pi/2$. For finite N, the effect of $C(N)$ is to shift the radius of convergence, R, to a larger value such that $R \to \infty$ for $N \to \infty$. In Figure 1 we plot the convergent power series obtained by the present method as given by (12) using $n_{\max} = 4$ and $N = 100$. The curve is indistinguishable from the plot of sech(x). Both the Taylor series expansion, T, and the complementary series, C, diverge sharply at $x = \pi/2$. Since C is essentially zero for $x < \pi/2$, it will not affect the sum $T + C$. However, its major role is to cancel the divergency for $x \geq \pi/2$. In the limit $N \to \infty$, T will be an exact representative of sech(x) for $x < \pi/2$ and C will equal zero in the same interval. For $x \geq \pi/2$, the divergences in T and C cancel each other with a remainder that is an exact representative of sech(x). In this manner, $T + C$ will represent sech(x) for all x.

For finite values of $n_{\max}$ and N, the series $T + C$ is an approximate representative of sech(x). Truncating the Taylor series at $n_{\max}$ introduces an error of order $\Delta^{n_{\max}+1}$. This error will be magnified N times due the recursive substitutions. The total error is then estimated by

$$\text{error} = \left(\frac{x}{N}\right)^{n_{\max}+1} N. \tag{15}$$

For the parameters used in Figure 1, this error is of order 10^{-6} at $x = 5$. This can be reduced to extremely small values such as 10^{-131} with $n_{\max} = 100$. However, the number of terms in the series $T + C$ will be of order 10^{200} which is extremely large and hinders any analytical manipulations.

As another example, we consider $f(x) = 1/(1 + x^2)$ with Taylor series diverging at $x = 1$. Much of the formulation we followed for the previous case holds here and the specifics of the function alter only the recursion relations, (9):

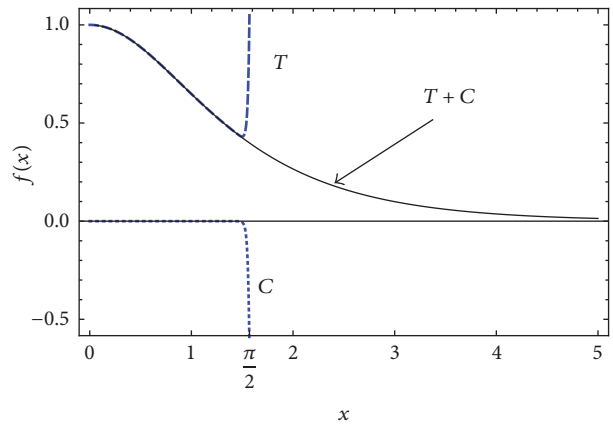

FIGURE 1: The solid curve corresponds to the convergent power series obtained by the present method as given by (12) using $n_{\max} = 4$ and $N = 100$. It is indistinguishable from the solid curve of $f(x) = \operatorname{sech}(x)$. Dashed curve corresponds to the Taylor series expansion, T, and the dotted curve corresponds to the complementary series, C.

$$a_2^j\left(a_0^j\right) = \left[3 - 4a_0^j\right]\left(a_0^j\right)^2, \tag{16}$$

$$a_3^j\left(a_0^j, a_1^j\right) = 2\left[1 - 2a_0^j\right]a_0^j a_1^j, \tag{17}$$

$$\begin{aligned}&a_4^j\left(a_0^j\right) \\ &\quad = \left[16 - 64a_0^j + 85\left(a_0^j\right)^2 - 52\left(a_0^j\right)^3 + 16\left(a_0^j\right)^4\right]a_0^j.\end{aligned} \tag{18}$$

The convergent power series is obtained by using these recursion relations in (12). Plots similar to those of Figure 1 are obtained.

We present now a proof that the convergent power series produced by the recursive procedure always regenerates the Taylor series in addition to a complementary one.

Proposition 1. *If we expand $f(x)$ in a Taylor series, T, around $x = 0$ truncated at $n_{\max}$ and use the recursive procedure, as described above, the resulting convergent power series always takes the form $T + C$ where C is a power series of orders larger than $n_{\max}$. This is true for any number of iterations, N, maximum power of the Taylor series, $n_{\max}$, and for all functions that satisfy the general differential equation*

$$f''(x) = F\left[f(x)\right], \tag{19}$$

where $F[\cdot]$ is an analytic real functional that does not contain derivatives.

