

## Article

# A New Generalized Taylor-Like Explicit Method for Stiff Ordinary Differential Equations

Essam R. El-Zahar <sup>1,2,\*</sup> , José Tenreiro Machado <sup>3,\*</sup>  and Abdelhalim Ebaid <sup>4</sup> 

<sup>1</sup> Department of Mathematics, College of Sciences and Humanities in Al-Kharj, Prince Sattam bin Abdulaziz University, Alkharj 11942, Saudi Arabia

<sup>2</sup> Department of Basic Engineering Science, Faculty of Engineering, Menofia University, Shebin El-Kom 32511, Egypt

<sup>3</sup> Institute of Engineering, Polytechnic of Porto, Rua Dr. António Bernardino de Almeida, 431, 4249-015 Porto, Portugal

<sup>4</sup> Department of Mathematics, Faculty of Sciences, University of Tabuk, P.O. Box 741, Tabuk 71491, Saudi Arabia; aebaid@ut.edu.sa

\* Correspondence: er.elzahar@psau.edu.sa or essam\_zahar2006@yahoo.com (E.R.E.-Z.); jtm@isep.ipp.pt (J.T.M.)

Received: 15 October 2019; Accepted: 21 November 2019; Published: 1 December 2019



**Abstract:** A new generalised Taylor-like explicit method for stiff ordinary differential equations (ODEs) is proposed. The algorithm is presented in its component and vector forms. The error and stability analysis of the method are developed showing that it has an arbitrary high order of convergence and the L-stability property. Moreover, it is verified that several integration schemes are special cases of the new general form. The method is applied on stiff problems and the numerical solutions are compared with those of the classical Taylor-like integration schemes. The results show that the proposed method is accurate and overcomes the shortcoming of the classical Taylor-like schemes in their component and vector forms.

**Keywords:** ordinary differential equations; nonlinear high order methods; L-stability; Taylor-like explicit methods

## 1. Introduction

Most of the mathematical models describing real-world phenomena are often stiff ordinary differential equations (SODEs). Such problems involve a wide range of temporal scales and solving these SODEs requires careful treatment. Efficient numerical and semi-analytical numerical methods for solving stiff problems must have good accuracy, wide region of stability and low computational effort. It is well known that explicit linear multistep methods are not absolutely stable. In addition, most of the implicit methods are absolutely stable and work adequately with SODEs, but they involve a higher computational load per step than the explicit methods [1–22]. Wu and Xia [9–11] presented a vector form for A-stable explicit one step Taylor-like method for solving stiff ODE systems. Wu and Xia [9] proposed a vector form for the two low accuracy methods [5] to be applied on stiff ODE systems. Wu and Xia [10] proposed a vector form for the sixth-order Taylor-like explicit method [7] to be applied on stiff ODE systems. Wu and Xia [9,10] showed the superiority of the Taylor-like methods when formulated in vector form compared to their component form. More recently, Wu and Xia [11] derived a general form of the Taylor-like explicit method and derived its corresponding vector form. One of the main advantages of using Taylor-like methods is that the approximate solution is given as an arbitrary order piecewise analytical function defined on the sub-intervals of the whole integration interval. This property offers different facility for adaptive error control [19,20]. Moreover, the Taylor-like

method [11] is an arbitrary high order A-stable method that avoids extremely small stepsizes during the integration procedure. To avoid the analytical computation of the successive derivatives involved in the Taylor-like methods, numerical differentiation [21,22], automatic differentiation [23], differential transformation [24–26] and Infinity Computer with a new numeral system [27–32] can be used. In fact, Taylor-like explicit methods [5,7,9–11] have computational drawbacks with zero-component derivative or zero-vector norm in their component or vector forms, respectively. Moreover, the computational errors due to the round-off, particularly in the case of long time intervals or large time-step sizes, may lead to very small values of the derivatives or very large values of the derivatives' ratio at some grid points. Consequently, poor results are obtained. To overcome these limitations a new generalised Taylor-like explicit method for SODEs and stiff ODE systems is present. The method is developed both in its component and vector forms. The error and stability analysis of the method are presented. It is shown that the new method has an arbitrary order of convergence and the L-stability property. Indeed, many other integration schemes are essentially special cases of the proposed general form method. The method is applied on stiff test problems and the numerical results are compared with those in the literature. The results show that the proposed method is accurate and avoids the shortcoming of the classical Taylor-like explicit methods both in their component and vector forms.

## 2. Generalised Taylor-Like Method

Consider the initial-value problem given by:

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0, \quad (1)$$

where  $y(t), f(t, y(t)) \in \mathbb{R}$  and  $t \in [t_0, T] \subset \mathbb{R}$ . The interval  $[t_0, T]$  is divided into sub-intervals  $[t_j, t_{j+1}]$  with  $t_j = t_0 + jh$ , such that  $h$  is the time-step size. Assume that the solution of (1) can be approximated by

$$\bar{y}(t) = a(e^{b(t-t_j)} - 1) + \sum_{n=0}^m c_n \frac{(t-t_j)^n}{n!}, \quad a, b \neq 0, \quad t \in [t_j, t_{j+1}], \quad (2)$$

where  $a, b$  and  $c_n$  are unknowns and determined later from the local truncation error and stability analysis of the method. By considering Taylor's expansions of  $y(t)$  and  $\bar{y}(t)$  about  $t_j$ , we have:

$$y_{j+1} = y_j + \sum_{n=1}^{\infty} y_j^{(n)} \frac{h^n}{n!}, \quad (3)$$

where  $y_j^{(n)} = \frac{d^n y}{dt^n}(t_j)$  and

$$\bar{y}_{j+1} = c_0 + \sum_{n=1}^m (ab^n + c_n) \frac{h^n}{n!} + \sum_{n=m+1}^{\infty} ab^n \frac{h^n}{n!}. \quad (4)$$

The approximation in (2) is constructed so that  $y_{j+1}$  and  $\bar{y}_{j+1}$  agree and their derivatives up to order  $m+1$ , that is

$$\left( c_0 + \sum_{n=1}^m (ab^n + c_n) \frac{h^n}{n!} + \sum_{n=m+1}^{\infty} ab^n \frac{h^n}{n!} \right) - \left( y_j + \sum_{n=1}^{\infty} y_j^{(n)} \frac{h^n}{n!} \right) = O(h^{m+2}). \quad (5)$$

Equating the coefficients of the same powers of  $h^L$ ,  $L = 0, 1, \dots, m+1$ , in (5) by zero, results in the following system of nonlinear equations

$$\left. \begin{aligned} c_0 - y_j &= 0 \\ ab^n + c_n - y_j^{(n)} &= 0, \quad n = 1, \dots, m \\ ab^{m+1} - y_j^{(m+1)} &= 0 \end{aligned} \right\}. \quad (6)$$

Solving the first and last equations in (6) results in

$$c_0 = y_j, \quad a = \frac{y_j^{(m+1)}}{b^{m+1}}. \quad (7)$$

Consequently, solving the remaining equations in (6) results in

$$c_n = y_j^{(n)} - \frac{y_j^{(m+1)}}{b^{m+1-n}}, \quad n = 1, \dots, m. \quad (8)$$

