mathematics

MDPI

Article

Exploring a Novel Multi-Stage Differential Transform Method Coupled with Adomian Polynomials for Solving Implicit Nonlinear ODEs with Analytical Solutions

Brahim Benhammouda [1,2,*] and Hector Vazquez-Leal [3]

[1] Academic Wing, Khalifa Bin Zayed Air College, Al Ain 1380, United Arab Emirates
[2] Department of Mathematics, Cert, Abu Dhabi Women's Campus, Higher Colleges of Technology, Abu Dhabi 25026, United Arab Emirates
[3] Facultad de Instrumentación Electrónica, Universidad Veracruzana, Circuito Gonzalo Aguirre Beltrán S/N, Xalapa 91000, Veracruz, Mexico; hvazquez@uv.mx
* Correspondence: bbenhammouda@hct.ac.ae

Abstract: In engineering, physics, and other fields, implicit ordinary differential equations are essential to simulate complex systems. However, because of their intrinsic nonlinearity and difficulty separating higher-order derivatives, implicit ordinary differential equations pose substantial challenges. When applied to these types of equations, traditional numerical methods frequently have problems with convergence or require a significant amount of computing power. In this work, we present the multi-stage differential transform method, a novel semi-analytical approach for effectively solving first- and second-order implicit ordinary differential systems, in conjunction with Adomian polynomials. The main contribution of this method is that it simplifies the solution procedure and lowers processing costs by enabling the differential transform method to be applied directly to implicit systems without transforming them into explicit or quasi-linear forms. We obtain straightforward and effective algorithms that build solutions incrementally utilizing the characteristics of Adomian polynomials, providing benefits in theory and practice. By solving several implicit ODE systems that are difficult for traditional software programs such as Maple 2024, Mathematica 14, or Matlab 24.1, we validate our approach. The multi-stage differential transform method's contribution includes expanded convergence intervals for numerical results, more accurate approximate solutions for wider domains, and the efficient calculation of exact solutions as a convergent power series. Because of its ease of implementation in educational computational tools and substantial advantages in terms of simplicity and efficiency, our method is suitable for researchers and practitioners working with complex implicit differential equations.

Keywords: implicit systems of nonlinear ordinary differential; differential transform method; Runge–Kutta method; Adomian polynomials

MSC: 65P99

check for updates

Academic Editors: Quanxin Zhu, Andrei D. Polyanin and Patricia J. Y. Wong

Received: 5 December 2024
Revised: 10 January 2025
Accepted: 18 January 2025
Published: 23 January 2025

Citation: Benhammouda, B.; Vazquez-Leal, H. Exploring a Novel Multi-Stage Differential Transform Method Coupled with Adomian Polynomials for Solving Implicit Nonlinear ODEs with Analytical Solutions. *Mathematics* **2025**, *13*, 358. https://doi.org/10.3390/math13030358

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Differential equations are essential for modeling a broad range of phenomena in physics, engineering, biology, and economics, among other disciplines. To forecast future behavior based on existing conditions, these equations offer a mathematical tool for characterizing dynamical systems and processes. For explicit ODE systems, numerous

solution methods are available, including the Runge–Kutta method [1,2], the power series method [3], the Adomian decomposition method [4–7], the homotopy perturbation method [8], and the differential transform method (DTM) [9–11]. In addition, many software packages are capable of solving these explicit systems. Because of their capacity to represent complex systems in which the relationship between variables is not explicitly defined, implicit ordinary differential equations (IODEs) have attracted much interest among the various kinds of differential equations. In these systems, the higher derivative of the unknown cannot, in general, be easily isolated due to nonlinearities. Although these systems can offer greater flexibility in modeling complex problems, they can be more challenging to solve and require more advanced numerical methods. Packages such as Mathematica 14, Matlab 24.1, and Maple 2024 are not able to solve these systems in many situations, and examples of such cases are provided. Furthermore, applying implicit numerical integration methods to such systems will require the solution of nonlinear algebraic systems at every step. This will involve a large computational effort, especially for large-dimension systems.

Implicit differential equations form a major topic of study in the science of dynamical systems since they emerge in engineering situations. IODEs involve several major difficulties compared to explicit ODEs. First, the existence and uniqueness of solutions can be challenging. Second, the numerical integration of IODEs is more complex; conventional approaches sometimes depend on converting these equations into explicit forms, which can be computationally expensive. In addition, IODEs often have nonlinearities that could cause numerical method convergence problems, therefore complicating the process of finding precise responses. These elements call for the development of specific methods for successfully treating IODEs in different uses [12].

Furthermore, PDEs and ODEs occur in many technological and scientific domains. Even though many of these equations are PDEs, they are frequently converted to ODEs using traveling waves or other standard reductions [13]. Some nonlinearities in the PDEs lead to implicit ordinary differential equations.

Forces that depend on accelerations produce IODEs in mechanical systems, especially those involving dry friction, which are extremely complex. Because of these dependencies, implicit differential equations cannot be solved for higher derivatives, creating significant mathematical difficulties about the existence and uniqueness of solutions. Dry friction in sliding and rolling mechanical systems can behave non-deterministically, producing multiple solutions with the same starting conditions—a phenomenon associated with the Painlevé paradoxes. Furthermore, dry friction can appear as viscous resistance proportional to the square of velocity due to implicit equations in these systems, which can also produce stagnation zones where motion stops. Since implicit equations have considerable effects on system dynamics and might produce unexpected behaviors, they must be carefully considered while studying mechanical systems. To effectively forecast and control the motion of complex mechanical systems, a deep comprehension of these equations is essential [14].

Through index analysis and computational differentiation, differential algebraic equations (DAEs) can frequently be converted into equivalent IODEs under specific circumstances, which is particularly interesting given that DAEs have applications in fields such as control theory, fluid dynamics, circuit analysis, and mechanical and chemical engineering [12,15–17]. Among these diverse engineering applications, the field of electrical engineering provides particularly compelling examples of IODEs. The analysis of nonlinear RLC circuits is one of the most important uses of these equations since it offers a framework for comprehending the complex dynamics that result from the interaction of resistance, inductance, and capacitance. In these circuits, the system's behavior is frequently characterized by equations that are not explicitly solvable for the higher derivatives, hence implicit formulations are required [16].

Differential systems of the type $\mathbf{F}\left(\mathbf{x}^{(m)}(t), \ldots, \mathbf{x}'(t), \mathbf{x}(t), t\right) = \mathbf{0}$, where the Jacobian of $\mathbf{F}$ with respect to the mth derivative $\mathbf{x}^{(m)}$ is nonsingular, are known as implicit ordinary differential systems. These equations require more complex methods for analysis and solution than explicit ones, where the mth derivative can be isolated. Implicit relationships complicate the solution procedure because they may not be immediately applicable to standard methods [16,17]. Consequently, several numerical methods for finding approximate solutions have been developed through research on IODEs.

By parameterizing IODEs, researchers can describe the equations in a way that shows the connections among variables without having to rewrite the system in an explicit form. Through this parametrization, researchers can obtain general solutions by converting implicit ODEs into standard systems of ordinary differential equations. These results can then be used to solve related differential algebraic problems [18]. While this parametrization approach provides a theoretical framework, the practical implementation of solutions requires specific numerical methods. A variety of techniques have been proposed to solve explicit ODEs. Runge–Kutta methods, for example, are frequently employed due to their efficiency in resolving ordinary differential equations. In contrast, they encounter difficulties with implicit equations as a result of the need for iterative solutions, which can result in intensified computational costs and convergence issues [13,19]. Differential equations can be transformed into algebraic equations using finite difference methods. Despite their effectiveness, these techniques may be computationally costly since they need exact discretization and have stability problems [13,19]. In addition, Picard iteration and incremental harmonic balance methods have been implemented; nevertheless, they can fail and commonly face convergence issues [13,19].

From theoretical convergence frameworks to useful numerical techniques and visualizations, the study of implicit ODEs uses a variety of approaches. To guarantee the presence of unique solutions, for instance, consecutive approximations in Banach spaces offer an organized method using operator techniques [20]. By using the Vessiot distribution, the geometric theory reduces implicit ODEs to ordinary dynamical systems analysis by transforming them into vector field distributions on manifolds [21]. Techniques that emphasize convergence conditions, such as the backward Euler method, are used to numerically solve initial and boundary value problems [22]. The Runge–Kutta and multi-step methods for implicit equations are also modified to maintain the same level of accuracy as explicit ODEs [23]. The mathematical basis of quadrature methods is further expanded by introducing an existence theorem and providing examples [24].

Given the importance of IODEs and the aforementioned numerical issues, we introduce a new semi-analytical method called the multi-stage differential transform method (MsDTM), coupled with Adomian polynomials. This approach presents a systematic means of obtaining implicit ODE solutions by constructing solutions incrementally. The following summarizes the MsDTM. The MsDTM subdivides the solution interval into a set number of sub-intervals for its operation. In the first sub-interval, the initial condition enables the first application of the DTM to the differential equation, producing the solution for this interval. Next, the algorithm processes the second sub-interval. Again, the DTM applies to the differential equation using the initial condition now derived from the first sub-interval's solution. This step ensures continuity and consistency of the solution between sub-intervals. For each subsequent sub-interval, the same procedure is repeated. For each case, the prior sub-interval's solution updates the next sub-interval's initial condition. As we follow the algorithm, this iterative process yields a progressive solution. When the algorithm reaches the final step, it finds a solution for the whole interval. Finally, each sub-interval's contribution ensures the method accurately approximates the solution across the entire domain. Readers can find more details in references [25–27]. The ease of use of

MsDTM in computational tools such as Maple, Matlab, or Mathematica is one of the main advantages since it allows access to both scholars and practitioners. The MsDTM, combined with Adomian polynomials, seeks to provide precise solutions to IODEs by utilizing the advantages of both the differential transform method and Adomian polynomials.

Then, we propose a novel and efficient technique to solve implicit mth order ($m = 1,\ m = 2$) systems of ODEs of the form:

$$\mathbf{F}\left(\mathbf{x}^{(m)}(t), \ldots, \mathbf{x}'(t), \mathbf{x}(t), t\right) = \mathbf{0},\ t \in [0, T], \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the sought solution and $\mathbf{x}^{(m)}(t)$ denotes its mth derivative and $\mathbf{x} : (\mathbb{R}^n)^{(m+1)} \times [0, T] \longrightarrow \mathbb{R}^n$. This system is subject to the following initial conditions:

$$\mathbf{x}^{(i)}(0) = \mathbf{x}_i,\ 0 \leq i \leq m-1, \tag{2}$$

where $\mathbf{x}_i \in \mathbb{R}^n$ are given vectors. Throughout this paper, we assume the function $\mathbf{F}(\mathbf{y}_m, \ldots, \mathbf{y}_1, \mathbf{y}_0, t)$ to be analytical with respect to its arguments, and the Jacobian $J := \mathbf{F}'_{\mathbf{y}_m}(\mathbf{y}_m, \ldots, \mathbf{y}_1, \mathbf{y}_0, t) \in \mathbb{R}^{n \times n}$ to be nonsingular. Furthermore, the present work focuses on the numerical method for solving IODEs. Due to the complexity of the IODEs discussed in this paper, we assume the solutions of these systems to be analytical. Some sub-classes of (1) are of the following explicit form:

$$\mathbf{x}^{(m)}(t) = \mathbf{G}\left(\mathbf{x}^{(m-1)}(t), \ldots, \mathbf{x}'(t), \mathbf{x}(t), t\right),\ t \in [0, T], \tag{3}$$

and of the quasi-linear implicit form:

$$\mathbf{A}\left(\mathbf{x}^{(m-1)}(t), \ldots, \mathbf{x}'(t), \mathbf{x}(t), t\right)\mathbf{x}^{(m)}(t) = \mathbf{G}\left(\mathbf{x}^{(m-1)}(t), \ldots, \mathbf{x}'(t), \mathbf{x}(t), t\right),\ t \in [0, T], \tag{4}$$

where $\mathbf{A} := \mathbf{A}\left(\mathbf{x}^{(m-1)}(t), \ldots, \mathbf{x}'(t), \mathbf{x}(t), t\right) \in \mathbb{R}^{n \times n}$ is a nonsingular matrix.