Proof. It is trivial to prove this for a specific case, such as sech(x). For the general case, we prove this only for $n_{\max} = 4$ and $N = 2$. The Taylor series expansion of sech(x) around $x = 0$ is

$$\operatorname{sech}(x) = 1 - \frac{1}{2}x^2 + \frac{5}{24}x^4 + \cdots. \tag{20}$$

In our recursive procedure, this is put in the equivalent form

$$\begin{pmatrix} \operatorname{sech}(\Delta) \\ \operatorname{sech}'(\Delta) \end{pmatrix} \simeq \begin{pmatrix} a_0 + a_1\Delta + \dfrac{1-2a_0^2}{2}\Delta^2 + \dfrac{1}{3!}a_1\left(1-6a_0^2\right)\Delta^3 + \dfrac{1}{4!}a_0\left(1-8a_0^2+12a_0^4-12a_1^2\right)\Delta^4 \\ \left(1-2a_0^2\right)\Delta + \dfrac{1}{2}a_1\left(1-6a_0^2\right)\Delta^2 + \dfrac{1}{6}a_0\left(1-8a_0^2+12a_0^4-12a_1^2\right)\Delta^3 \end{pmatrix}^N, \tag{21}$$

where the approximation stems from using finite N and $n_{\max}$, and $\Delta = x/N$. For $N = 1$, $a_0 = 1$, and $a_1 = 0$, (20) is regenerated. However, in our recursive procedure a_0 and a_1 are kept as variables since they will be substituted for at each recursive step. Only at the last step are their numerical values inserted. For $N = 2$, we resubstitute in the last equation for a_0 and a_1 by their updated expressions, as follows:

$$\begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \longrightarrow \begin{pmatrix} a_0 + a_1\Delta + \dfrac{1-2a_0^2}{2}\Delta^2 + \dfrac{1}{3!}a_1\left(1-6a_0^2\right)\Delta^3 + \dfrac{1}{4!}a_0\left(1-8a_0^2+12a_0^4-12a_1^2\right)\Delta^4 \\ \left(1-2a_0^2\right)\Delta + \dfrac{1}{2}a_1\left(1-6a_0^2\right)\Delta^2 + \dfrac{1}{6}a_0\left(1-8a_0^2+12a_0^4-12a_1^2\right)\Delta^3 \end{pmatrix}. \tag{22}$$

Substituting the updated expressions for a_0 and a_1 in (21), we get

$$\begin{aligned} \operatorname{sech}(x) \simeq\ & a_0 + a_1\left(\frac{x}{N}\right) + 2a_0\left(1-2a_0^2\right)\left(\frac{x}{N}\right)^2 \\ & + \frac{4}{3}a_1\left(1-6a_0^2\right)\left(\frac{x}{N}\right)^3 \\ & + \frac{2}{3}a_0\left(1-8a_0^2+12a_0^4-12a_1^2\right)\left(\frac{x}{N}\right)^4. \end{aligned} \tag{23}$$

Clearly for $N = 2$, the last equation gives the Taylor expansion, that is, (21) with $N = 1$. The complimentary series, C, is absent here since we have terminated the expansions at $n = n_{\max} = 4$. For $N = 3$, another step of resubstituting updated expressions is needed, and so on.

Now, we present the proof for the more general case, namely, when $f(x)$ is unspecified but is a solution to (19). We start with the following Taylor series expansion of $f(x)$ and its derivative

$$\begin{aligned} & \begin{pmatrix} f(\Delta) \\ f'(\Delta) \end{pmatrix} \\ & = \begin{pmatrix} a_0 + a_1\Delta + a_2(a_0,a_1)\Delta^2 + a_3(a_0,a_1)\Delta^3 + a_4(a_0,a_1)\Delta^4 \\ a_1 + 2a_2(a_0,a_1)\Delta + 3a_3(a_0,a_1)\Delta^2 + 4a_4(a_0,a_1)\Delta^3 \end{pmatrix}. \end{aligned} \tag{24}$$