From (2), (7) and (8) we obtain the following integration algorithm:

$$\bar{y}_{j+1} = y_j + \frac{y_j^{(m+1)}}{b^{m+1}} (e^{bh} - 1) + \sum_{n=1}^m \left( y_j^{(n)} - \frac{y_j^{(m+1)}}{b^{m+1-n}} \right) \frac{h^n}{n!}, \quad b \neq 0. \quad (9)$$

### 2.1. Local Truncation Error

The local truncation error  $T_{j+1}$  is readily obtained from subtracting (9) from the Taylor series expansion in (3) and collecting terms in  $h$

$$T_{j+1} = \sum_{s=0}^{\infty} \left( y_j^{(m+1)} b^{s+1} - y_j^{(m+2+s)} \right) \frac{h^{m+2+s}}{(m+2+s)!}. \quad (10)$$

It is clear that relation (9) has at least  $m+1$  order of accuracy.

### 2.2. Stability Analysis

In order to examine the integration scheme (9) for the stability, let us consider the differential equation,

$$y' = \lambda y,$$

where  $\lambda$  is a complex constant and  $\text{Re}(\lambda) < 0$ . For this equation, Equation (9) results in

$$y_{j+1} = y_j + y_j \frac{\lambda^{m+1}}{b^{m+1}} (e^{bh} - 1) + y_j \sum_{n=1}^m \left( \lambda^n - \frac{\lambda^{m+1}}{b^{m+1-n}} \right) \frac{h^n}{n!}, \quad b \neq 0. \quad (11)$$

For  $b = \frac{y_j^{(k)}}{y_j^{(k-1)}} = \lambda$ ,  $k = 1, 2, \dots, m+1$ , Equation (11) results in

$$y_{j+1} = y_j e^{\lambda h}.$$

Setting  $z = \lambda h$  in the above equation, the amplification factor is given by

$$R(z) = e^z$$

and thus we have obtained the following A-stable and L-stable method with at least  $(m+1)$  order of accuracy

$$y_{j+1} = \sum_{n=0}^m y_j^{(n)} \frac{h^n}{n!} + y_j^{(m+1)} \left( \frac{y_j^{(k-1)}}{y_j^{(k)}} \right)^{m+1} \left( e^{\frac{y_j^{(k)} h}{y_j^{(k-1)}}} - \sum_{n=0}^m \left( \frac{y_j^{(k)} h}{y_j^{(k-1)}} \right)^n \frac{1}{n!} \right), y_j^{(k)} y_j^{(k-1)} \neq 0, \quad (12)$$

where  $k = 1, 2, \dots, m+1$ .

The local truncation error of (12) is given by

$$T_{j+1} = \sum_{s=0}^{\infty} \left( y_j^{(m+1)} \left( \frac{y_j^{(k)}}{y_j^{(k-1)}} \right)^{s+1} - y_j^{(m+2+s)} \right) \frac{h^{m+2+s}}{(m+2+s)!}. \quad (13)$$

### 2.3. Consistency

Subtracting  $y_j$  from both sides of (12) and dividing the result by  $h$  leads to

$$\frac{y_{j+1} - y_j}{h} = y_j^{(1)} + \sum_{n=2}^m y_j^{(n)} \frac{h^{n-1}}{n!} + y_j^{(m+1)} \left( \frac{y_j^{(k-1)}}{y_j^{(k)}} \right)^{m+1} \left( \frac{e^{\frac{y_j^{(k)} h}{y_j^{(k-1)}}} - 1}{h} - \left( \frac{y_j^{(k)}}{y_j^{(k-1)}} \right) - \sum_{n=2}^m \left( \frac{y_j^{(k)}}{y_j^{(k-1)}} \right)^n \frac{h^{n-1}}{n!} \right). \quad (14)$$

Taking the limit as  $h$  tends to zero, on both sides of (14), yields

$$\lim_{h \rightarrow 0} \frac{y_{j+1} - y_j}{h} = y_j^{(1)} + y_j^{(m+1)} \left( \frac{y_j^{(k-1)}}{y_j^{(k)}} \right)^{m+1} \left( \lim_{h \rightarrow 0} \frac{e^{\frac{y_j^{(k)} h}{y_j^{(k-1)}}} - 1}{h} - \left( \frac{y_j^{(k)}}{y_j^{(k-1)}} \right) \right) = y_j^{(1)} = f(t_j, y_j), \quad (15)$$

showing that the method defined in (12) is consistent.

Thus we have the following theorem:

**Theorem 1.** The generalised Taylor-like explicit method (12) is L-stable and convergent with at least  $(m+1)$  order of accuracy.

For a fixed value of  $m$  and different values of  $k$  ( $k = 1, 2, \dots, m+1$ ) the method formulated in (12) results in  $(m+1)$  L-stable different integration schemes with the same order of accuracy  $O(m+1)$ .

**Remark 1.** For  $m = 0$  and  $k = 1$  the Equation (12) is reduced to

$$y_{j+1} = y_j e^{y_j^{(1)}/y_j^{(1)} h}, y_j y_j^{(1)} \neq 0, \quad (16)$$

which is the first order L-stable method in [5,9].

**Remark 2.** For  $m = 0$  and  $k = 2$  the Equation (12) is reduced to

$$y_{j+1} = y_j + \frac{(y_j^{(1)})^2}{y_j^{(2)}} \left( e^{y_j^{(2)}/y_j^{(1)} h} - 1 \right), y_j^{(1)} y_j^{(2)} \neq 0, \quad (17)$$

which is the second order L-stable methods in [5,9].

**Remark 3.** For  $m = 5$  and  $k = 6$  the Equation (12) is reduced to

$$y_{j+1} = \sum_{n=0}^5 y_j^{(n)} \frac{h^n}{n!} + y_j^{(6)} \left( \frac{y_j^{(5)}}{y_j^{(6)}} \right)^6 \left( e^{\frac{y_j^{(6)} h}{y_j^{(5)}}} - \sum_{n=0}^5 \left( \frac{y_j^{(6)} h}{y_j^{(5)}} \right)^n \frac{1}{n!} \right), \quad y_j^{(6)} y_j^{(5)} \neq 0, \quad (18)$$

which is the sixth-order L-stable method in [7].

**Remark 4.** For  $m = 4$  and  $k = 6$  the Equation (12) is reduced to

$$y_{j+1} = \sum_{n=0}^4 y_j^{(n)} \frac{h^n}{n!} + y_j^{(5)} \left( \frac{y_j^{(5)}}{y_j^{(6)}} \right)^6 \left( e^{\frac{y_j^{(6)} h}{y_j^{(5)}}} - \sum_{n=0}^4 \left( \frac{y_j^{(6)} h}{y_j^{(5)}} \right)^n \frac{1}{n!} \right), \quad y_j^{(6)} y_j^{(5)} \neq 0, \quad (19)$$

which is the sixth-order L-stable method in [9].