The technique we propose combines the DTM with Adomian polynomials [28–33] to create a powerful solution method. The key idea is to apply the DTM directly to the system (1) without converting it to the semi-explicit form (3) or the quasi-linear form (4). By leveraging an important property of Adomian polynomials, we have derived a simple and efficient algorithm for this new technique, significantly enhancing its simplicity and effectiveness.

Our approach can yield the exact solution as a convergent power series, provided that all computations are performed with exact precision. To extend the convergence interval of the DTM solution, we have developed a multi-stage DTM algorithm (MsDTM). Both, the DTM and the MsDTM algorithms can be easily implemented using software packages such as Maple, Mathematica, or Matlab.

To demonstrate the effectiveness and accuracy of our technique, we solved several implicit systems of ODEs using the new method. It is important to note that all the numerical examples presented in this paper are not solvable by Maple 2024, numerically and symbolically. Furthermore, the Runge–Kutta method is also inapplicable to these examples. The numerical results show that our technique has successfully solved all given numerical examples. Specifically, the standard DTM provides the exact solution as a convergent power series, while the MsDTM produces accurate approximate solutions over larger intervals. To the best of our knowledge, this paper is the first to explore the numerical solution of implicit ordinary differential systems.

This paper's main contribution is a proposed MsDTM that solves IODEs without needing transformation to explicit or quasi-linear forms, thus reducing computational costs. Adomian polynomials, when integrated, expand nonlinear functions into power series to solve difficult problems. The method, unlike traditional Runge–Kutta methods, handles first- and second-order implicit systems and generates accurate solutions expressed as convergent power series. With high accuracy, the multi-stage approach extends convergence intervals.

We structure the paper as follows: In Section 2, we review Adomian polynomials and the DTM. Section 3 presents two theorems that form the basis of the new algorithms for solving the implicit initial-value problems (1) and (2). In Section 4, we solve several first and second-order implicit systems of ODEs with different types of nonlinearities to illustrate the accuracy of the new method. In Section 5, we focus on the numerical procedure and examples. Finally, in Section 6, we present our concluding remarks.

2. Adomian Polynomials and the DTM

In this section, we give a brief review of Adomian polynomials [28–33] and the differential transform method. The differential transforms of the functions $\mathbf{x}(t)$ and $\mathbf{y}(t)$ are defined by the following:

$$x_k = \frac{1}{k!}\left[\frac{d^k\mathbf{x}(t)}{dt^k}\right]_{t=0}, \; y_k = \frac{1}{k!}\left[\frac{d^k\mathbf{y}(t)}{dt^k}\right]_{t=0}, \; k \geq 0, \tag{5}$$

and the inverse differential transforms of x_k, y_k, $k \geq 1$ are given by the following:

$$\mathbf{x}(t) = \sum_{k=0}^{\infty} x_k t^k, \; \mathbf{y}(t) = \sum_{k=0}^{\infty} y_k t^k. \tag{6}$$

From (5) and (6), we have the following:

$$\mathbf{x}(t) = \sum_{k=0}^{\infty} \frac{1}{k!}\left[\frac{d^k\mathbf{x}(t)}{dt^k}\right]_{t=0} t^k, \; \mathbf{y}(t) = \sum_{k=0}^{\infty} \frac{1}{k!}\left[\frac{d^k\mathbf{y}(t)}{dt^k}\right]_{t=0} t^k. \tag{7}$$

An approximate solution is given by the following:

$$\mathbf{x}(t) = \sum_{k=0}^{K} x_k t^k, \; \mathbf{y}(t) = \sum_{k=0}^{K} y_k t^k, \tag{8}$$

where K is the number of terms in the approximation. The accuracy of the solution should improve as we include more terms in this expansion.

If $\mathbf{x}(t)$ and $\mathbf{y}(t)$ are expanded as in (6), then the nonlinear function $G(\mathbf{x}, \mathbf{y})$ can be expanded using the Adomian polynomials as follows:

$$G\Big(\sum_{k=0}^{\infty} x_k t^k, \sum_{k=0}^{\infty} y_k t^k\Big) = \sum_{k=0}^{\infty} G_k(x_0, x_1, \ldots, x_k; y_0, y_1, \ldots, y_k) t^k, \tag{9}$$

where x_k and y_k, $k \geq 1$ denote the coefficients of expansion (6) and G_k denotes the Adomian polynomials, as follows:

$$\begin{aligned}
G_0 &= G(x_0, y_0),\\
G_1 &= x_1 G^{(1,0)} + y_1 G^{(0,1)},\\
G_2 &= x_2 G^{(1,0)} + y_2 G^{(0,1)} + \frac{x_1^2}{2} G^{(2,0)} + \frac{y_1^2}{2} G^{(0,2)} + x_1 y_1 G^{(1,1)},\\
G_3 &= x_3 G^{(1,0)} + y_3 G^{(0,1)} + x_1 x_2 G^{(2,0)} + y_1 y_2 G^{(0,2)} + (x_1 y_2 + x_2 y_1) G^{(1,1)}\\
&\quad + \frac{x_1^3}{6} G^{(3,0)} + \frac{y_1^3}{6} G^{(0,3)} + \frac{x_1^2 y_1}{2} G^{(2,1)} + \frac{x_1 y_1^2}{2} G^{(1,2)},\\
G_4 &= x_4 G^{(1,0)} + y_4 G^{(0,1)} + \left(x_1 x_3 + \frac{x_2^2}{2}\right) G^{(2,0)} + \left(y_1 y_3 + \frac{y_2^2}{2}\right) G^{(0,2)}\\
&\quad + (x_1 y_3 + x_2 y_2 + x_3 y_1) G^{(1,1)} + \frac{x_1^2 x_2}{2} G^{(3,0)} + \frac{y_1^2 y_2}{2} G^{(0,3)} + \left(x_1 x_2 y_1 + \frac{x_1^2 y_2}{2}\right) G^{(2,1)}\\
&\quad + \left(x_1 y_1 y_2 + \frac{x_2 y_1^2}{2}\right) G^{(1,2)} + \frac{x_1^4}{4!} G^{(4,0)} + \frac{x_1^3 y_1}{6} G^{(3,1)} + \frac{x_1^2 y_1^2}{4} G^{(2,2)}\\
&\quad + \frac{x_1 y_1^3}{4} G^{(1,3)} + \frac{y_1^4}{4!} G^{(0,4)},
\end{aligned}$$

where $G^{(k,l)} := \dfrac{\partial^{k+l} G(x_0, y_0)}{\partial x_0^k \partial y_0^l}$, $k \geq 0$, $l \geq 0$.

One can calculate any other Adomian polynomial from the following formula:

$$G_k := G_k(x_0, x_1, \ldots, x_k; y_0, y_1, \ldots, y_k) = \frac{1}{k!} \frac{d^k}{d\xi^k} \left[G\left(\sum_{i=0}^{\infty} \xi^i x_i, \sum_{i=0}^{\infty} \xi^i y_i \right) \right]_{\xi=0}, \; k \geq 0. \tag{10}$$

One important property of the Adomian polynomials, which we will exploit to develop our algorithms, is the fact that the Adomian polynomials G_k, $k \geq 1$ are affine with respect to x_k and y_k.

3. The New Algorithm

The proposed algorithm effectively integrates the differential transform method (DTM) with Adomian polynomials [28–33]. The primary concept is to apply the DTM directly to the system defined in class (1), where the nonlinear vector function is expanded into a power series using Adomian polynomials [34]. By leveraging the fact that Adomian polynomials F_k, $k \geq 1$ are affine with respect to x_k and y_k, we derive a linear algebraic recursion system for the differential transforms of the solution. A significant benefit of this technique is that it avoids the need to convert system (1) into forms (3) or (4) before applying the DTM, greatly simplifying the algorithm. This approach is supported by two new theorems, which apply to the general first and second-order nonlinear systems given by (1).

Theorem 1. *Consider the implicit nonlinear ordinary differential system* $\mathbf{F}(\mathbf{x}', \mathbf{x}, t) = \mathbf{0}$, *with* $\mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^n$, $\mathbf{F} \in \mathbb{R}^n \times \mathbb{R}^n \times [0, T]$, *where the Jacobian* $J = \mathbf{F}_{\mathbf{x}'}(\mathbf{x}', \mathbf{x}, t)$ *is non-singular. Assume that the function* $\mathbf{F}$ *is analytical with respect to its arguments and let* $\mathbf{x}(t) = \sum_{k=0}^{\infty} x_k t^k$. *Let* $F_k = F_k(x_1, \ldots, kx_k, (k+1)x_{k+1}; x_1, \ldots, x_k)$, $k \geq 1$ *be the vector of k-th Adomian polynomials of the components of the vector function* $\mathbf{F}$. *Then, the differential transform of the solution* $\mathbf{x}(t)$ *is given by the linear algebraic system* $kJx_k = -r_{k-1}$, *where*

$r_{k-1} = F_{k-1}(x_1, \ldots, (k-1)x_{k-1}, 0; x_1, \ldots, x_{k-1})$, $k \geq 2$, *with the first recursion* x_1 *being computed from* $\mathbf{F}(x_1, x_0, 0) = \mathbf{0}$.

Proof. Assume that the solution $\mathbf{x}(t)$ can be expanded as follows:

$$\mathbf{x}(t) = \sum_{k=0}^{\infty} x_k t^k, \tag{11}$$

where x_k is the differential transform of the solution $\mathbf{x}(t)$. Then, we expand the left side of the equation $\mathbf{F}(\mathbf{x}'(t), \mathbf{x}(t), t) = \mathbf{0}$ in terms of the Adomian polynomials to obtain the following:

$$\mathbf{F}\left(\sum_{k=0}^{\infty}(k+1)x_{k+1}t^k, \sum_{k=0}^{\infty} x_k t^k, t\right) = \sum_{k=0}^{\infty} F_k t^k = 0. \tag{12}$$

This gives the following:

$$F_k(x_1, \ldots, kx_k, (k+1)x_{k+1}; x_1, \ldots, x_k) = \mathbf{0},\ k \geq 1, \tag{13}$$

or

$$(k+1)Jx_{k+1} + F_k(x_1, \ldots, kx_k, 0; x_1, \ldots, x_k),\ k \geq 1, \tag{14}$$

which leads to the following recursion system for computing the differential transform of the derivative $\mathbf{x}(t)$

$$kJx_k = -F_{k-1}(x_1, \ldots, (k-1)x_{k-1}, 0; x_1, \ldots, x_{k-1}),\ k \geq 2. \tag{15}$$

□

Theorem 2. *Consider the implicit nonlinear ordinary differential system* $\mathbf{F}(\mathbf{x}'', \mathbf{x}', \mathbf{x}, t) = \mathbf{0}$, *with* $\mathbf{x}(0) = \mathbf{x}_0$, $\mathbf{x}'(0) = \mathbf{x}_1 \in \mathbb{R}^n$, $\mathbf{F} \in (\mathbb{R}^n)^3 \times [0, T]$, *where the Jacobian* $J = \mathbf{F}_{\mathbf{x}''}(\mathbf{x}'', \mathbf{x}', \mathbf{x}, t)$ *is nonsingular. Assume that the function* $\mathbf{F}$ *is analytical with respect to its arguments and let* $\mathbf{x}(t) = \sum_{k=0}^{\infty} x_k t^k$. *Then, the differential transform of the solution* $\mathbf{x}(t)$ *is given by the linear algebraic system, as follows:* $(k+1)(k+2)Jx_{k+2} = -F_k(2x_2, \ldots, k(k+1)x_{k+1}, 0; 2x_2, \ldots, (k+1)x_{k+1}; x_1, \ldots, x_k)$, $k \geq 1$, *with* x_2 *being computed from* $\mathbf{F}(2x_2, x_1, x_0, 0) = \mathbf{0}$. *Here,* $F_k = F_k(6x_3, \ldots, (k+1)(k+2)x_{k+2}; 2x_2, \ldots, (k+1)x_{k+1}; x_1, \ldots, x_k), k \geq 1$ *is the vector of k-th Adomian polynomials of the components of the vector function* $\mathbf{F}$.