Substituting on the right-hand-side for a_0 and a_1 by $f(\Delta)$ and $f'(\Delta)$, respectively, we get

$$\begin{aligned} f(\Delta) = {} & a_0 + a_1\Delta + 4a_2\Delta^2 + \left(5a_3 + 2a_2\frac{\partial a_2}{\partial a_1} + a_1\frac{\partial a_2}{\partial a_0}\right)\Delta^3 + \left(6a_4 + 3a_3\frac{\partial a_2}{\partial a_1} + 2a_2\frac{\partial a_3}{\partial a_1}\right. \\ & \left. + 2a_2^2\frac{\partial^2 a_2}{\partial a_1^2} + a_2\frac{\partial a_2}{\partial a_0} + a_1\frac{\partial a_3}{\partial a_0} + 2a_1a_2\frac{\partial^2 a_2}{\partial a_1\partial a_0} + \frac{1}{2}a_1^2\frac{\partial^2 a_2}{\partial a_0^2}\right)\Delta^4. \end{aligned} \tag{25}$$

The partial derivatives can not be calculated unless the functional forms of the recursion coefficients are known. One possibility is to specify the function being expanded, $f(x)$, as we did at the start of this proof. The other possibility is to exploit the differential equation that $f(x)$ is a solution for, namely, (19). Substituting the Taylor expansion of $f(\Delta)$, from (24) in (19) and expanding up to the fourth power in Δ, we obtain the following relations:

$$\begin{aligned} a_2 &= -\frac{F}{2}, \\ a_3 &= -\frac{a_1F'}{6}, \\ a_4 &= -\frac{a_2F'}{12} - \frac{a_1^2F''}{24}, \end{aligned} \tag{26}$$

which lead to

$$\begin{aligned} \frac{\partial a_2}{\partial a_0} &= -\frac{F'}{2}, \\ \frac{\partial a_2}{\partial a_1} &= -\frac{\Delta F'}{2}, \\ \frac{\partial^2 a_2}{\partial a_0^2} &= -\frac{F''}{2}, \\ \frac{\partial^2 a_2}{\partial a_1^2} &= -\frac{\Delta^2 F''}{2}, \end{aligned}$$

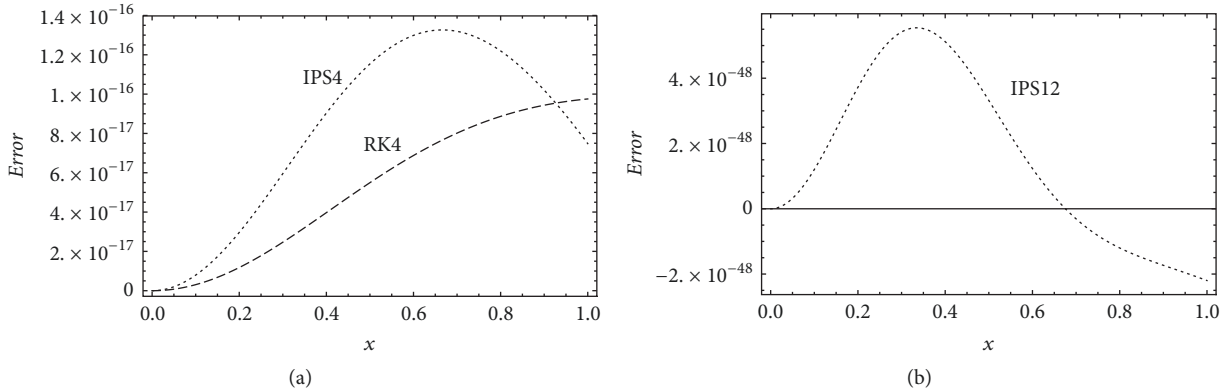

Figure 2: Error defined by the difference between the numerical and exact soliton solution of the NLSE, (29). (a) Dashed line corresponds to RK4 and dotted line corresponds to IPS4. (b) IPS12. Parameters used: $N = 1000$, number of digits $N_d = 50$, $f(0) = 1$, and $f'(0) = 0$.

$$\frac{\partial^2 a_2}{\partial a_0 \partial a_1} = -\frac{\Delta F''}{2},$$
$$\frac{\partial a_3}{\partial a_1} = -\frac{F'}{6}. \tag{27}$$

Using these equations in (28), we obtain

$$f(\Delta) = a_0 + a_1\Delta + 4a_2\Delta^2 + 8a_3(a_0, a_1)\Delta^3 + 16a_4(a_0, a_1)\Delta^4. \tag{28}$$

For $N = 2$, the last equation regenerates the Taylor series of $f(x)$, namely, (24) with $N = 1$, and this completes the proof. □

3. Application to Nonlinear Differential Equations

The method described in the previous section can be used as a powerful solver and integrator of nonlinear differential equations both analytically and numerically. In Section 3.1, we apply the method on a number of well-known problems. In Section 3.2, we show the power of the method in terms of detailed analysis of the error and CPU time.