**Remark 5.** For  $m = 5$  and  $k = 7$  the Equation (12) is reduced to

$$y_{j+1} = \sum_{n=0}^5 y_j^{(n)} \frac{h^n}{n!} + y_j^{(6)} \left( \frac{y_j^{(6)}}{y_j^{(7)}} \right)^6 \left( e^{\frac{y_j^{(7)} h}{y_j^{(6)}}} - \sum_{n=0}^5 \left( \frac{y_j^{(7)} h}{y_j^{(6)}} \right)^n \frac{1}{n!} \right), \quad y_j^{(6)} y_j^{(7)} \neq 0, \quad (20)$$

which is the seventh-order L-stable method in [8].

**Remark 6.** When  $k$  is set to  $m + 2$ , Equation (12) is reduced to

$$y_{j+1} = \sum_{n=0}^m y_j^{(n)} \frac{h^n}{n!} + y_j^{(m+1)} \left( \frac{y_j^{(m+1)}}{y_j^{(m+2)}} \right)^{m+1} \left( e^{\frac{y_j^{(m+2)} h}{y_j^{(m+1)}}} - \sum_{n=0}^m \left( \frac{y_j^{(m+2)} h}{y_j^{(m+1)}} \right)^n \frac{1}{n!} \right), \quad y_j^{(m+2)} y_j^{(m+1)} \neq 0, \quad (21)$$

and the local truncation error (13) is reduced to

$$T_{j+1} = \sum_{s=0}^{\infty} \left( y_j^{(m+1)} \left( \frac{y_j^{(m+2)}}{y_j^{(m+1)}} \right)^{s+2} - y_j^{(m+3+s)} \right) \frac{h^{m+3+s}}{(m+3+s)!}, \quad (22)$$

that is the classical general form of Taylor-like  $m + 2$  order L-stable method in [11].

### 3. Extension to Vector Form

The method (12) can also be extended directly to systems of ODEs by using the new definitions of vector product and quotient defined in [9,12] as

$$b \cdot \left( \frac{a}{b} \right)^n = \begin{cases} \frac{\rho^n - \kappa^n}{\rho - \kappa} a + \frac{\rho \kappa^n - \kappa \rho^n}{\rho - \kappa} b, & \rho \neq \kappa \\ n \rho^{n-1} a - (n-1) \rho^n b, & \rho = \kappa \end{cases}, \quad n = 1, 2, \dots \quad (23)$$

$$b \cdot e^{a/b} = \begin{cases} \frac{e^\rho - e^\kappa}{\rho - \kappa} a + \frac{\rho e^\kappa - \kappa e^\rho}{\rho - \kappa} b, & \rho \neq \kappa \\ e^\rho a + (1 - \rho) e^\rho b, & \rho = \kappa \end{cases}, \quad (24)$$

where  $a, b \in C^m$  and  $\rho = \frac{1}{2} \left( \beta + \sqrt{\beta^2 + 4\alpha} \right)$ ,  $\kappa = \frac{1}{2} \left( \beta - \sqrt{\beta^2 + 4\alpha} \right)$ ,  $\alpha = -\frac{(a,a)}{(b,b)}$ ,  $\beta = 2 \frac{\text{Re}(a,b)}{(b,b)}$ .

The vector version of the proposed method is of the form

$$y_{j+1} = \sum_{n=0}^m y_j^{(n)} \frac{h^n}{n!} + y_j^{(k)} \cdot \left( \frac{y_j^{(k-1)}}{y_j^{(k)}} \right)^m \cdot \left( y_j^{(k-1)} \cdot e^{\frac{y_j^{(k)} h}{y_j^{(k-1)}}} - \sum_{n=0}^m y_j^{(k-1)} \cdot \left( \frac{y_j^{(k)} h}{y_j^{(k-1)}} \right)^n \frac{1}{n!} \right) \cdot \frac{y_j^{(m+1)}}{(y_j^{(k)})^2}, \quad (25)$$

where  $\|y_j^{(k)}\|_2 \times \|y_j^{(k-1)}\|_2 \neq 0$ .

The local truncation error of (25) is given by

$$T_{j+1} = \varepsilon + \sum_{s=0}^{\infty} \left( y_j^{(m+1)} \left( \frac{y_j^{(k)}}{y_j^{(k-1)}} \right)^{s+1} - y_j^{(m+2+s)} \right) \frac{h^{m+2+s}}{(m+2+s)!}, \quad (26)$$

where  $\varepsilon$  is a vector and  $\|\varepsilon\|_2 = O(h^{m+3})$ . The error in (26) is derived in a similar way as in [9,10]. The case  $k = m + 2$  can be found in [11].

It is clear that the new method can overcome the restrictions in the classical Taylor-like explicit method in its component applicable form (i.e.,  $y_j^{(m+2)} y_j^{(m+1)} \neq 0$ ) and in its vector applicable form (i.e.,  $\|y_j^{(m+2)}\| \times \|y_j^{(m+1)}\| \neq 0$ ). With the proposed method, the value of  $k$  can be adapted automatically to use any arbitrary pairs  $y_j^{(k)} y_j^{(k-1)} \neq 0$ , or  $\|y_j^{(k)}\| \times \|y_j^{(k-1)}\| \neq 0$ ,  $k = 1, 2, \dots, m + 1$ , while maintaining the order of convergence and the L-stability property.

#### 4. Numerical Results

In this section, we provide six numerical experiments to illustrate the theoretical results obtained in Sections 2 and 3. All numerical experiments are carried out using Matlab 8.3. The test problems were collected from the literature and the results are compared in the follow-up.

**Problem 1.** Consider the following SODE [8]:

$$y'(t) = -100y(t) + 99e^{2t}, \quad y(0) = 0. \quad (27)$$

The theoretical solution of (27) is given by

$$y(t) = \frac{33}{34} \left( e^{2t} - e^{-100t} \right). \quad (28)$$

Problem 1 is solved numerically using the generalised Taylor-like method (GTL) in its component form. The results are shown in Tables 1–3. Table 1 compares the solution error presented in [8] using the seventh order Sin and Explicit Taylor-like methods (STL7 and ETL7) with one given by the new method of seventh order (GTL7) having  $m = 6$  and  $k = 7$ . The results show that the GTL7 leads to a more accurate solution than the STL7 and ETL7. Table 2 lists the maximum solution error  $E_{\max} = \max_{0 \leq t_i \leq 0.5} |y(t_i) - y_i|$  using the time-step size  $h = 0.05$  at different values of  $m$  and  $k$ ,  $k \leq m + 1$ . The results show that as the order  $m + 1$ , or the value of  $k$ , increases, the solution error decreases. Moreover, increasing  $k$  is more effective than increasing  $m$  for improving the solution accuracy, and setting  $k = m + 1$  leads to more accurate results. Table 3 shows the computed order of convergence at different values of  $m$  and  $k$  using two different step sizes  $h = \{0.1, 0.05\}$ . The results confirm that the order of convergence depends only on the value of  $m$  for all values of  $k \leq m + 1$ .