Proof. Assume that the solution $\mathbf{x}(t)$ can be expanded as in (11). Then, we expand the left side of the equation $\mathbf{F}(\mathbf{x}''(t), \mathbf{x}'(t), \mathbf{x}(t), t) = 0$ in terms of the Adomian polynomials to obtain the following:

$$\mathbf{F}\left(\sum_{k=0}^{\infty}(k+1)(k+2)x_{k+2}t^k, \sum_{k=0}^{\infty}(k+1)x_{k+1}t^k, \sum_{k=0}^{\infty} x_k t^k, t\right) = \sum_{k=0}^{\infty} F_k t^k = \mathbf{0}. \tag{16}$$

This gives the following:

$$F_k(2x_2, \ldots, (k+1)(k+2)x_{k+2}; 2x_2, \ldots, (k+1)x_{k+1}; x_1, \ldots, x_k) = 0,\ k \geq 1, \tag{17}$$

or

$$(k+1)(k+2)Jx_{k+2} + F_k(2x_2, \ldots, k(k+1)x_{k+1}, 0; 2x_2, \ldots, (k+1)x_{k+1}; x_1, \ldots, x_k),\ k \geq 1, \tag{18}$$

which leads to the following recursion system for computing the differential transform of the solution $\mathbf{x}(t)$

$$(k-1)kJx_k = -F_{k-2}(2x_2, \ldots, (k-2)(k-1)x_{k-1}, 0; 2x_2, \ldots, (k-1)x_{k-1}; x_1, \ldots, x_{k-2}),\ k \geq 3. \tag{19}$$

□

Finally, we have Algorithms 1–3:

Algorithm 1 : DTM algorithm for solution of $\mathbf{F}(\mathbf{x}', \mathbf{x}, t) = 0,\ \mathbf{x}(0) = \mathbf{x}_0$.

Input
$K,\ n,\ \mathbf{x}_0 \in \mathbb{R}^n,\ \mathbf{F} \in \mathbb{R}^n$

Output
approximate solution: $\mathbf{x}(t) = \sum_{k=0}^{K} x_k t^k$
For $2 \leq k \leq K$ do
compute $r_{k-1} := F_{k-1}(x_1, \ldots, (k-1)x_{k-1}, 0; x_1, \ldots, x_{k-1})$
end do

Initialization
$x_0 := \mathbf{x}_0$
solve for x_1 the algebraic system: $\mathbf{F}(x_1, x_0, 0) = \mathbf{0}$
compute $J = \mathbf{F}_{\mathbf{x}'}(x_1, x_0, 0)$
For $2 \leq k \leq K$ do
solve for x_k the linear algebraic system: $kJx_k = -r_{k-1}$
end do

Algorithm 2 : DTM algorithm for solution of $\mathbf{F}(\mathbf{x}'', \mathbf{x}', \mathbf{x}, t) = \mathbf{0},\ \mathbf{x}(0) = \mathbf{x}_0,\ \mathbf{x}'(0) = \mathbf{x}_1$.

Input
$K,\ n,\ \mathbf{x}_0,\ \mathbf{x}_1 \in \mathbb{R}^n,\ \mathbf{F} \in \mathbb{R}^n$

Output
approximate solution: $\mathbf{x}(t) = \sum_{k=0}^{K} x_k t^k$
For $3 \leq k \leq K$ do
compute $r_{k-2} := F_{k-2}(2x_2, \ldots, (k-2)(k-1)x_{k-1}, 0; 2x_2, \ldots, (k-1)x_{k-1}; x_1, \ldots, x_{k-2})$
end do

Initialization
$x_0 := \mathbf{x}_0,\ x_1 := \mathbf{x}_1$
solve for x_2 the algebraic system: $\mathbf{F}(2x_2, x_1, x_0, 0) = \mathbf{0}$
compute $J = \mathbf{F}_{\mathbf{x}''}(2x_2, x_1, x_0, 0)$
For $3 \leq k \leq K$ do
solve for x_k the linear algebraic system: $k(k-1)Jx_k = -r_{k-2}$
end do

We can extend the interval of convergence of the solution by using a multi-stage DTM (MsDTM). The procedure of the MsDTM for $\mathbf{F}(\mathbf{x}', \mathbf{x}, t) = \mathbf{0}$, with $\mathbf{x}(0) = \mathbf{x}_0$ is described in the following algorithm:

Algorithm 3 : MsDTM algorithm.

1. Choose the number N of sub-divisions and the order of approximation K.
2. Sub-divide the interval $[0, T]$ into N equal sub-intervals $[t_{i-1}, t_i]$, of size $h = T/N$, where $t_i = ih,\ i = 1, \ldots, N$.
3. For $i = 1$, apply the DTM to $\mathbf{F}(\mathbf{x}', \mathbf{x}, t) = \mathbf{0}$ on $[0, t_i]$ using $\mathbf{x}_0^i = \mathbf{x}(0)$ to obtain an approximate solution $\mathbf{x}^i(t) = \sum_{k=0}^{K} x_k^i t^k$.
4. From $i = 2$ to $i = N$, apply the DTM to $\mathbf{F}(\mathbf{x}', \mathbf{x}, t) = \mathbf{0}$ on $[t_{i-1}, t_i]$ using $\mathbf{x}_0^i = \mathbf{x}^{i-1}(t_{i-1})$ to obtain an approximate solution $\mathbf{x}^i(t) = \sum_{k=0}^{K} x_k^i (t - t_{i-1})^k$.
5. Repeat step 4.
6. An approximate solution to the initial-value problem $\mathbf{F}(\mathbf{x}', \mathbf{x}, t) = \mathbf{0}$, $\mathbf{x}(0) = \mathbf{x}_0$ on $[0, T]$ is given by

$$\mathbf{x}(t) = \begin{cases} \mathbf{x}^1(t), & 0 \le t \le t_1 \\ \mathbf{x}^2(t), & t_1 < t \le t_2 \\ \vdots & \\ \mathbf{x}^N(t), & t_{N-1} < t \le T \end{cases}$$

A similar multi-stage algorithm can be given for second-order implicit systems $\mathbf{F}(\mathbf{x}'', \mathbf{x}', \mathbf{x}, t) = \mathbf{0}$.

4. Numerical Results

In this section, we showcase the effectiveness and accuracy of our proposed technique by solving several first and second-order implicit ODEs and systems of ODEs with different nonlinearities. Notably, these examples cannot be solved using Maple 2024, either numerically or through power series expansions. Additionally, conventional methods like Runge–Kutta do not apply to these types of ODEs and systems. Utilizing Algorithms 1 and 2, our technique successfully computes the solutions in power series form. The implementation of our algorithms was carried out in Maple 2024. Our method is capable of providing the exact power series expansion of the solution, provided that all computations are performed with precision. To solve these examples over larger intervals, we used the multi-stage versions of Algorithms 1 and 2. The numerical results show that the MsDTM provides accurate solutions to these implicit systems of ODEs.

Example 1. *In this example, we consider the following fully implicit nonlinear system of first-order ordinary differential equations, as follows:*

$$\mathbf{x}_1' - \frac{\mathbf{x}_1}{\mathbf{x}_2'^2 + \mathbf{x}_2^2} + \mathbf{x}_1 - \mathbf{x}_2 = 0, \tag{20}$$

$$\mathbf{x}_2' - \frac{\mathbf{x}_2}{\mathbf{x}_1'^2 + \mathbf{x}_1^2} + \mathbf{x}_1 + \mathbf{x}_2 = 0, \tag{21}$$

where $t > 0$. The sought solution is as follows: $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)^T$. The differential system (20) and (21) has the form (1), with $n = 2$, and $m = 1$. This system is subject to the following initial conditions:

$$\mathbf{x}(0) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{22}$$

We should emphasize here that Maple 2024 software cannot solve this system. However, our technique has successfully obtained the exact solution of this implicit system in power series. By applying Algorithm 1 to (20) and (21) with initial condition (22), we can recursively determine the DTM expansion coefficients x_k for $k \ge 2$. According to the initial condition (22), we have the following:

$$x_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{23}$$

Then, we determine the differential transform $x_1 = (x_{11}, x_{12})^T$ *from the algebraic system as follows:*

$$\mathbf{F}(x_1, x_0, 0) = \mathbf{0}. \tag{24}$$

The above system has a unique solution x_1 *and simplifies to the following:*

$$x_{11} - \frac{x_{01}}{x_{12}^2 + x_{02}^2} + x_{01} - x_{02} = 0, \tag{25}$$

$$x_{12} - \frac{x_{02}}{x_{11}^2 + x_{01}^2} + x_{01} + x_{02} = 0. \tag{26}$$

Solving this algebraic system, we obtain the following:

$$x_1 = \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \tag{27}$$

Then, we compute the Jacobian $J := \mathbf{F}_{\mathbf{x}'}$ *of the function* $\mathbf{F}$ *with respect to* $\mathbf{x}'$ *at the point* $(x_1, x_0, 0)$*. We obtain the following:*

$$J = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}. \tag{28}$$

Using Theorem 1, we have the following recursion for the differential transform x_k

$$x_k = -(kJ)^{-1} r_{k-1},\ k \geq 2, \tag{29}$$

where

$$r_{k-1} := F_{k-1}(x_1, \ldots, (k-1)x_{k-1}, 0; x_1, \ldots, (k-1)x_{k-1}, 0; x_1, \ldots, \ldots, x_{k-1}). \tag{30}$$

Using Algorithm 1, we compute the inverse of the matrix and the right-hand sides r_{k-1} *for* $k \geq 2$ *and deduce the following:*

For $k = 2$*, we have the following:*

$$x_2 = -(2J)^{-1} r_1 = -\begin{pmatrix} \frac{1}{2} & 0 \\ -1 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -\frac{1}{2} \end{pmatrix}. \tag{31}$$

For $k = 3$*, we have the following:*

$$x_3 = -(3J)^{-1} r_2 = -\begin{pmatrix} \frac{1}{3} & 0 \\ -\frac{2}{3} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} \frac{1}{2} \\ 1 \end{pmatrix} = \begin{pmatrix} -\frac{1}{6} \\ 0 \end{pmatrix}. \tag{32}$$