3.1. Examples. For the sake of demonstration, we consider the following nonlinear differential equation:

$$\frac{1}{2}f(x) - \frac{1}{2}f''(x) - f^3(x) = 0. \tag{29}$$

The reason for selecting this equation is that $f(x) = \mathrm{sech}(x)$ is one of its exact solutions. Substituting the power series expansion $f(x) = \sum_{n=0}^{\infty} a_n x^n$, we obtain, as usual, the recursion relations

$$a_2(a_0) = \frac{1}{2!}\left[1 - 2(a_0)^2\right]a_0,$$
$$a_3(a_0, a_1) = \frac{1}{3!}\left[1 - 6(a_0)^2\right]a_1,$$
$$a_4(a_0) = \frac{1}{4!}\left[1 - 8a_0^2 + 12a_0^4 - 12a_1^2\right]a_0, \tag{30}$$

where a_0 and a_1 turn out to be independent parameters which in the present case correspond to the initial conditions on the solution and its first derivative. It is not surprising that these recursion relations are identical with those we found for the $\mathrm{sech}(x)$ in the previous section, (9). Therefore, substituting the above recursion relations in $f(x) = \sum_{n=0}^{\infty} a_n x^n$ we obtain the Taylor series expansion, T, of $f(x) = \mathrm{sech}(x)$. Removing the divergency in T follows exactly the same steps as in the previous section, and thus an exact solution in terms of a convergent power series is obtained, as also plotted in Figure 1.

For $f(x) = 1/(1 + x^2)$, the relevant differential equation is

$$f''(x) + f^3(x) - 6f^2(x) = 0. \tag{31}$$

Substituting the power series expansion in this equation, the recursion relations will be given by (16)–(18). Similarly, the convergent series solution will be obtained, as in the previous section.

3.2. Numerical Method. As a numerical method, the power series is very powerful and efficient [10]. The power series method with $N_{\max} = 4$, denoted by IPS4, is used to solve the NLSE, (29) and the error is calculated as the difference between the numerical solution and the exact solution, namely, $\mathrm{sech}(x)$. The equation is then resolved using the fourth-order Runge-Kutta (RK4) method. In Figure 2, we plot the error of both methods which turn out to be of the same order. Using the iterative power series method with $N_{\max} = 12$, (IPS12), the error drops to infinitesimally low values. Neither the CPU time nor the memory requirements for IPS12 are much larger than those for IPS4; it is straight forward upgrade to higher orders which leads to ultrahigh efficiency. This is verified by the set of tables, Tables 1–4, where we compute $\mathrm{sech}(1)$ using both the RK4 and the iterative power series method and show the corresponding CPU time. For the same N, Tables 1 and 3 show that both RK4 and

Table 1: RK4 sech(x) solution of the NLSE, (29), computed at $x = 1$.

	RK4	
N	sech(1)	CPU time
1000	0.648054273663946375368359898768277544096139599301 5	0.087257
2000	0.648054273663889210246100221264500124274840065198 3	0.166534
3000	0.648054273663886152279172931038209808629107170775 2	0.229016
4000	0.648054273663885637731963076388630151209976316175 0	0.314480
5000	0.648054273663885497123260022116052414392932141142 6	0.427610
Exact	0.648054273663885399574977353226150323108489312071 9	

Table 2: IPS4 sech(x) solution of the NLSE, (29), computed at $x = 1$.

	IPS4	
N	sech(1)	CPU time
1000	0.648054273663932395523335036778670040071525554837 3	0.056793
2000	0.648054273663888327749280922338930859615560227688 4	0.137941
3000	0.648054273663885977382311996445974018504602460630 2	0.215395
4000	0.648054273663885582302302305056575982947535628028 1	0.304501
5000	0.648054273663885474396857902660464127013059522618 6	0.372284
Exact	0.648054273663885399574977353226150323108489312071 9	

Table 3: IPS12 sech(x) solution of the NLSE, (29), computed at $x = 1$.