Figure 1 shows the exact and numerical solutions of Problem 1 at  $h = 0.05$ . Figure 2 shows the behaviour of the solution error of different-order GTL component form at  $h = 0.05$ .

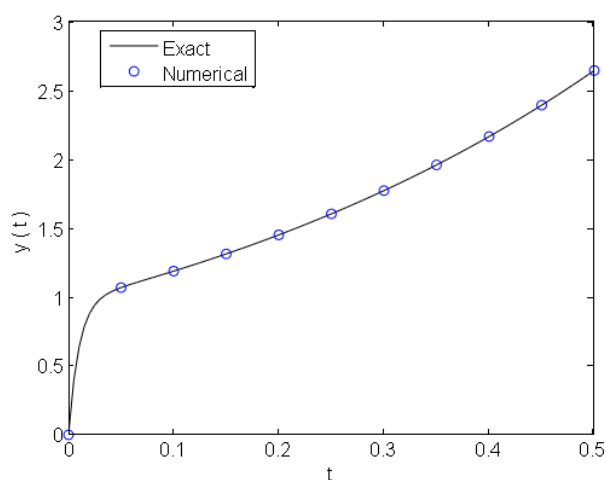


Figure 1. Exact and numerical solutions of Problem 1.

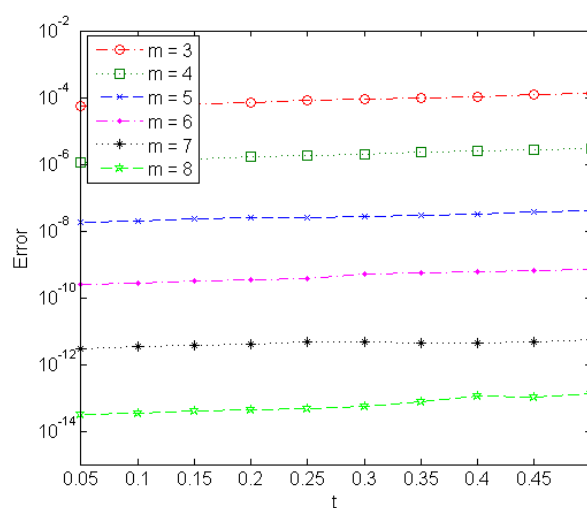


Figure 2. Solution error of different-order generalised Taylor-like method (GTL) in solving Problem 1 at  $h = 0.05$ .

Table 1. Comparison of the solution error for the STL7, ETL7 and GTL7 in the solution of Problem 1.

Error $ y(t_i) - y_i $						
$t_i$	$h = 0.025$			$h = 0.05$		
	STL7 [8]	ETL7 [8]	GTL7	STL7 [8]	ETL7 [8]	GTL7
0.05	$8.883 \times 10^{-2}$	$2.676 \times 10^{-10}$	$1.484 \times 10^{-12}$	$4.920 \times 10^{-1}$	$1.813 \times 10^{-8}$	$2.505 \times 10^{-10}$
0.10	$9.326 \times 10^{-3}$	$2.938 \times 10^{-10}$	$1.777 \times 10^{-12}$	$2.427 \times 10^{-1}$	$2.016 \times 10^{-8}$	$2.785 \times 10^{-10}$
0.15	$9.204 \times 10^{-4}$	$2.753 \times 10^{-10}$	$1.965 \times 10^{-12}$	$1.214 \times 10^{-1}$	$2.228 \times 10^{-8}$	$3.078 \times 10^{-10}$
0.20	$9.045 \times 10^{-5}$	$3.559 \times 10^{-10}$	$2.172 \times 10^{-12}$	$6.073 \times 10^{-2}$	$2.444 \times 10^{-8}$	$3.402 \times 10^{-10}$
0.25	$8.886 \times 10^{-6}$	$3.453 \times 10^{-11}$	$2.482 \times 10^{-12}$	$3.037 \times 10^{-2}$	$2.476 \times 10^{-8}$	$3.851 \times 10^{-10}$
0.30	$8.711 \times 10^{-7}$	$3.449 \times 10^{-11}$	$3.548 \times 10^{-13}$	$1.519 \times 10^{-2}$	$2.691 \times 10^{-8}$	$4.912 \times 10^{-10}$
0.35	$8.325 \times 10^{-8}$	$3.754 \times 10^{-11}$	$5.329 \times 10^{-15}$	$7.597 \times 10^{-3}$	$2.967 \times 10^{-8}$	$5.333 \times 10^{-10}$
0.40	$4.195 \times 10^{-9}$	$3.967 \times 10^{-11}$	$4.441 \times 10^{-16}$	$3.800 \times 10^{-3}$	$3.279 \times 10^{-8}$	$5.910 \times 10^{-10}$
0.45	$1.247 \times 10^{-10}$	$1.826 \times 10^{-11}$	$4.441 \times 10^{-16}$	$1.900 \times 10^{-3}$	$3.623 \times 10^{-8}$	$6.529 \times 10^{-10}$
0.50	$2.690 \times 10^{-1}$	$1.613 \times 10^{-11}$	$4.441 \times 10^{-16}$	$9.506 \times 10^{-4}$	$4.005 \times 10^{-8}$	$7.216 \times 10^{-10}$

**Table 2.** Maximum solution error for different values of  $m$  and  $k$ ,  $k \leq m + 1$ .

$k$	Error $E_{\max}$			
	$m = 3$	$m = 4$	$m = 5$	$m = 6$
2	$3.1315 \times 10^{-1}$	$3.6102 \times 10^{-1}$	$2.4102 \times 10^{-1}$	$2.1939 \times 10^{-1}$
3	$6.2574 \times 10^{-3}$	$7.2204 \times 10^{-3}$	$4.8204 \times 10^{-3}$	$4.3878 \times 10^{-3}$
4	$1.3071 \times 10^{-4}$	$1.4431 \times 10^{-4}$	$9.6410 \times 10^{-5}$	$8.7756 \times 10^{-5}$
5		$2.9867 \times 10^{-6}$	$1.9267 \times 10^{-6}$	$1.7551 \times 10^{-6}$
6			$4.0047 \times 10^{-8}$	$3.5082 \times 10^{-8}$
7				$7.2161 \times 10^{-10}$

**Table 3.** Computed order of convergence for different values of  $m$  and  $k$ ,  $k \leq m + 1$ .

$k$	Order of Convergence			
	$m = 3$	$m = 4$	$m = 5$	$m = 6$
2	3.7000	4.6003	5.9730	6.6588
3	3.6993	4.6003	5.9730	6.6588
4	3.7300	4.5997	5.9730	6.6588
5		4.6310	5.9727	6.6589
6			5.9898	6.6583
7				6.6845

**Problem 2.** Consider the oscillating SODE [17]:

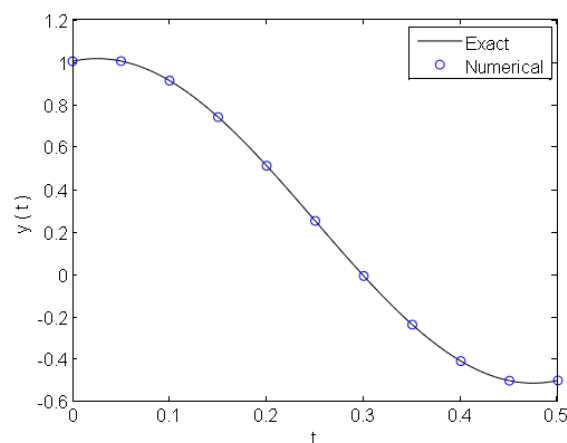
$$y'(t) = 1 - 2\pi \sin(2\pi t) - \frac{1}{\varepsilon} (y - t - \cos(2\pi t)), \quad y(0) = 1, \quad t \in [0, 0.5], \quad (29)$$

where  $\varepsilon$  is a parameter controlling the stiffness.