For $k = 4$*, we have the following:*

$$x_4 = -(4J)^{-1} r_3 = -\begin{pmatrix} \frac{1}{4} & 0 \\ -\frac{1}{2} & \frac{1}{4} \end{pmatrix} \begin{pmatrix} 0 \\ -\frac{1}{6} \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{24} \end{pmatrix}. \tag{33}$$

For $k = 5$*, we have the following:*

$$x_5 = -(5J)^{-1} r_4 = -\begin{pmatrix} \frac{1}{5} & 0 \\ -\frac{2}{5} & \frac{1}{5} \end{pmatrix} \begin{pmatrix} -\frac{1}{24} \\ -\frac{1}{12} \end{pmatrix} = \begin{pmatrix} \frac{1}{120} \\ 0 \end{pmatrix}. \tag{34}$$

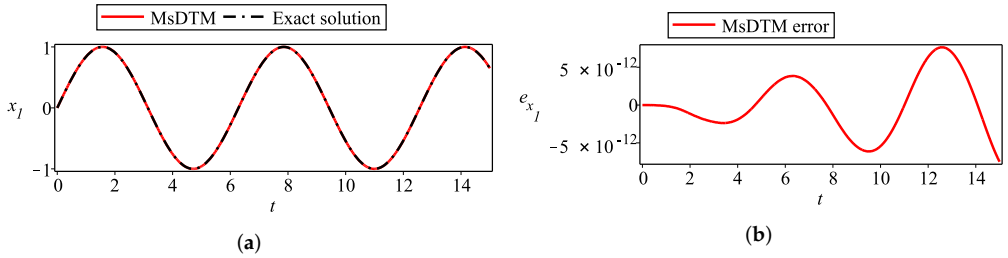
For $k = 6$, we have the following:

$$x_6 = -(6J)^{-1} r_5 = -\begin{pmatrix} \frac{1}{6} & 0 \\ -\frac{1}{3} & \frac{1}{6} \end{pmatrix} \begin{pmatrix} 0 \\ \frac{1}{120} \end{pmatrix} = \begin{pmatrix} 0 \\ -\frac{1}{720} \end{pmatrix}. \tag{35}$$

Finally, we construct the following approximate solution with $K = 7$:

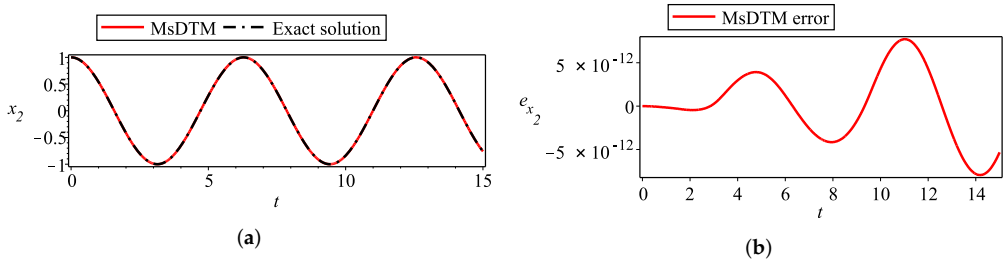
$$\mathbf{x}(t) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} + t\begin{pmatrix} 1 \\ 0 \end{pmatrix} + t^2\begin{pmatrix} 0 \\ -\frac{1}{2} \end{pmatrix} + t^3\begin{pmatrix} -\frac{1}{6} \\ 0 \end{pmatrix} + t^4\begin{pmatrix} 0 \\ \frac{1}{24} \end{pmatrix} + t^5\begin{pmatrix} \frac{1}{120} \\ 0 \end{pmatrix} + t^6\begin{pmatrix} 0 \\ -\frac{1}{720} \end{pmatrix}. \tag{36}$$

One can easily check that the above approximation forms the first terms of the Maclaurin power series of the exact solution $\mathbf{x}(t) = (\sin(t), \cos(t))^T$. It is worth noting here that we obtain the exact values of the successive DTM expansion coefficients. It is known that by increasing the number of terms in the series solution, the error decreases, and the convergence interval of the truncated series increases. To compute accurate numerical solutions over large internals and extend the interval of convergence of the solution, we apply the multi-stage DTM (MsDTM) algorithm given in Section 3. To solve this example with the MsDTM, we use an order of approximation $K = 7$ and subdivide the solution interval $[0, T] = [0, 15]$ into $N = 400$ sub-intervals. Starting the recursion with $x_0 = x(0) = (0, 1)^T$, we obtained the numerical results shown in Figures 1 and 2. Each of these figures shows the numerical solution (solid line), the exact solution (dashed line), and the corresponding absolute error. From the graph (a), we can see that the numerical solution approximates well the exact solution. In each graph (b), we can see the corresponding absolute error. These graphs show that the errors of approximation using the MsDTM are less than 1×10^{-11}. These numerical results show that equations in this example are satisfied with high accuracy and that the MsDTM computes accurate numerical solutions over large intervals. To increase the accuracy in the approximate MsDTM solution, one can increase the approximation order K or the number N of sub-divisions of the solution interval $[0, T]$. To maintain a given accuracy while increasing the size of the solution interval, one may need to increase the number of terms K and or increase the number N of interval subdivisions.



(**a**) (**b**)

Figure 1. Solution component and its absolute error for $K = 7, N = 400$. (**a**) Approximate solution component x_1. (**b**) Absolute error e_{x_1}.



(**a**) (**b**)

Figure 2. Solution component and its absolute error for $K = 7, N = 400$. (**a**) Approximate solution component x_2. (**b**) Absolute error e_{x_2}.

Example 2. *In this example, we consider the following fully implicit nonlinear system of first-order ordinary differential equations:*

$$10\mathbf{x}_1' - \cos(\pi\mathbf{x}_2') + \frac{5}{3}\mathbf{x}_1 + 10\mathbf{x}_2 + \cos\left(\pi\mathbf{x}_1 - \frac{\pi}{6}\mathbf{x}_2\right) = 0, \tag{37}$$

$$10\mathbf{x}_2' + \sin(\pi\mathbf{x}_1') - 10\mathbf{x}_1 + \frac{5}{3}\mathbf{x}_2 + \sin\left(\frac{\pi}{6}\mathbf{x}_1 + \pi\mathbf{x}_2\right) = 0, \tag{38}$$

where $t > 0$. The sought solution is as follows: $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)^T$. The differential system (37) and (38) has the form (1), with $n = 2$, and $m = 1$. This system is subject to the following initial conditions:

$$\mathbf{x}(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \tag{39}$$

We should emphasize here that Maple 2024 software cannot solve this system. However, our technique has successfully obtained the exact solution of this implicit system in power series. By applying Algorithm 1 to (37) and (38) with initial condition (39), we can recursively determine the DTM expansion coefficients x_k for $k \geq 2$. According to the initial condition (39), we have the following:

$$x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \tag{40}$$

Then, we determine the differential transform $x_1 = (x_{11}, x_{12})^T$ from the algebraic system, as follows:

$$\mathbf{F}(x_1, x_0, 0) = \mathbf{0}. \tag{41}$$

The above system has a unique solution x_1 and simplifies to the following:

$$10x_{11} - \cos(\pi x_{12}) + \frac{5}{3} + \cos(\pi) = 0, \tag{42}$$

$$10x_{12} + \sin(\pi x_{11}) - 10 + \sin\left(\frac{\pi}{6}\right) = 0. \tag{43}$$

Solving this algebraic system, we obtain the following:

$$x_1 = \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} = \begin{pmatrix} -\frac{1}{6} \\ 1 \end{pmatrix}. \tag{44}$$

Then, we compute the Jacobian $J := \mathbf{F}_{\mathbf{x}'}$ of the function $\mathbf{F}$ with respect to $\mathbf{x}'$ at the point $(x_1, x_0, 0)$. We obtain the following:

$$J = \begin{pmatrix} \frac{1}{2}\pi\sqrt{3} & 10 \\ 10 & 0 \end{pmatrix}. \tag{45}$$

Using Theorem 1, we have the following recursion for the differential transform x_k:

$$x_k = -(kJ)^{-1} r_{k-1},\ k \geq 2, \tag{46}$$

where

$$r_{k-1} := F_{k-1}(x_1, \ldots, (k-1)x_{k-1}, 0; x_1, \ldots, (k-1)x_{k-1}, 0; x_1, \ldots, \ldots, x_{k-1}). \tag{47}$$

Using Algorithm 1, we compute the inverse of the matrix and the right-hand sides r_{k-1} for $k \geq 2$ and deduce the following:

For $k = 2$, we have the following:

$$x_2 = -(2J)^{-1} r_1 = -\begin{pmatrix} 0 & \frac{1}{20} \\ \frac{1}{20} & -\frac{1}{400}\pi\sqrt{3} \end{pmatrix} \begin{pmatrix} \frac{10}{3} + \frac{35}{72}\pi\sqrt{3} \\ \frac{175}{18} \end{pmatrix} = \begin{pmatrix} -\frac{35}{72} \\ -\frac{1}{6} \end{pmatrix}. \quad (48)$$

For $k = 3$, we have the following:

$$x_3 = -(3J)^{-1} r_2 = -\begin{pmatrix} 0 & \frac{1}{30} \\ \frac{1}{30} & -\frac{1}{600}\pi\sqrt{3} \end{pmatrix} \begin{pmatrix} \frac{55}{12} - \frac{107}{864}\pi\sqrt{3} \\ -\frac{535}{216} \end{pmatrix} = \begin{pmatrix} \frac{107}{1296} \\ -\frac{11}{72} \end{pmatrix}. \quad (49)$$

For $k = 4$, we have the following:

$$x_4 = -(4J)^{-1} r_3 = -\begin{pmatrix} 0 & \frac{1}{40} \\ \frac{1}{40} & -\frac{1}{800}\pi\sqrt{3} \end{pmatrix} \begin{pmatrix} -\frac{175}{162} - \frac{1081}{15552}\pi\sqrt{3} \\ -\frac{5405}{3888} \end{pmatrix} = \begin{pmatrix} \frac{1081}{31104} \\ \frac{35}{1296} \end{pmatrix}. \quad (50)$$

For $k = 5$, we have the following:

$$x_5 = -(5J)^{-1} r_4 = -\begin{pmatrix} 0 & \frac{1}{50} \\ \frac{1}{50} & -\frac{1}{1000}\pi\sqrt{3} \end{pmatrix} \begin{pmatrix} -\frac{4705}{15552} + \frac{6121}{373248}\pi\sqrt{3} \\ \frac{30605}{93312} \end{pmatrix} = \begin{pmatrix} -\frac{6121}{933120} \\ \frac{941}{155520} \end{pmatrix}. \quad (51)$$

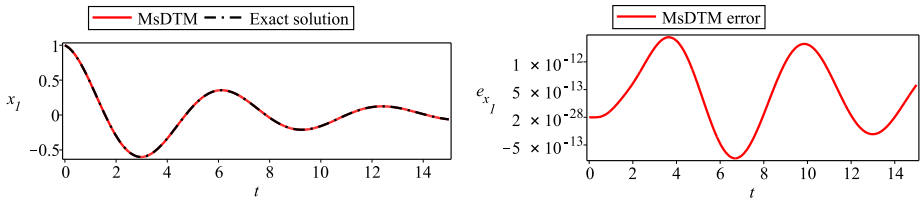
For $k = 6$, we have the following:

$$x_6 = -(6J)^{-1} r_5 = -\begin{pmatrix} 0 & \frac{1}{60} \\ \frac{1}{60} & -\frac{1}{1200}\pi\sqrt{3} \end{pmatrix} \begin{pmatrix} \frac{1177}{15552} + \frac{5551}{2239488}\pi\sqrt{3} \\ \frac{27755}{559872} \end{pmatrix} = \begin{pmatrix} -\frac{5551}{6718464} \\ -\frac{1177}{933120} \end{pmatrix}. \quad (52)$$