	IPS12	
N	sech(1)	CPU time
1000	0.648054273663885399574977353226150323107959435407 9	0.280617
2000	0.648054273663885399574977353226150323108489181636 1	0.595639
3000	0.648054273663885399574977353226150323108489311063 4	0.891370
4000	0.648054273663885399574977353226150323108489312040 0	1.080357
5000	0.648054273663885399574977353226150323108489312069 7	1.366386
Exact	0.648054273663885399574977353226150323108489312071 9	

Table 4: IPS12 sech(x) solution of the NLSE, (29), computed at $x = 1$, but with much less number of iterations than in Table 3.

	IPS12	
N	sech(1)	CPU time
3	0.648054279407966562946911415434898005581408843095 3	0.000782
6	0.648054273664377034628396977958780764625605810013 5	0.001429
9	0.648054273663887200745285656707469792278748959023 8	0.002123
12	0.648054273663885425979357301574795403045193256502 7	0.002768
15	0.648054273663885400149502325770247990974136901658 9	0.003594
Exact	0.648054273663885399574977353226150323108489312071 9	

IPS4 produce the first 16 digits of the exact value (underlined numbers in the last raw) and consume almost the same CPU time. Table 3 shows that, for the same N, IPS12, reproduces the first 49 digits of the exact value. The CPU time needed for such ultrahigh accuracy is just about 3 times that of the RK4 and IPS4. Of course the accuracy can be arbitrarily increased by increasing N or more efficiently $N_{\max}$. For IPS12 to produce only the first 16 digits, as in RK4 and IPS4, only very small number of iterations is needed, as shown in Table 4. The CPU time in this case is about 100 times less than that of RK4 and IPS4, highlighting the high efficiency of the power series method.

A more challenging test on the power series method is the chaotic Lorenz system [25] given by

$$\begin{aligned} \dot{z}_1 &= -\sigma z_1 + \sigma z_2, \\ \dot{z}_2 &= -z_1 z_3 + r z_1 - z_2, \\ \dot{z}_3 &= z_1 z_2 - b z_3, \end{aligned} \tag{32}$$

where we take the usual values $\sigma = 10$, $b = -8/3$, and $R = 28$ with initial conditions $z_1(0) = z_2(0) = 1$ and $z_3(0) = 20$. It is straight forward to generalise the method to three differential equations; therefore we do not show the

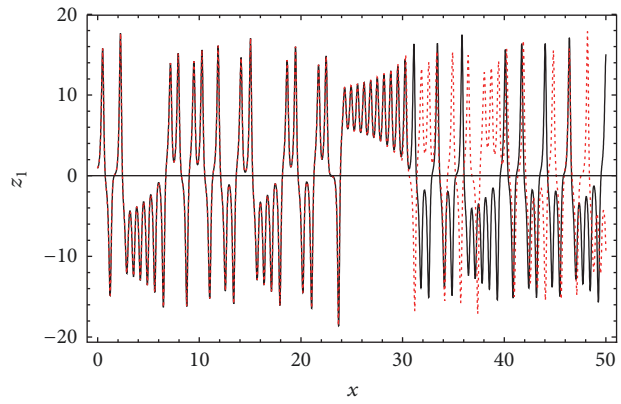

FIGURE 3: Numerical solution of the chaotic Lorenz system, (32), using the RK4 and IPS methods. Parameters: for IPS (solid black line), $N_{\max} = 16$, $N = 5000$. For RK4 (dashed red line), $N = 2 \times 10^5$. For both curves we used $N_d = 1000$, $z_1 = 1 = z_2 = 1$, $z_3 = 20$, $\sigma = 10$, $b = -8/3$, and $R = 28$.