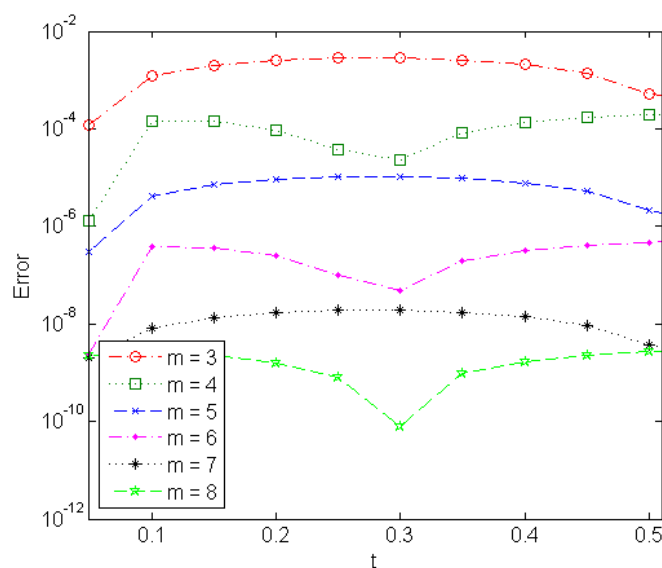
The theoretical solution is given by

$$y(t) = t + \cos(2\pi t). \quad (30)$$

From the theoretical point of view, we have  $y^{(k)}(0) = 0$ ,  $k = 3, 5, 7, \dots$ , which results in the computational overflow when adopting the high order classical ETL method. Problem 2 is solved numerically at  $\varepsilon = 1/200$  using the GTL component form with forth, fifth and sixth orders (i.e., the GTL4, GTL5, GTL6). The solution error, maximum solution error and the computed order of convergence are listed in Table 4. The results show that the GTL is not only stable and accurate but also overcomes the overflow in computation at  $t = 0$  by reducing the value of  $k$  from 6 to 2, without losing the convergence order and the L-stability property. Figure 3 shows the exact and numerical solutions at  $h = 0.05$ . Figure 4 depicts the solution error of different-order GTL component form at  $h = 0.05$ .

**Figure 3.** Exact and numerical solutions of Problem 2.





**Figure 4.** Solution error of different-order GTL in solving Problem 2 at  $h = 0.05$ .

**Table 4.** Numerical results for problem 2 using the component form of GTL4, GTL5 and GTL6.

Error $ y(t_i) - y_i $						
$t_i$	$h = 0.05$			$h = 0.025$		
	GTL4	GTL5	GTL6	GTL4	GTL5	GTL6
0.05	$1.1805 \times 10^{-4}$	$1.3329 \times 10^{-6}$	$2.9825 \times 10^{-7}$	$3.4591 \times 10^{-5}$	$2.2243 \times 10^{-7}$	$1.9292 \times 10^{-8}$
0.10	$1.2102 \times 10^{-3}$	$1.4098 \times 10^{-4}$	$4.2739 \times 10^{-6}$	$1.0607 \times 10^{-4}$	$5.0636 \times 10^{-6}$	$7.8031 \times 10^{-8}$
0.15	$1.9675 \times 10^{-3}$	$1.4222 \times 10^{-4}$	$7.0659 \times 10^{-6}$	$1.5377 \times 10^{-4}$	$4.6107 \times 10^{-6}$	$1.1578 \times 10^{-7}$
0.20	$2.5142 \times 10^{-3}$	$9.4632 \times 10^{-5}$	$9.1280 \times 10^{-6}$	$1.8598 \times 10^{-4}$	$2.6845 \times 10^{-6}$	$1.4094 \times 10^{-7}$
0.25	$2.8145 \times 10^{-3}$	$3.7845 \times 10^{-5}$	$1.0296 \times 10^{-5}$	$1.9994 \times 10^{-4}$	$5.0224 \times 10^{-7}$	$1.5220 \times 10^{-7}$
0.30	$2.8391 \times 10^{-3}$	$2.2603 \times 10^{-5}$	$1.0456 \times 10^{-5}$	$1.9429 \times 10^{-4}$	$1.8512 \times 10^{-6}$	$1.4851 \times 10^{-7}$
0.35	$2.5857 \times 10^{-3}$	$8.1111 \times 10^{-5}$	$9.5920 \times 10^{-6}$	$1.6959 \times 10^{-4}$	$3.8450 \times 10^{-6}$	$1.3023 \times 10^{-7}$
0.40	$2.0791 \times 10^{-3}$	$1.3136 \times 10^{-4}$	$7.7892 \times 10^{-6}$	$1.2825 \times 10^{-4}$	$5.5345 \times 10^{-6}$	$9.9157 \times 10^{-8}$
0.45	$1.3689 \times 10^{-3}$	$1.6884 \times 10^{-4}$	$5.2238 \times 10^{-6}$	$7.4287 \times 10^{-5}$	$6.6865 \times 10^{-6}$	$5.8272 \times 10^{-8}$
0.50	$5.2450 \times 10^{-4}$	$1.8981 \times 10^{-4}$	$2.1466 \times 10^{-6}$	$1.2760 \times 10^{-5}$	$7.1855 \times 10^{-6}$	$1.1294 \times 10^{-8}$
$E_{\max}$	$2.8391 \times 10^{-3}$	$1.8981 \times 10^{-4}$	$1.0456 \times 10^{-5}$	$1.9994 \times 10^{-4}$	$7.1855 \times 10^{-6}$	$1.5220 \times 10^{-7}$
Order	3.8278	4.7233	6.1019			

In fact, since the GTL overcomes the overflow in computations by reducing the value of  $k$ , the accuracy may be little decreased while the order of convergence remains constant. Being that the effect occurs more in the component than in the vector form, we conclude that the component form GTL is more suitable for scalar stiff problems than for stiff ODE systems where the GTL vector form can be applied.