Finally, we construct the following approximate solution with $K = 7$:

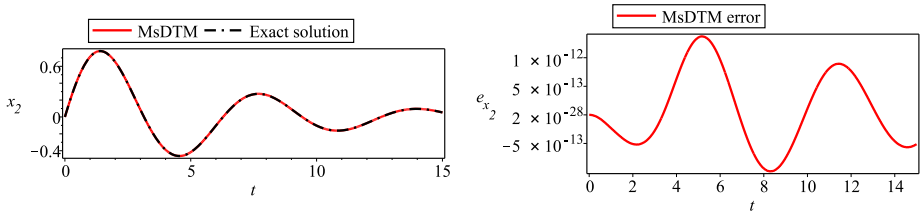
$$\mathbf{x}(t) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + t\begin{pmatrix} -\frac{1}{6} \\ 1 \end{pmatrix} + t^2\begin{pmatrix} -\frac{35}{72} \\ -\frac{1}{6} \end{pmatrix} + t^3\begin{pmatrix} \frac{107}{1296} \\ -\frac{11}{72} \end{pmatrix} + t^4\begin{pmatrix} \frac{1081}{31104} \\ \frac{35}{1296} \end{pmatrix} + t^5\begin{pmatrix} -\frac{6121}{933120} \\ \frac{941}{155520} \end{pmatrix} + t^6\begin{pmatrix} -\frac{5551}{6718464} \\ -\frac{1177}{933120} \end{pmatrix}. \quad (53)$$

One can easily check that the above approximation forms the first terms of the Maclaurin power series of the exact solution $\mathbf{x}(t) = (e^{-t/6}\cos(t), e^{-t/6}\sin(t))^T$. It is worth noting here that we obtain the exact values of the successive DTM expansion coefficients. It is known that by increasing the number of terms in the series solution, the error decreases, and the convergence interval of the truncated series increases (Figures 3 and 4).



(**a**) (**b**)

Figure 3. Solution component and its absolute error for $K = 7, N = 400$. (**a**) Approximate solution component x_1. (**b**) Absolute error e_{x_1}.



(**a**) (**b**)

Figure 4. Solution component and its absolute error for $K = 7, N = 400$. (**a**) Approximate solution component x_2. (**b**) Absolute error e_{x_2}.

Example 3. *In this example, we consider the following fully implicit nonlinear system of first-order ordinary differential equations:*

$$\arctan(t\mathbf{x}_1') - \mathbf{x}_2' - \frac{1}{1+t^2} - \arctan\left(\frac{t}{1+t^2}\right) = 0, \tag{54}$$

$$\mathbf{x}_1' + \arctan\left(t\mathbf{x}_2' + \frac{t}{1+t^2}\right) - \frac{1}{1+t^2} = 0, \tag{55}$$

where $t > 0$. The sought solution is $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)^T$. The differential system (54) and (55) has the form (1), with $n = 2$, and $m = 1$. This system is subject to the following initial conditions:

$$\mathbf{x}(0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{56}$$

We should emphasize here that Maple 2024 software cannot solve this system. However, our technique has successfully obtained the exact solution of this implicit system in power series. By applying Algorithm 1 to (54) and (55) with initial condition (56), we can recursively determine the DTM expansion coefficients x_k for $k \geq 2$. According to the initial condition (56), we have the following:

$$x_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{57}$$

Then, we determine the differential transform $x_1 = (x_{11}, x_{12})^T$ from the algebraic system as follows:

$$\mathbf{F}(x_1, x_0, 0) = \mathbf{0}. \tag{58}$$

The above system has a unique solution x_1 and simplifies to the following:

$$x_{12} + 1 = 0, \tag{59}$$

$$x_{11} - 1 = 0. \tag{60}$$

Solving this algebraic system, we obtain the following:

$$x_1 = \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}. \tag{61}$$

Then, we compute the Jacobian $J := \mathbf{F}_{\mathbf{x}'}$ of the function $\mathbf{F}$ with respect to $\mathbf{x}'$ at the point $(x_1, x_0, 0)$. We obtain the following:

$$J = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. \tag{62}$$

Using Theorem 1, we have the following recursion for the differential transform: x_k

$$x_k = -(kJ)^{-1} r_{k-1}, \; k \geq 2, \tag{63}$$

where

$$r_{k-1} := F_{k-1}(x_1, \ldots, (k-1)x_{k-1}, 0; x_1, \ldots, (k-1)x_{k-1}, 0; x_1, \ldots, \ldots, x_{k-1}). \tag{64}$$

Using Algorithm 1, we compute the inverse of the matrix and the right-hand sides r_{k-1} for $k \geq 2$ and deduce the following:

For $k = 2$, we have the following:

$$x_2 = -(2J)^{-1} r_1 = -\begin{pmatrix} 0 & \frac{1}{2} \\ -\frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{65}$$

For $k = 3$, we have the following:

$$x_3 = -(3J)^{-1} r_2 = -\begin{pmatrix} 0 & \frac{1}{3} \\ -\frac{1}{3} & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -\frac{1}{3} \\ \frac{1}{3} \end{pmatrix}. \tag{66}$$

For $k = 4$, we have the following:

$$x_4 = -(4J)^{-1} r_3 = -\begin{pmatrix} 0 & \frac{1}{4} \\ -\frac{1}{4} & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{67}$$

For $k = 5$, we have the following:

$$x_5 = -(5J)^{-1} r_4 = -\begin{pmatrix} 0 & \frac{1}{5} \\ -\frac{1}{5} & 0 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \end{pmatrix} = \begin{pmatrix} \frac{1}{5} \\ -\frac{1}{5} \end{pmatrix}. \tag{68}$$

For $k = 6$, we have the following:

$$x_6 = -(6J)^{-1} r_5 = -\begin{pmatrix} 0 & \frac{1}{6} \\ -\frac{1}{6} & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{69}$$

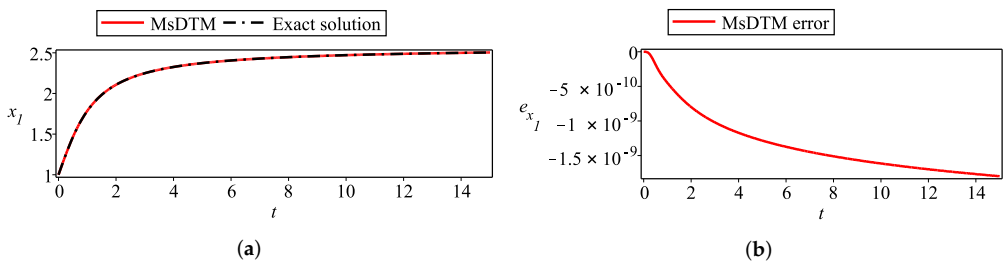
For $k = 7$, we have the following:

$$x_7 = -(7J)^{-1} r_6 = -\begin{pmatrix} 0 & \frac{1}{7} \\ -\frac{1}{7} & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -\frac{1}{7} \\ \frac{1}{7} \end{pmatrix}. \tag{70}$$

Finally, we construct the following approximate solution with $K = 7$:

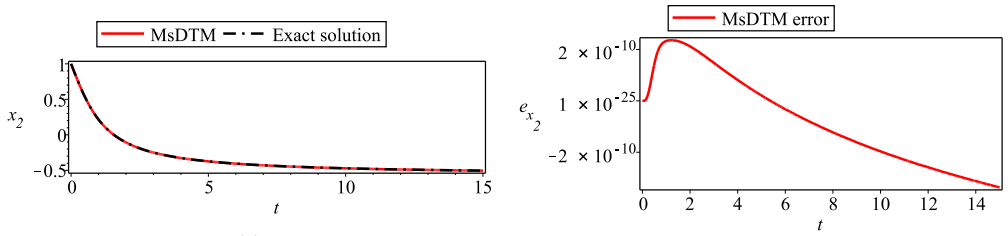
$$\mathbf{x}(t) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + t \begin{pmatrix} 1 \\ -1 \end{pmatrix} + t^3 \begin{pmatrix} -\frac{1}{3} \\ \frac{1}{3} \end{pmatrix} + t^5 \begin{pmatrix} \frac{1}{5} \\ -\frac{1}{5} \end{pmatrix} + t^7 \begin{pmatrix} -\frac{1}{7} \\ \frac{1}{7} \end{pmatrix}. \tag{71}$$

One can easily check that the above approximation forms the first terms of the Maclaurin power series of the exact solution $\mathbf{x}(t) = (1 + \arctan(t), 1 - \arctan(t))^T$. It is worth noting here that we obtain the exact values of the successive DTM expansion coefficients. It is known that by increasing the number of terms in the series solution, the error decreases, and the convergence interval of the truncated series increases (Figures 5 and 6).



(**a**) (**b**)

Figure 5. Solution component and its absolute error for $K = 7, N = 400$. (**a**) Approximate solution component x_1. (**b**) Absolute error e_{x_1}.



(**a**) (**b**)

Figure 6. Solution component and its absolute error for $K = 7, N = 400$. (**a**) Approximate solution component x_2. (**b**) Absolute error e_{x_2}.

Example 4. *In this example, we consider the following second-order implicit nonlinear equation:*

$$2\mathbf{x}''(t) + \sin(\mathbf{x}''(t) - t^3) + 4\mathbf{x}'(t) - t^4 - 2t^3 - 4 = 0, \ t \geq 0, \tag{72}$$

subject to the initial conditions

$$\mathbf{x}(0) = 0, \ \mathbf{x}'(0) = 1. \tag{73}$$

We should emphasize here that Maple 2024 software could not solve this equation numerically or in power series in its present form. However, there is no trouble for our technique to obtain the solution of this implicit equation in power series. By applying Algorithm 2 to (72) with initial condition (73), we can recursively determine the DTM expansion coefficients x_k for $k \geq 2$. The differential Equation (72) has the following form:

$$\mathbf{F}(\mathbf{x}'', \mathbf{x}', \mathbf{x}, t) = 0, \tag{74}$$

where $\mathbf{F}(\mathbf{x}'', \mathbf{x}', \mathbf{x}, t) = 2\mathbf{x}'' + \sin(\mathbf{x}'' - t^3) + 4\mathbf{x}' - t^4 - 2t^3 - 4$. According to the initial conditions (73), we have the following:

$$x_0 = \mathbf{x}(0) = 0, \ x_1 = \mathbf{x}'(0) = 1. \tag{75}$$

Then, we determine the second DTM expansion coefficient x_2 from the algebraic equation, as follows:

$$\mathbf{F}(2x_2, x_1, x_0, 0) = 0. \tag{76}$$

This equation has a unique solution x_2 and simplifies to the following:

$$4x_2 + \sin(2x_2) = 0. \tag{77}$$

Solving this equation, we obtain the following:

$$x_2 = 0. \tag{78}$$

Then, we compute the Jacobian J of the function $\mathbf{F}$ with respect to $\mathbf{x}''$ at the point $(2x_2, x_1, x_0, 0)$. For the present example, J is just a scalar, as follows:

$$J = 2 + \cos(2x_2) = 3. \tag{79}$$

Using Theorem 2, we have the following recursion for the DTM expansion coefficients x_k:

$$x_k = -\left(\frac{1}{3k(k-1)}\right) r_{k-2}, \ k \geq 3, \tag{80}$$

where

$$r_{k-2} := F_{k-2}(2x_2, \ldots, (k-2)(k-1)x_{k-1}, 0; 2x_2, \ldots, (k-1)x_{k-1}; x_1, \ldots, x_{k-2}). \tag{81}$$