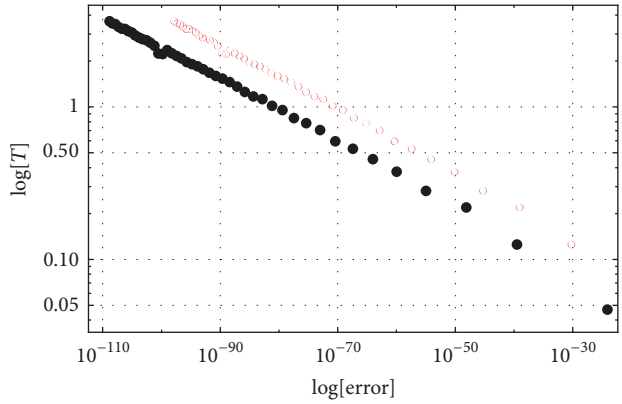

FIGURE 4: CPU time versus error on a log-log scale for the iterative power series solution of the NLSE, (29). Black filled circles correspond to error calculated from the difference between the power series solution and the exact solution. Red empty circles correspond to the error estimated by the formula (15). Parameters used: $N_{\max} = 50$, $n_d = 500$, and N ranges between 2 and 90.

details of the calculation. In Figure 3, the results of solving the Lorenz system using RK4 and IPS12 are shown. For the same parameters, namely discretization, RK4 reaches stability at about $x = 30$; that is, the curve for $x < 30$ is unchanged by increasing N, but for $x > 30$, the curve keeps changing by increasing N. In comparison, IPS12 reaches stability at about $x = 50$. In chaotic systems, it is quite challenging to go that deep in the chaotic region. Hence, the need for such high accuracy methods.

Achieving higher accuracy requires larger CPU time usage. Therefore, it is important to investigate how the CPU time, denoted here by T, depends on the main parameters of the method, namely N and $N_{\max}$. A typical plot is shown in Figure 4, where we plot on a log-log scale the CPU time versus the error. The linear relationship indicates $T \propto \text{error}^p$, where p is the slope of the line joining the points in Figure 4. The error can be calculated in two ways: (i) the difference between the numerical solution and (ii) theoretical estimate, (15). Both ways are shown in the figure and they have the same slope. However, as expected, error defined by (15), which is actually an upper limit on the error, is always larger than the first one. To find how the CPU time depends explicitly on N and $N_{\max}$, we argue that the dependence should be of the form $T \propto NN_{\max}^3$. This is justified by the fact that CPU time should be linearly proportional to the number of terms computed. The number of terms computed increases linearly with the number of iterations N. The number of terms in the power series is linearly proportional to $N_{\max}$. When substituted in the NLSE with cubic nonlinearity, the resulting number of terms, and thus T, will be proportional to $N_{\max}^3$. In Figure 5, it is shown indeed that the ratio $T/NN_{\max}^3$ saturates asymptotically to a constant for large N and $N_{\max}$ since the scaling behaviors mentioned here apply for large N and $N_{\max}$. The proportionality constant, c, is very small and corresponds to the CPU time of calculating one term. It is dependent on the machine, the programming optimization [10], and the number of digits used, N_d. In terms of the number of digits, the CPU time increases, as shown in Figure 6, where it is noticed that CPU time is almost constant for number of digits $N_d < 500$.

4. Conclusions

We have presented an iterative power series method that solves the problem of finite radius of convergence. We have