**Problem 3.** Consider the following stiff ODE system [[7,10,16]]:

$$\begin{aligned} y_1' &= -20y_1 - 0.25y_2 - 19.75y_3, & y_1(0) &= 1 \\ y_2' &= 20y_1 - 20.25y_2 - 0.25y_3, & y_2(0) &= 0 \\ y_3' &= 20y_1 - 19.75y_2 - 0.25y_3, & y_3(0) &= -1. \end{aligned} \quad (31)$$

The theoretical solution is given by

$$\begin{aligned} y_1 &= 0.5 \left[ e^{-0.5t} + e^{-20t} (\sin(20t) + \cos(20t)) \right] \\ y_2 &= 0.5 \left[ e^{-0.5t} + e^{-20t} (\sin(20t) - \cos(20t)) \right] \\ y_3 &= 0.5 \left[ e^{-0.5t} - e^{-20t} (\sin(20t) + \cos(20t)) \right]. \end{aligned} \quad (32)$$

Due to the round-off error in computations, the ratio between  $y_j^6$  and  $y_j^5$ ,  $j = 2, 3$  may be very large at some grid points and the classical component form of the ETL6 [7,10] results in low accuracy, or even in an overflow, as shown in Table 5. On the other hand, the present component form of the GTL6 can overcome this drawback easily by reducing the value of  $k$  from 6 to 5 for  $h = \{0.01, 0.1\}$  at  $t = \{0.55, 0.3\}$ , respectively. Table 6 reveals that the vector forms of the ETL6 and GTL6 can overcome the overflow in computations with the component form of the ETL6 at  $h = 0.1$  and both of them have accurate results since there are no vectors with zero norms. Figure 5 depicts the exact and numerical solutions at  $h = 0.1$ . Figures 6 and 7 show the solution error of different-order vector form GTL at  $h = 0.1$  and  $h = 0.01$  respectively.

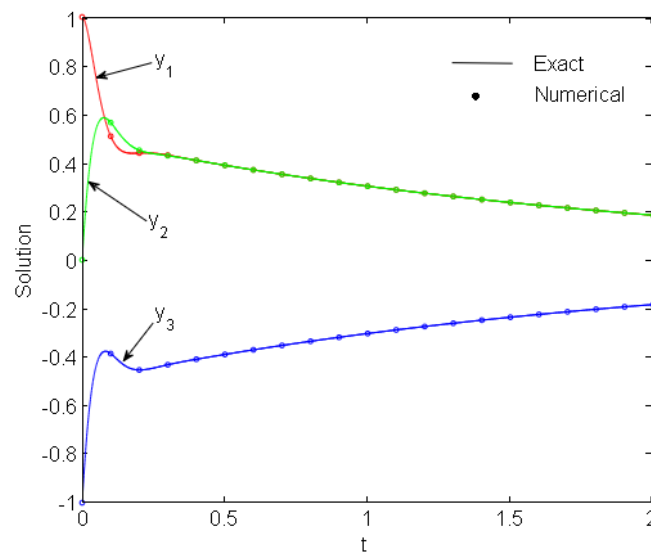


Figure 5. Exact and numerical solutions of Problem 3.

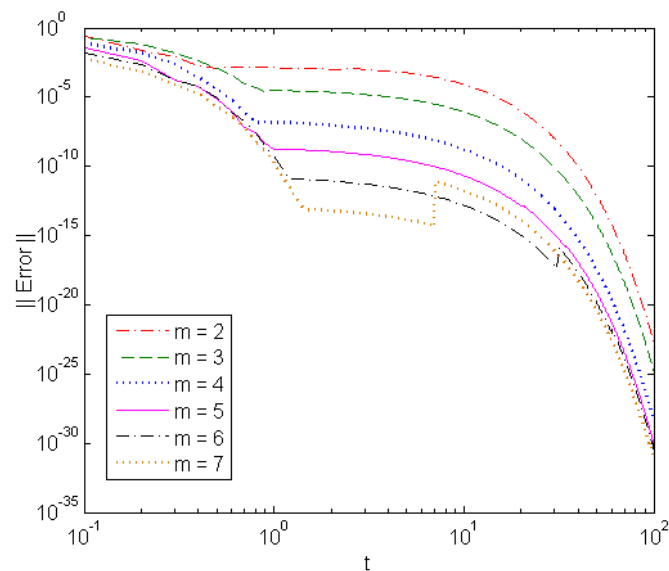


Figure 6. Solution error of different-order GTL in solving Problem 2 at  $h = 0.1$ .

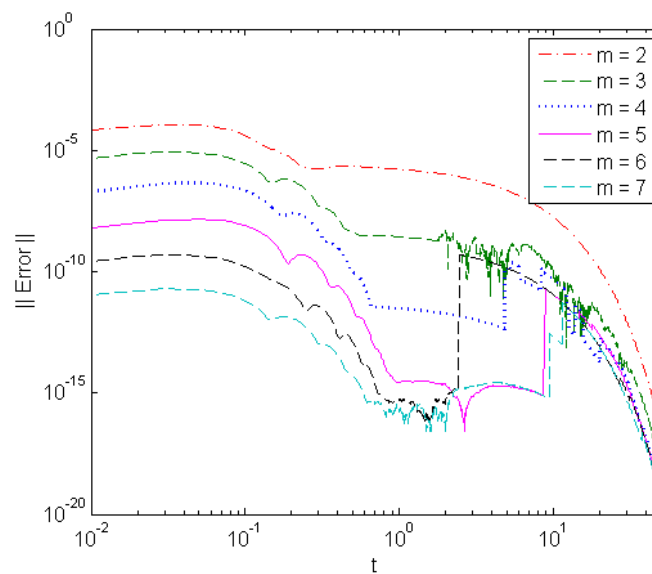


Figure 7. Solution error of different-order GTL in solving Problem 2 at  $h = 0.01$ .

Table 5. Numerical results for Problem 3 using the component forms of the ETL6 and GTL6.

$t$	$h$	$Y$	Theoretical Results	Numerical Results			
				ETL6 [7]		Present GTL6	
				Solution	Error	Solution	Error
50	0.01	$y_1$	$6.943971932482010 \times 10^{-12}$	$1.750801573149305 \times 10^{-10}$	$1.68 \times 10^{-10}$	$6.943971918388283 \times 10^{-12}$	$1.41 \times 10^{-20}$
		$y_2$	$6.943971932482010 \times 10^{-12}$	$1.750801573149305 \times 10^{-10}$	$1.68 \times 10^{-10}$	$6.943971918388283 \times 10^{-12}$	$1.41 \times 10^{-20}$
		$y_3$	$-6.943971932482010 \times 10^{-12}$	$-1.750801573149305 \times 10^{-10}$	$1.68 \times 10^{-10}$	$-6.943971918388284 \times 10^{-12}$	$1.41 \times 10^{-20}$
100	0.1	$y_1$	$9.643749239819589 \times 10^{-23}$	Overflow		$9.643749274555274 \times 10^{-23}$	$3.47 \times 10^{-31}$
		$y_2$	$9.643749239819589 \times 10^{-23}$			$9.643749274555277 \times 10^{-23}$	$3.47 \times 10^{-31}$
		$y_3$	$-9.643749239819589 \times 10^{-23}$			$-9.643749274555273 \times 10^{-23}$	$3.47 \times 10^{-31}$

Table 6. Numerical results for Problem 3 using the vector forms of the ETL6 and GTL6 methods.