Using Algorithm 2, we compute the right-hand sides r_{k-2} for $k \geq 3$ and obtain the following:

$$r_1 = 8x_2 = 0. \tag{82}$$

Using (80), we deduce the following:

$$x_3 = 0. \tag{83}$$

Then, we have the following:

$$r_2 = 12x_3 - 18x_3^2 \sin(2x_2) = 0. \tag{84}$$

Using (80), we deduce the following:

$$x_4 = 0. \tag{85}$$

Then, we have the following:

$$r_3 = -2 + 16x_4 - \cos(2x_2) - 72x_4x_3 \sin(2x_2) - 36x_3^3 \cos(2x_2) = -3. \tag{86}$$

Using (80), we deduce the following:

$$x_5 = \frac{1}{20}. \tag{87}$$

In a similar way, we can show that $r_k = 0$ for $k \geq 4$, hence, $x_k = 0$ for $k \geq 6$. Finally, we construct the following approximate solution based on five terms:

$$\mathbf{x}(t) = \sum_{k=0}^{5} x_k t^k = t + \frac{1}{20} t^5. \tag{88}$$

One can easily check that the exact solution is $\mathbf{x}(t) = t + \frac{1}{20}t^5$. It is worth noting here that we obtain the exact values of the successive DTM expansion coefficients.

Example 5. *In this example, we consider the following fully implicit nonlinear system of second-order ordinary differential equations:*

$$3\mathbf{x_1}'' + \mathbf{x_2}'' + \sin(\mathbf{x_1}'' + 2\mathbf{x}_1 - 2\mathbf{x}_2) - \mathbf{x_2}'^2 + 4(\mathbf{x}_1 - \mathbf{x}_2 + \mathbf{x}_1\mathbf{x}_2) = 0, \tag{89}$$

$$4\mathbf{x_2}'' + \sin(\mathbf{x_1}'' - 2\cos(2t)) - \mathbf{x_1}'^2 + 4(2\mathbf{x_2} - 2\mathbf{x}_1 + \mathbf{x}_1\mathbf{x}_2) = 0, \tag{90}$$

where $t > 0$. The sought solution is as follows: $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)^T$. The differential system (89) and (90) has the form (1), with $n = 2$, and $m = 2$. This system is subject to the following initial conditions:

$$\mathbf{x}(0) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \ \mathbf{x}'(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{91}$$

We should emphasize here that Maple 2024 software cannot solve this system. However, our technique has successfully obtained the exact solution of this implicit system in power series. By applying Algorithm 2 to (89) and (90) with initial condition (91), we can recursively determine

the DTM expansion coefficients x_k for $k \geq 3$. According to the initial condition (91), we have the following:

$$x_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \; x_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{92}$$

Then, we determine the differential transform $x_2 = (x_{21}, x_{22})^T$ from the algebraic system, as follows:

$$\mathbf{F}(2x_2, x_1, x_0, 0) = \mathbf{0}. \tag{93}$$

The above system has a unique solution x_2 and simplifies to the following:

$$6x_{21} + 2x_{22} + \sin(2x_{21} - 2) - 4 = 0, \tag{94}$$

$$8x_{22} + \sin(2x_{21} - 2) + 8 = 0. \tag{95}$$

Solving this algebraic system, we obtain the following:

$$x_2 = \begin{pmatrix} x_{21} \\ x_{22} \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}. \tag{96}$$

Then, we compute the Jacobian $J := \mathbf{F}_{\mathbf{x}''}$ of the function $\mathbf{F}$ with respect to $\mathbf{x}''$ at the point $(x_2, x_1, x_0, 0)$. We obtain the following:

$$J = \begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix}. \tag{97}$$

Using Theorem 2, we have the following recursion for the differential transform: x_k

$$x_k = -(k(k-1)J)^{-1} r_{k-2}, \; k \geq 3, \tag{98}$$

where

$$r_{k-2} := F_{k-2}(2x_2, \ldots, (k-2)(k-1)x_{k-1}, 0; 2x_2, \ldots, (k-1)x_{k-1}; x_1, \ldots, \ldots, x_{k-2}). \tag{99}$$

Using Algorithm 2, we compute the inverse of the matrix and the right-hand sides r_{k-2} for $k \geq 3$ and deduce the following:

For $k = 3$, we have the following:

$$x_3 = -(6J)^{-1} r_1 = \begin{pmatrix} -\frac{2}{45} & \frac{1}{90} \\ \frac{1}{90} & -\frac{2}{45} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{100}$$

For $k = 4$, we have the following:

$$x_4 = -(12J)^{-1} r_2 = \begin{pmatrix} -\frac{1}{45} & \frac{1}{180} \\ \frac{1}{180} & -\frac{1}{45} \end{pmatrix} \begin{pmatrix} 12 \\ -12 \end{pmatrix} = \begin{pmatrix} -\frac{1}{3} \\ \frac{1}{3} \end{pmatrix}. \tag{101}$$

For $k = 5$, we have the following:

$$x_5 = -(20J)^{-1} r_3 = \begin{pmatrix} -\frac{1}{75} & \frac{1}{300} \\ \frac{1}{300} & -\frac{1}{75} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{102}$$

For $k = 6$, we have the following:

$$x_6 = -(30J)^{-1} r_4 = \begin{pmatrix} -\frac{2}{225} & \frac{1}{450} \\ \frac{1}{450} & -\frac{2}{225} \end{pmatrix} \begin{pmatrix} -4 \\ 4 \end{pmatrix} = \begin{pmatrix} \frac{2}{45} \\ -\frac{2}{45} \end{pmatrix}. \tag{103}$$

For $k = 7$, we have the following:

$$x_7 = -(42J)^{-1} r_5 = \begin{pmatrix} -\frac{2}{315} & \frac{1}{630} \\ \frac{1}{630} & -\frac{2}{315} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{104}$$

For $k = 8$, we have the following:

$$x_8 = -(56J)^{-1} r_6 = \begin{pmatrix} -\frac{1}{210} & \frac{1}{840} \\ \frac{1}{840} & -\frac{1}{210} \end{pmatrix} \begin{pmatrix} \frac{8}{15} \\ -\frac{8}{15} \end{pmatrix} = \begin{pmatrix} -\frac{1}{315} \\ \frac{1}{315} \end{pmatrix}. \tag{105}$$

For $k = 9$, we have the following:

$$x_9 = -(72J)^{-1} r_7 = \begin{pmatrix} -\frac{1}{270} & \frac{1}{1080} \\ \frac{1}{1080} & -\frac{1}{270} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{106}$$

For $k = 10$, we have the following:

$$x_{10} = -(90J)^{-1} r_8 = \begin{pmatrix} -\frac{2}{675} & \frac{1}{1350} \\ \frac{1}{1350} & -\frac{2}{675} \end{pmatrix} \begin{pmatrix} -\frac{4}{105} \\ \frac{4}{105} \end{pmatrix} = \begin{pmatrix} \frac{2}{14175} \\ -\frac{2}{14175} \end{pmatrix}. \tag{107}$$

Finally, we construct the following approximate solution with $K = 10$:

$$\mathbf{x}(t) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} + t^2 \begin{pmatrix} 1 \\ -1 \end{pmatrix} + t^4 \begin{pmatrix} -\frac{1}{3} \\ \frac{1}{3} \end{pmatrix} + t^6 \begin{pmatrix} \frac{2}{45} \\ -\frac{2}{45} \end{pmatrix} + t^8 \begin{pmatrix} -\frac{1}{315} \\ \frac{1}{315} \end{pmatrix} + t^{10} \begin{pmatrix} \frac{2}{14175} \\ -\frac{2}{14175} \end{pmatrix}. \tag{108}$$

One can easily check that the above approximation forms the first terms of the Maclaurin power series of the exact solution $\mathbf{x}(t) = (\sin^2(t), \cos^2(t))^T$. It is worth noting here that we obtain the exact values of the successive DTM expansion coefficients. It is known that by increasing the number of terms in the series solution, the error decreases, and the convergence interval of the truncated series increases (Figures 7 and 8).

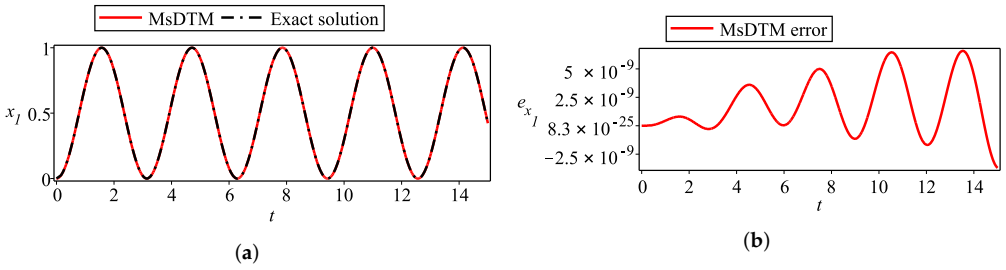


Figure 7. Solution component and its absolute error for $K = 10, N = 500$. (**a**) Approximate solution component x_1. (**b**) Absolute error e_{x_1}.

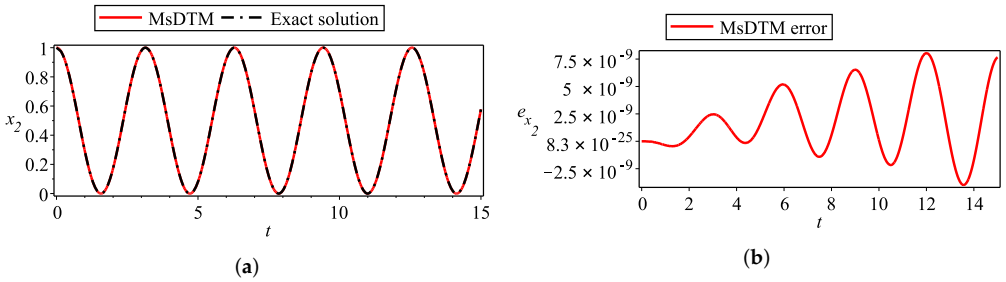


Figure 8. Solution component and its absolute error for $K = 10, N = 500$. (**a**) Approximate solution component x_2. (**b**) Absolute error e_{x_2}.