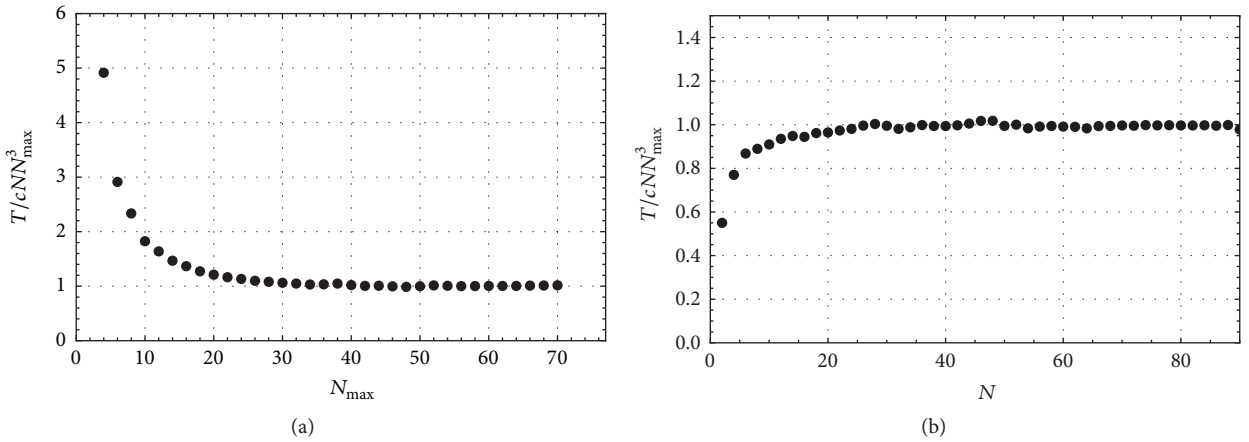

FIGURE 5: CPU time versus $N_{\max}$ (a) and N (b) for the NLSE, (29). Parameters used: for (a), $N = 400$. For (b), $N_{\max} = 70$. For both plots, $N_d = 1000$, $c = 6.2 \times 10^{-7}$, $f(0) = 1$, $f'(0) = 0$, and $\Delta = 1/N$.

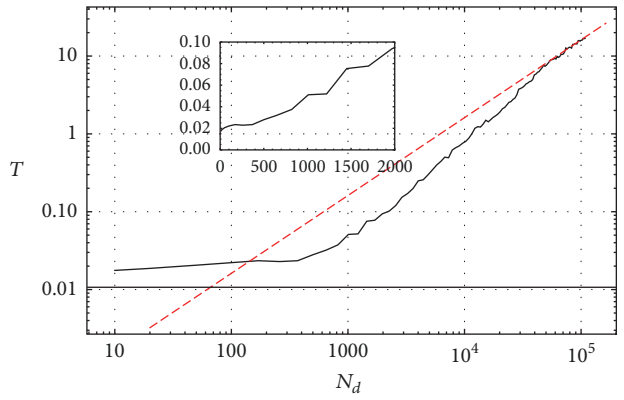

FIGURE 6: CPU time versus the number of digits for the power series solution of NLSE, (29). Parameters used: $N_{\max} = 4$, $N = 400$, $f(0) = 1$, $f'(0) = 0$, and $\Delta = 1/N$. Inset shows a zoom on linear scale. Red dashed line is an asymptote of slope equal 1.

proved that the iterative power series is always composed of a sum of the typical power series of the function and a complementary series that cancels the divergency. The method is divided into two schemes where in the first we find a convergent power series for a given function and in the second we solve a given nonlinear differential equation. The result of the iterative power series expansion of sech(x) is remarkably convergent for arbitrary radius of convergence and accuracy, as shown by Figures 1 and 2 and Tables 1–4. Extremely high accuracy can be obtained by using higher-order iterative power series via increasing $N_{\max}$ with relatively low CPU time usage. Robustness and efficiency of the method have been shown by solving the chaotic Lorenz system and the NLSE. Extensive analysis of the error and CPU time characterising the method is performed. Although we have focused on the localised sech(x) solution of the NLSE, all other solitary wave solutions (conoidal waves) can be obtained using the present method, just by choosing the appropriate initial conditions.

The method can be generalised to partial and fractional differential equations making its domain of applicability even wider.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, vol. 7, Dover Publications, New York, NY, USA, 1965.

[2] R. E. Scraton, "A note on the summation of divergent power series," *Mathematical Proc. of the Camb. Phil. Soc*, vol. 66, pp. 109–114, 1969.

[3] A. D. Polyanin and V. F. Zaitsev, *Handbook of Nonlinear Partial Differential Equations*, Chapman and Hall, New York, NY, USA, 2003.

[4] C. S. Gardner, J. M. Greene, M. D. Kruskal, and R. M. Miura, "Method for solving the Korteweg-deVries equation," *Physical Review Letters*, vol. 19, no. 19, pp. 1095–1097, 1967.

[5] P. D. Lax, "Integrals of nonlinear equations of evolution and solitary waves," *Communications on Pure and Applied Mathematics*, vol. 21, pp. 467–490, 1968.

[6] V. B. Matveev and M. A. Salle, *Darboux Transformations and Solitons*, Springer-Verlag, Berlin, Gemany, 1991.

[7] R. Hirota, *Topics in Current Physics 17*, R. K. Eullough and P. I. Caudrey, Eds., Springer-Verlag, Berlin, Germany, 1980.

[8] G. Adomian, *Solving Frontier Problems of Physics: The Decomposition Method*, Kluwer Academic, Dordrecht, The Netherlands, 1994.

[9] S. Liao and Y. Tan, "A general approach to obtain series solutions of nonlinear differential equations," *Studies in Applied Mathematics*, vol. 119, no. 4, pp. 297–354, 2007.