$t$	$h$	$Y$	Theoretical Results	Numerical Results			
				ETL6 [10]		Present GTL6	
				Solution	Error	Solution	Error
100	0.1	$y_1$	$9.643749239819589 \times 10^{-23}$	$9.643749287321430 \times 10^{-23}$	$4.75 \times 10^{-31}$	$9.643749301944275 \times 10^{-23}$	$6.21 \times 10^{-31}$
		$y_2$	$9.643749239819589 \times 10^{-23}$	$9.643749287321430 \times 10^{-23}$	$4.75 \times 10^{-31}$	$9.643749301944272 \times 10^{-23}$	$6.21 \times 10^{-31}$
		$y_3$	$-9.643749239819589 \times 10^{-23}$	$-9.643749287321429 \times 10^{-23}$	$4.75 \times 10^{-31}$	$-9.643749301944270 \times 10^{-23}$	$6.21 \times 10^{-31}$

**Problem 4.** Consider the following stiff ODE system [2,13,15].

$$\begin{aligned} y_1' &= 2y_1 + y_2 + 2\sin(t), \quad y_1(0) = 2, \\ y_2' &= 998y_1 - 999y_2 + 999(\cos(t) - \sin(t)), \quad y_2(0) = 3. \end{aligned} \quad (33)$$

The theoretical solution is given by

$$\begin{aligned} y_1 &= 2e^{-t} + \sin(t) \\ y_2 &= 2e^{-t} + \cos(t). \end{aligned} \quad (34)$$

Problem 4 is solved using both the component and vector forms of the ETL6 and GTL6. The numerical results are listed in Tables 7 and 8. Table 7 shows that the component form GTL6 results in more accurate results than the ETL6 for  $h = 0.005$ . Furthermore, the GTL6 can overcome the overflow in computations for  $h = 0.001$  by reducing the value of  $k$  from 6 to 5 at  $t = 0.283$ . Table 8 shows that the vector form GTL6 leads to the same results as ETL6 for  $h = 0.01$  due to the absence of zero-vector norms. For  $h = 0.005$ , ETL6 exhibits an overflow while GTL6 overcomes this overflow by reducing the value of  $k$  from 6 to 5 at  $t = 76.93$ . Figure 8 shows the exact and numerical solutions at

$h = 0.1$ . Figures 9–11 illustrate the solution error of different-order vector form GTL at  $h = 0.001, 0.005$  and  $0.1$ , respectively.

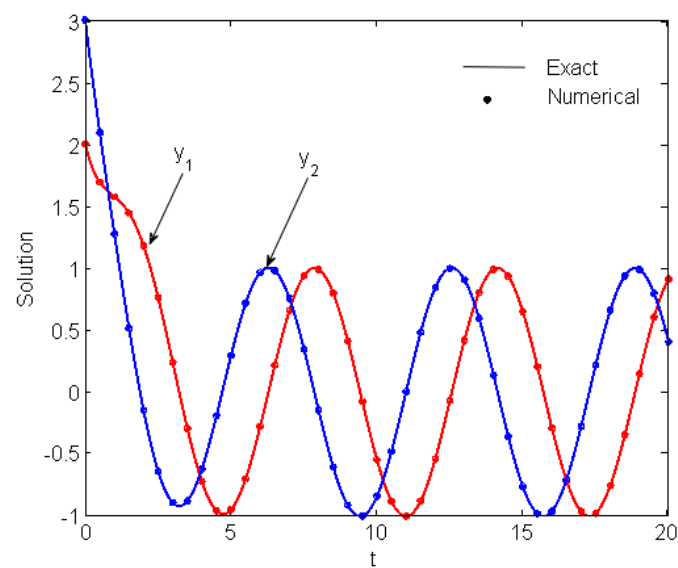


Figure 8. Exact and numerical solutions of Problem 4.

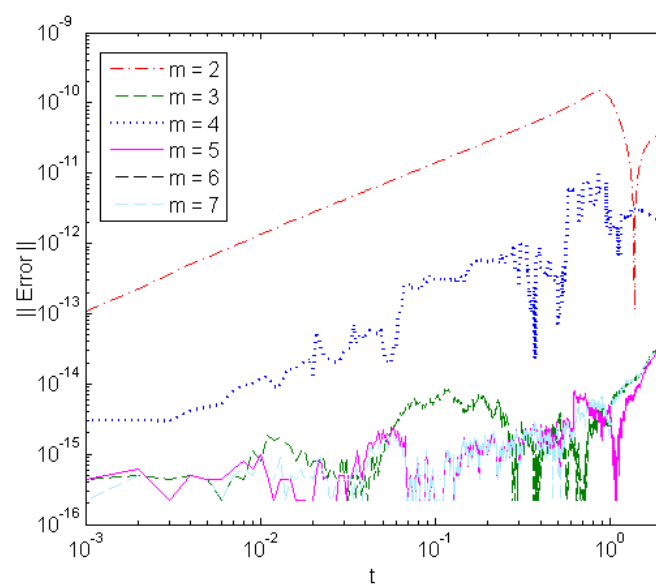


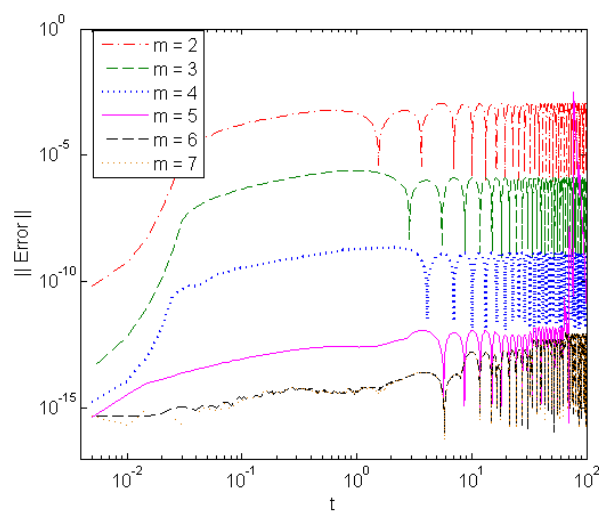
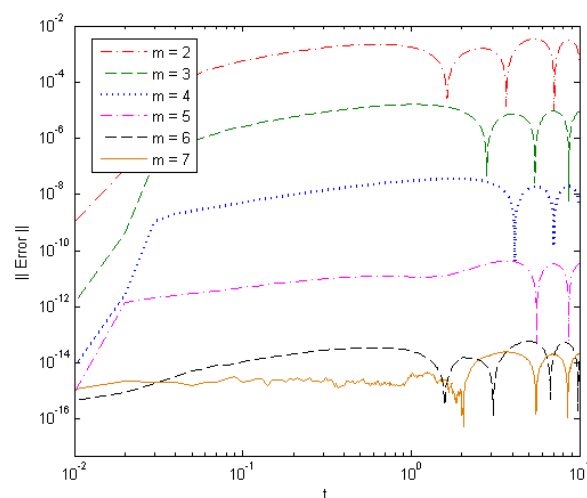
Figure 9. Solution error of different-order GTL in solving Problem 4 at  $h = 0.001$ .