Example 6. *In this example, we consider the following fully implicit nonlinear system of second-order ordinary differential equations:*

$$\mathbf{x}_1'' + 3\mathbf{x}_2'' - \frac{1}{1+(\mathbf{x}_1''+\mathbf{x}_2'')^2} + 2(\mathbf{x}_2 - 1)\sin(t) - 2\mathbf{x}_2'\cos(t) + 1 = 0, \tag{109}$$

$$3\mathbf{x}_1'' + \mathbf{x}_2'' + \frac{1}{1+(\mathbf{x}_1''+\mathbf{x}_2'')^2} + 2(\mathbf{x}_1 - 1)\sin(t) - 2\mathbf{x}_1'\cos(t) - 1 = 0, \tag{110}$$

where $t > 0$. The sought solution is $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)^T$. The differential system (109) and (110) has the form (1), with $n = 2$, and $m = 2$. This system is subject to the following initial conditions:

$$\mathbf{x}(0) = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \ \mathbf{x}'(0) = \begin{pmatrix} 1 \\ -1 \end{pmatrix}. \tag{111}$$

We should emphasize here that Maple 2024 software cannot solve this system. However, our technique has successfully obtained the exact solution of this implicit system in power series. By applying Algorithm 2 to (109) and (110) with initial condition (111), we can recursively determine the DTM expansion coefficients x_k for $k \geq 3$. According to the initial condition (111), we have the following:

$$x_0 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \ x_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}. \tag{112}$$

Then, we determine the differential transform $x_2 = (x_{21}, x_{22})^T$ from the following algebraic system:

$$\mathbf{F}(2x_2, x_1, x_0, 0) = \mathbf{0}. \tag{113}$$

The above system has a unique solution x_2 and simplifies to the following:

$$2x_{21} + 6x_{22} - \frac{1}{1+(x_{21}+x_{22})^2} + 3 = 0, \tag{114}$$

$$6x_{21} + 2x_{22} + \frac{1}{1+(x_{21}+x_{22})^2} - 3 = 0. \tag{115}$$

Solving this algebraic system, we obtain the following:

$$x_2 = \begin{pmatrix} x_{21} \\ x_{22} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix}. \tag{116}$$

Then, we compute the Jacobian $J := \mathbf{F}_{\mathbf{x}''}$ of the function $\mathbf{F}$ with respect to $\mathbf{x}''$ at the point $(x_2, x_1, x_0, 0)$. We obtain the following:

$$J = \begin{pmatrix} 1 & 3 \\ 3 & 1 \end{pmatrix}. \tag{117}$$

Using Theorem 2, we have the following recursion for the differential transform: x_k

$$x_k = -(k(k-1)J)^{-1} r_{k-2}, \ k \geq 3, \tag{118}$$

where

$$r_{k-2} := F_{k-2}(2x_2, \ldots, (k-2)(k-1)x_{k-1}, 0; 2x_2, \ldots, (k-1)x_{k-1}; x_1, \ldots, \ldots, x_{k-2}). \tag{119}$$

Using Algorithm 2, we compute the inverse of the matrix and the right-hand sides r_{k-2} for $k \geq 3$ and deduce the following:

For $k = 3$, we have the following:

$$x_3 = -(6J)^{-1} r_1 = \begin{pmatrix} \frac{1}{48} & -\frac{1}{16} \\ -\frac{1}{16} & \frac{1}{48} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{120}$$

For $k = 4$, we have the following:

$$x_4 = -(12J)^{-1} r_2 = \begin{pmatrix} \frac{1}{96} & -\frac{1}{32} \\ -\frac{1}{32} & \frac{1}{96} \end{pmatrix} \begin{pmatrix} -3 \\ 3 \end{pmatrix} = \begin{pmatrix} -\frac{1}{8} \\ \frac{1}{8} \end{pmatrix}. \tag{121}$$

For $k = 5$, we have the following:

$$x_5 = -(20J)^{-1} r_3 = \begin{pmatrix} \frac{1}{160} & -\frac{3}{160} \\ -\frac{3}{160} & \frac{1}{160} \end{pmatrix} \begin{pmatrix} -\frac{8}{3} \\ \frac{8}{3} \end{pmatrix} = \begin{pmatrix} -\frac{1}{15} \\ \frac{1}{15} \end{pmatrix}. \tag{122}$$

For $k = 6$, we have the following:

$$x_6 = -(30J)^{-1} r_4 = \begin{pmatrix} \frac{1}{240} & -\frac{1}{80} \\ -\frac{1}{80} & \frac{1}{240} \end{pmatrix} \begin{pmatrix} -\frac{1}{4} \\ \frac{1}{4} \end{pmatrix} = \begin{pmatrix} -\frac{1}{240} \\ \frac{1}{240} \end{pmatrix}. \tag{123}$$

For $k = 7$, we have the following:

$$x_7 = -(42J)^{-1} r_5 = \begin{pmatrix} \frac{1}{336} & -\frac{1}{112} \\ -\frac{1}{112} & \frac{1}{336} \end{pmatrix} \begin{pmatrix} \frac{14}{15} \\ -\frac{14}{15} \end{pmatrix} = \begin{pmatrix} \frac{1}{90} \\ -\frac{1}{90} \end{pmatrix}. \tag{124}$$

For $k = 8$, we have the following:

$$x_8 = -(56J)^{-1} r_6 = \begin{pmatrix} \frac{1}{448} & -\frac{3}{448} \\ -\frac{3}{448} & \frac{1}{448} \end{pmatrix} \begin{pmatrix} \frac{217}{360} \\ -\frac{217}{360} \end{pmatrix} = \begin{pmatrix} \frac{31}{5760} \\ -\frac{31}{5760} \end{pmatrix}. \tag{125}$$

For $k = 9$, we have the following:

$$x_9 = -(72J)^{-1} r_7 = \begin{pmatrix} \frac{1}{576} & -\frac{1}{192} \\ -\frac{1}{192} & \frac{1}{576} \end{pmatrix} \begin{pmatrix} \frac{8}{315} \\ -\frac{8}{315} \end{pmatrix} = \begin{pmatrix} \frac{1}{5670} \\ -\frac{1}{5670} \end{pmatrix}. \tag{126}$$

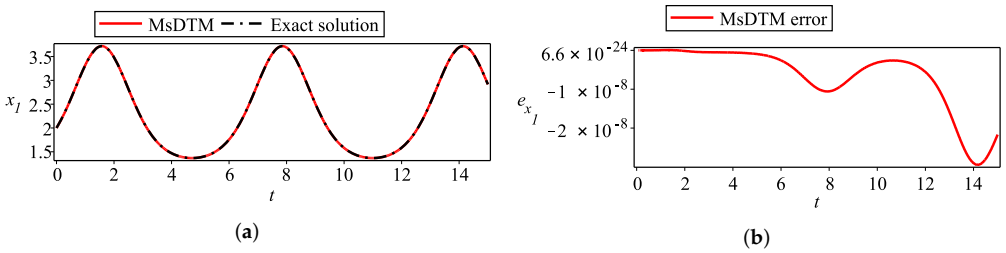
For $k = 10$, we have the following:

$$x_{10} = -(90J)^{-1} r_8 = \begin{pmatrix} \frac{1}{720} & -\frac{1}{240} \\ -\frac{1}{240} & \frac{1}{720} \end{pmatrix} \begin{pmatrix} -\frac{2951}{20160} \\ \frac{2951}{20160} \end{pmatrix} = \begin{pmatrix} -\frac{2951}{3628800} \\ \frac{2951}{3628800} \end{pmatrix}. \tag{127}$$

Finally, we construct the following approximate solution with $K = 10$:

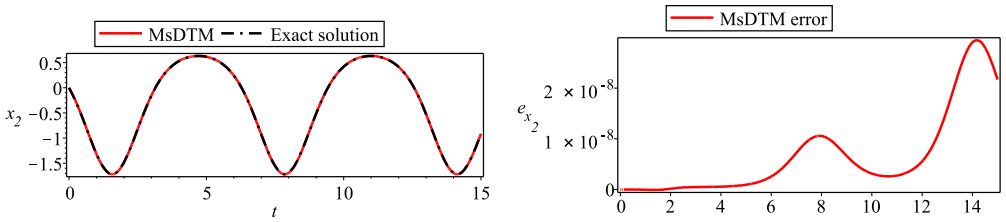
$$\begin{aligned} \mathbf{x}(t) \quad = \quad & \begin{pmatrix} 2 \\ 0 \end{pmatrix} + t \begin{pmatrix} 1 \\ -1 \end{pmatrix} + t^2 \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} + t^4 \begin{pmatrix} -\frac{1}{8} \\ \frac{1}{8} \end{pmatrix} + t^5 \begin{pmatrix} -\frac{1}{5} \\ \frac{1}{5} \end{pmatrix} + t^6 \begin{pmatrix} -\frac{1}{240} \\ \frac{1}{240} \end{pmatrix} + t^7 \begin{pmatrix} \frac{1}{90} \\ -\frac{1}{90} \end{pmatrix} \\ & + t^8 \begin{pmatrix} \frac{31}{5760} \\ -\frac{31}{5760} \end{pmatrix} + t^9 \begin{pmatrix} \frac{1}{5670} \\ -\frac{1}{5670} \end{pmatrix} + t^{10} \begin{pmatrix} -\frac{2951}{3628800} \\ -\frac{2951}{3628800} \end{pmatrix}. \end{aligned}$$

One can easily check that the above approximation forms the first terms of the Maclaurin power series of the exact solution $\mathbf{x}(t) = (1 + e^{\sin(t)}, 1 - e^{\sin(t)})^T$. It is worth noting here that we obtain the exact values of the successive DTM expansion coefficients. It is known that by increasing the number of terms in the series solution, the error decreases, and the convergence interval of the truncated series increases (Figures 9 and 10).



(**a**) (**b**)

Figure 9. Solution component and its absolute error for $K = 10, N = 400$. (**a**) Approximate solution component x_1. (**b**) Absolute error e_{x_1}.



(**a**) (**b**)

Figure 10. Solution component and its absolute error for $K = 10, N = 400$. (**a**) Approximate solution component x_2. (**b**) Absolute error e_{x_2}.

5. Discussion

In this work, we introduced a multi-stage differential transform method (MsDTM) combined with Adomian polynomials as an effective semi-analytical tool for solving fully implicit nonlinear ordinary differential systems. We demonstrated the validity of two theorems that underpin our numerical algorithms. The method was successfully applied to solve implicit ordinary differential systems of both the first- and second-order, with the potential for generalization to higher-order systems. The first unknown recursion from an algebraic system is initially solved uniquely, so the solution process starts here. This algebraic system can be nonlinear; if necessary, Newton's method can be used to compute an accurate solution. Subsequently, the differential transform method (DTM) is used to turn the implicit ordinary differential system into a linear algebraic recursion system with a constant nonsingular coefficient matrix. This linear system can then be solved using a variety of techniques. Section 3 describes the DTM algorithm, which enables us to solve the differential system in the form of a power series. The DTM technique will produce the exact solution in power series form if the differential system has an exact solution and all computations have been performed exactly. We thereby extend the interval of convergence by using the multi-stage DTM technique described in Section 3, even though the power series solution could only converge over a short interval. One of the MsDTM method's strengths is the extension of the convergence interval. We coded the DTM and MsDTM processes using the Maple 2024 package to apply the approach. These techniques were successfully applied to six numerical examples of fully implicit nonlinear differential systems. Maple's built-in procedures cannot resolve these problems numerically or in power series. For each of the six cases, the DTM converted the differential system into an algebraic recursion system for the power series solution's coefficients. We used the multi-stage MsDTM to improve the convergence of the DTM solution, hence strengthening the solution that emerged from the truncated power series. One of the main advantages of our approach is that it avoids needless computations, including first- or lower-order system reduction in the differential system. This considerably reduces the computing load, increases efficiency, and simplifies the overall technique. Furthermore, the DTM computes its coefficients by a straightforward procedure that can be easily coded in computer algebra

systems like Maple, Mathematica, or Matlab. Both the DTM and the MsDTM approaches handled the numerical cases solved here with effectiveness. We used the absolute error between the exact solutions and the numerical solutions to assess the accuracy of the results. Figures 1–10 show comparisons between the MsDTM approximations and the exact solutions for all numerical examples. For all approximations, the absolute error was very low. If required, the error can be further minimized by splitting the solution interval into additional intervals or by raising the order of the DTM approximations. For the numerical examples under consideration, the absolute error varied between 10^{-11} and 10^{-8}, the interval of solutions was $[0, 15]$, the chosen parameter was $K = 7$, and the average number of interval subdivisions was $N = 400$. On the other hand, techniques like Runge–Kutta (RK) require that the differential system be in an explicit form. Furthermore, using implicit numerical integration methods for ODEs typically necessitates solving nonlinear algebraic problems at each iteration step, a process that can be computationally intensive. However, by avoiding these complications, the MsDTM technique offers a more effective and useful way to solve IODEs.