[10] R. Barrio, M. Rodríguez, A. Abad, and F. Blesa, "Breaking the limits: the Taylor series method," *Applied Mathematics and Computation*, vol. 217, no. 20, pp. 7940–7954, 2011.

[11] G. Corliss and Y. F. Chang, "Solving ordinary differential equations using Taylor series," *ACM Transactions on Mathematical Software*, vol. 8, no. 2, pp. 114–144, 1982.

[12] Y. F. Chang and G. Corliss, "ATOMFT: solving ODEs and DAEs using Taylor series," *Computers & Mathematics with Applications*, vol. 28, no. 10-12, pp. 209–233, 1994.

[13] J. D. Pryce, "Solving high-index DAEs by Taylor series," *Numerical Algorithms*, vol. 19, no. 1-4, pp. 195–211, 1998.

[14] R. Barrio, "Performance of the Taylor series method for ODEs/DAEs," *Applied Mathematics and Computation*, vol. 163, no. 2, pp. 525–545, 2005.

[15] N. S. Nedialkov and J. D. Pryce, "Solving differential-algebraic equations by taylor series (I): Computing taylor coefficients," *BIT Numerical Mathematics*, vol. 45, no. 3, pp. 561–591, 2005.

[16] N. S. Nedialkov and J. D. Pryce, "Solving differential-algebraic equations by Taylor series. (II): Computing the system Jacobian," *BIT Numerical Mathematics*, vol. 47, no. 1, pp. 121–135, 2007.

[17] N. S. Nedialkov and J. D. Pryce, "Solving differential algebraic equations by Taylor series. (III): THE DAETS code," *JNAIAM. Journal of Numerical Analysis, Industrial and Applied Mathematics*, vol. 3, no. 1-2, pp. 61–80, 2008.

[18] N. S. Nedialkov, K. R. Jackson, and G. F. Corliss, "Validated solutions of initial value problems for ordinary differential equations," *Applied Mathematics and Computation*, vol. 105, no. 1, pp. 21–68, 1999.

[19] W. Tucker, "A Rigorous ODE Solver and Smale's 14th Problem," *Foundations of Computational Mathematics*, vol. 2, no. 1, pp. 53–117, 2002.

[20] R. Barrio and F. Blesa, "Systematic search of symmetric periodic orbits in 2DOF Hamiltonian systems," *Chaos, Solitons & Fractals*, vol. 41, no. 2, pp. 560–582, 2009.

[21] R. Barrio, F. Blesa, and S. Serrano, "Periodic, escape and chaotic orbits in the Copenhagen and the (n+1)-body ring problems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 5, pp. 2229–2238, 2009.

[22] R. Barrio, "Painting chaos: a gallery of sensitivity plots of classical problems," *International Journal of Bifurcation and Chaos*, vol. 16, no. 10, pp. 2777–2798, 2006.

[23] R. Barrio and S. Serrano, "A three-parametric study of the Lorenz model," *Physica D: Nonlinear Phenomena*, vol. 229, no. 1, pp. 43–51, 2007.

[24] Á. Jorba and M. Zou, "A software package for the numerical integration of ODEs by means of high-order Taylor methods," *Experimental Mathematics*, vol. 14, no. 1, pp. 99–117, 2005.

[25] E. Lorenz, "Deterministic nonperiodic flow," *Journal of Atomic and Molecular Sciences*, vol. 20, pp. 130–141, 1963.

[26] F. M. Allan, "Derivation of the Adomian decomposition method using the homotopy analysis method," *Applied Mathematics and Computation*, vol. 190, no. 1, pp. 6–14, 2007.

[27] Q. M. Al-Mdallal, M. I. Syam, and P. . Ariel, "Extended homotopy perturbation method and the axisymmetric flow past a porous stretching sheet," *International Journal for Numerical Methods in Fluids*, vol. 69, no. 5, pp. 909–925, 2012.

[28] M. A. Hajji and K. Al-Khaled, "Analytic studies and numerical simulations of the generalized Boussinesq equation," *Applied Mathematics and Computation*, vol. 191, no. 2, pp. 320–333, 2007.

[29] S. Liao, "On the homotopy analysis method for nonlinear problems," *Applied Mathematics and Computation*, vol. 147, no. 2, pp. 499–513, 2004.