A comprehensive summary of the six case studies used to assess the performance of the proposed MsDTM is given in Table 1. Using an approximation order of $K = 7$ and $N = 400$ subdivisions consistently yielded maximum absolute errors (MAE) $< 5 \times 10^{-11}$, indicating remarkable precision over the interval [0, 15]. The results show that this was true for first-order systems (Cases 1–3). A validation benchmark, Case 4, a second-order system with an exact solution, confirms the theoretical robustness of the DTM method. Maintaining MAE below 7.5×10^{-9} and 2.5×10^{-8} for more complex second-order systems (Cases 5–6) required a slightly greater number of subdivisions ($N = 500$ and $N = 400$ respectively) and an increased approximation order of $K = 10$. These results show the method's high accuracy for solving implicit ordinary differential equations of both the first- and second-order. Furthermore, the constant performance throughout several examples emphasizes the scalability and adaptability of MsDTM, therefore enabling MsDTM to be a powerful method for solving a broad spectrum of nonlinear IODEs.

Table 1. Summary of main results and parameters of the six case studies. MAE stands for maximum absolute error.

Case	Order	Equations	*K*	*N*	MAE	Interval
1	1	2	7	400	$<1 \times 10^{-11}$	[0, 15]
2	1	2	7	400	$<5 \times 10^{-11}$	[0, 15]
3	1	2	7	400	$<2 \times 10^{-9}$	[0, 15]
4	2	1	5	–	Exact solution	[0, Convergent]
5	2	2	10	500	$<7.5 \times 10^{-9}$	[0, 15]
6	2	2	10	400	$<2.5 \times 10^{-8}$	[0, 15]

6. Conclusions

A novel semi-analytical method, combining Adomian polynomials (AP) with the multi-stage differential transform method (MsDTM), was developed for solving implicit nonlinear ODEs. Direct application of the DTM to implicit systems simplifies the solution procedure and reduces computational costs, thus eliminating the need for explicit or quasi-linear transformations. Because AP methodically expanded nonlinear terms as power series, complex nonlinearities were efficiently handled, preserving the system's implicit nature. The MsDTM improved upon the DTM by dividing the solution domain into smaller sub-intervals and iteratively producing solutions using truncated series as initial conditions, thus achieving high accuracy over larger domains. Numerical tests demonstrated that our method is highly accurate and reliable for solving challenging first- and second-order

implicit nonlinear ODEs, even those that pose difficulties for standard methods such as Runge–Kutta, producing results very close to the exact solutions. Using AP, the MsDTM provides a robust and computationally efficient method to solve fully implicit nonlinear ODEs, improving accuracy, convergence, and applicability to mathematical problems. In research and education, the application of the MsDTM can be extended to Maple, Mathematica, and Matlab.

Future work topics will include the following:

1. Solving implicit nonlinear ordinary differential systems with movable singularities.
2. Comparing the accuracy and efficiency between our approach and other numerical and semi-analytical techniques to facilitate evaluation and enhancement.
3. Examining convergence and error analysis to strengthen the approach's theoretical foundation and applicability.
4. Modifying the approach to increase its real-world applicability by solving problems that include stochastic elements, boundary conditions, integral equations, implicit fractional differential equations, and time-varying algebraic equations.

Author Contributions: Conceptualization, B.B.; validation, H.V.-L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: This manuscript includes all the data used to support its findings.

Conflicts of Interest: The authors declare that no conflicts of interest exist regarding the publication of this manuscript.

References

1. Chauhan, V.; Srivastava, P.K. Computational techniques based on Runge-Kutta method of various order and type for solving differential equations. *Int. J. Math. Eng. Manag. Sci.* **2019**, *4*, 375–386. [CrossRef]
2. Sharma, P.L.; Kumar, A. Review Paper on the Runge-Kutta Methods to study Numerical Solutions of Initial Value Problems in Ordinary Differential Equations. *Int. J. Appl. Math. Stat. Sci.* **2021**, *10*, 45–54. [CrossRef]
3. Anakira, F.N.; Jameel, A.; Hijazi, M.; Alomari, A.K.; Man, N. A new approach for solving multi-pantograph type delay differential equations. *Int. J. Electr. Comput. Eng.* **2022**, *12*, 1859–1868. [CrossRef]
4. Wazwaz, A.M. A reliable modification of Adomian decomposition method. *Appl. Math. Comput.* **1999**, *102*, 77–86. [CrossRef]
5. Al-Eybani, A.M.D. Adomian Decomposition Method to Solve the Second Order Ordinary Differential Equations. *Int. J. Math. Phys. Sci. Res.* **2020**, *8*, 87–92. [CrossRef]
6. AL-Mazmumy, M.; Alsulami, A.A.; Bakodah, H.O.; Alzaid, N. Modified Adomian Method through Efficient Inverse Integral Operators to Solve Nonlinear Initial-Value Problems for Ordinary Differential Equations. *Axioms* **2022**, *11*, 698. [CrossRef]
7. Alsulami, A.A.; AL-Mazmumy, M.; Bakodah, H.O.; Alzaid, N. A Method for the Solution of Coupled System of Emden–Fowler–Type Equations. *Symmetry* **2022**, *14*, 843. [CrossRef]
8. He, J.H.; El-Dib, Y.O. Homotopy perturbation method for fangzhu oscillator. *J. Math. Chem.* **2020**, *58*, 2245–2253. [CrossRef]
9. Al-Ahmad, S.; Sulaiman, I.M.; Mamat, M.; Kamfa, K. Solutions of classes of differential equations using modified differential transform method. *J. Math. Comput. Sci.* **2020**, *10*, 2360–2382.
10. Al-Ahmad, S.; Sulaiman, I.M.; Mamat, M.; Ghazali, P.L. Modified differential transform scheme for solving systems of first order ordinary differential equations. *J. Math. Comput. Sci.* **2021**, *22*, 73–84. [CrossRef]
11. Mehne, H.H. Differential transform method: A comprehensive review and analysis. *Iran. J. Numer. Anal. Optim.* **2022**, *12*, 629–657. [CrossRef]
12. Hoefkens, J.; Berz, M.; Makino, K. Computing Validated Solutions of Implicit Differential Equations. *Adv. Comput. Math.* **2003**, *19*, 231–253. [CrossRef]
13. Van Gorder, R.A. Optimal homotopy analysis and control of error for implicitly defined fully nonlinear differential equations. *Numer. Algorithms* **2019**, *81*, 181–196. [CrossRef]
14. Lapshin, V.V. On the implicit equations of a mechanical systems motion. *J. Phys. Conf. Ser.* **2020**, *1705*, 012030. [CrossRef]
15. Chadli, O.; Ansari, Q.H.; Al-Homidan, S. Existence of Solutions for Nonlinear Implicit Differential Equations: An Equilibrium Problem Approach. *Numer. Funct. Anal. Optim.* **2016**, *37*, 1385–1419. [CrossRef]

16. Feckan, M. A Survey on the Melnikov Theory for Implicit Ordinary Differential Equations with Applications to RLC Circuits. In *Mathematics Applied to Engineering, Modelling, and Social Issues*; Smith, F.T., Dutta, H., Mordeson, J.N., Eds.; Springer International Publishing: Cham, Switzerland, 2019; Volume 200, pp. 121–160. [CrossRef]
17. Fliess, M.; Levine, J.; Martin, P.; Rouchon, P. Implicit differential equations and Lie-Backlund mappings. In Proceedings of the 1995 34th IEEE Conference on Decision and Control, New Orleans, LA, USA, 13–15 December 1995; Volume 3, pp. 2704–2709. [CrossRef]
18. Polyanin, A.D.; Zhurov, A.I. Parametrically defined nonlinear differential equations, differential–algebraic equations, and implicit ODEs: Transformations, general solutions, and integration methods. *Appl. Math. Lett.* **2017**, *64*, 59–66. [CrossRef]
19. Semler, C.; Gentleman, W.; Paidoussis, M.P. Numerical solutions of second order implicit non-linear ordinary differential equations. *J. Sound Vib.* **1996**, *195*, 553–574. [CrossRef]
20. Luxemburg, W.A.J. On the Convergence of Successive Approximations in the Theory of Ordinary Differential Equations. *Can. Math. Bull.* **1958**, *1*, 9–20. [CrossRef]
21. Braun, E.; Seiler, W.M.; Seiss, M. On the Numerical Analysis and Visualisation of Implicit Ordinary Differential Equations. *arXiv* **2020**. [CrossRef]
22. Jankowski, T. A Numerical Solution of Implicit Ordinary Differential Equations. *Demonstr. Math.* **1992**, *25*. [CrossRef]
23. Jackiewicz, Z.; Kwapisz, M. On numerical integration of implicit ordinary differential equations. *Appl. Math.* **1981**, *26*, 97–110. [CrossRef]
24. Verner, J.H. Quadratures for Implicit Differential Equations. *SIAM J. Numer. Anal.* **1970**, *7*, 373–385. [CrossRef]
25. Odibat, Z.M.; Bertelle, C.; Aziz-Alaoui, M.; Duchamp, G.H. A multi-step differential transform method and application to non-chaotic or chaotic systems. *Comput. Math. Appl.* **2010**, *59*, 1462–1472. [CrossRef]
26. Ertürk, V.S.; Odibat, Z.M.; Momani, S. The Multi-Step Differential Transform Method and Its Application to Determine the Solutions of Non-Linear Oscillators. *Adv. Appl. Math. Mech.* **2012**, *4*, 422–438. [CrossRef]
27. Alomari, A. A new analytic solution for fractional chaotic dynamical systems using the differential transform method. *Comput. Math. Appl.* **2011**, *61*, 2528–2534. [CrossRef]
28. Rach, R. A New Definition of the Adomian Polynomials. *Kybernetes* **2008**, *37*, 910–955. [CrossRef]
29. Rach, R. A Convenient Computational Form for the Adomian Polynomials. *J. Math. Anal. Appl.* **1984**, *102*, 415–419. [CrossRef]
30. Wazwaz, A.M. A New Algorithm for Calculating Adomian Polynomials for Nonlinear Operators. *Appl. Math. Comput.* **2000**, *111*, 53–69. [CrossRef]
31. Duan, J.S. Recurrence Triangle for Adomian Polynomials. *Appl. Math. Comput.* **2010**, *216*, 1235–1241. [CrossRef]
32. Duan, J.S. An Efficient Algorithm for the Multivariable Adomian Polynomials. *Appl. Math. Comput.* **2010**, *217*, 2456–2467. [CrossRef]
33. Duan, J.S. Convenient Analytic Recurrence Algorithms for the Adomian Polynomials. *Appl. Math. Comput.* **2011**, *217*, 6337–6348. [CrossRef]
34. Rach, R.; Adomian, G. Transformation of Series. *Appl. Math. Lett.* **1991**, *4*, 69–71.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.