Table 7. Numerical results for Problem 4 using the component form of the ETL6 and GTL6.

$t$	$h$	$Y$	Theoretical Results	Numerical Results			
				ETL6		GTL6	
				Solution	Error	Solution	Error
4	0.005	$y_1$	$-7.201712175304599 \times 10^{-1}$	$-7.285780200859474 \times 10^{-1}$	$8.41 \times 10^{-3}$	$-7.201712175725966 \times 10^{-1}$	$4.21 \times 10^{-11}$
		$y_2$	$-6.170123430861436 \times 10^{-1}$	$-6.254191456416346 \times 10^{-1}$	$8.41 \times 10^{-3}$	$-6.170123431282837 \times 10^{-1}$	$4.21 \times 10^{-11}$
2	0.001	$y_1$	1.179967993298907	Overflow		1.179967993029105	$2.70 \times 10^{-10}$
		$y_3$	$-1.454762700739170 \times 10^{-1}$			$-1.454762703437013 \times 10^{-1}$	$2.70 \times 10^{-10}$

**Table 8.** Numerical results for Problem 4 using the vector form of the ETL6 and GTL6 methods.

$t$	$h$	$Y$	Theoretical Results	Numerical Results			
				ETL6		GTL6	
				Solution	Error	Solution	Error
10	0.01	$y_1$	$-5.439303110298448 \times 10^{-1}$	$-5.439303110059717 \times 10^{-1}$	$2.39 \times 10^{-11}$	$-5.439303110059727 \times 10^{-1}$	$2.39 \times 10^{-11}$
		$y_2$	$-8.389807292169275 \times 10^{-1}$	$-8.389807291935291 \times 10^{-1}$	$2.34 \times 10^{-11}$	$-8.389807291935302 \times 10^{-1}$	$2.34 \times 10^{-11}$
100	0.005	$y_1$	$-5.063656411097588 \times 10^{-1}$	Overflow		$-5.063656411196315 \times 10^{-1}$	$9.87 \times 10^{-12}$
		$y_3$	$8.623188722876839 \times 10^{-1}$			$8.623188722778098 \times 10^{-1}$	$9.87 \times 10^{-12}$

**Figure 10.** Solution error of different-order GTL in solving Problem 4 at  $h = 0.005$ .**Figure 11.** Solution error of different-order GTL method in solving Problem 4 at  $h = 0.1$ .

**Problem 5.** Consider the following nonlinear stiff ODE system [10]:

$$\begin{aligned} y_1' &= -1002y_1 + 1000y_2^2, & y_1(0) &= 1, \\ y_2' &= y_1 - y_2(1 + y_2), & y_2(0) &= 1, \\ y_3' &= y_2 - 2y_3, & y_3(0) &= 1 - \frac{e^{0.01}}{64}. \end{aligned} \quad (35)$$

The theoretical solution is given by

$$\begin{aligned} y_1 &= e^{-2t}, \\ y_2 &= e^{-t}, \\ y_3 &= e^{-t} - \frac{e^{0.01-2t}}{64}. \end{aligned} \quad (36)$$

Tables 9 and 10 compare the results achieved by ETL6 [7,10] with those obtained by GTL6. From the theoretical point of view,  $y_3^{(6)} = 0$  at  $t = 0.01$  and so, as shown in Table 9, the component form ETL6 results in computational overflow, while THE GTL6 overcomes the overflow in computations at zero derivative component by changing the value of  $k$  from 6 to 5 at  $t = 0.01$ . Table 10 shows that both the ETL6 and GTL6 have the same accurate results in their vector forms due to the absence of zero-vector norms. Figure 12 shows the exact and numerical solutions at  $h = 0.01$ . Figure 13 depicts the solution error of different-order vector form GTL at  $h = 0.01$ .

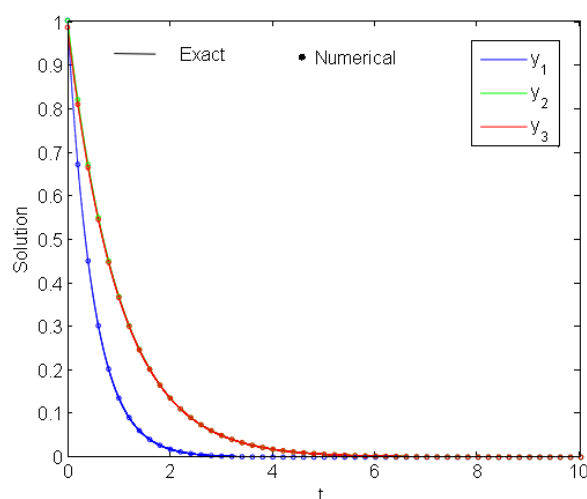


Figure 12. Exact and numerical solutions of Problem 5.

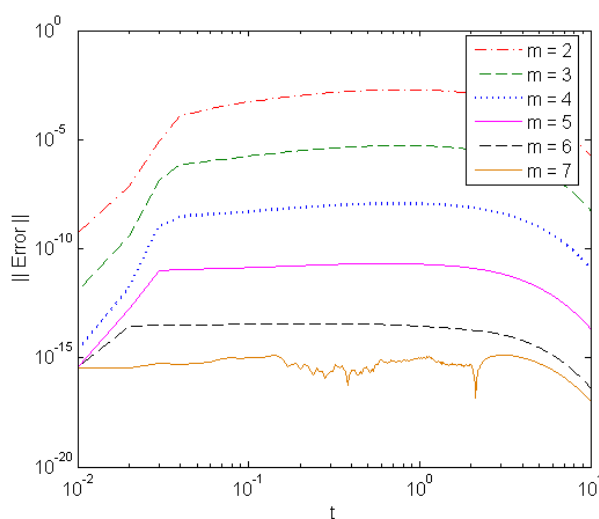


Figure 13. Solution error of different-order GTL in solving Problem 5 at  $h = 0.01$ .

Table 9. Numerical results for Problem 5 using the component forms of the ETL6 and GTL6 methods.

t	h	Y	Theoretical Results	Numerical Results			
				ETL6 [7]		GTL6	
				Solution	Error	Solution	Error
10	0.01	$y_1$	$2.061153622438558 \times 10^{-9}$	Overflow		$2.219610845565104 \times 10^{-9}$	$1.59 \times 10^{-10}$
		$y_2$	$4.539992976248485 \times 10^{-5}$			$4.711274610485264 \times 10^{-5}$	$1.71 \times 10^{-6}$
		$y_3$	$4.539989723328859 \times 10^{-5}$			$4.711104436407183 \times 10^{-5}$	$1.71 \times 10^{-6}$

**Table 10.** Numerical results for Problem 5 using the vector forms of the ETL6 and GTL6 methods.

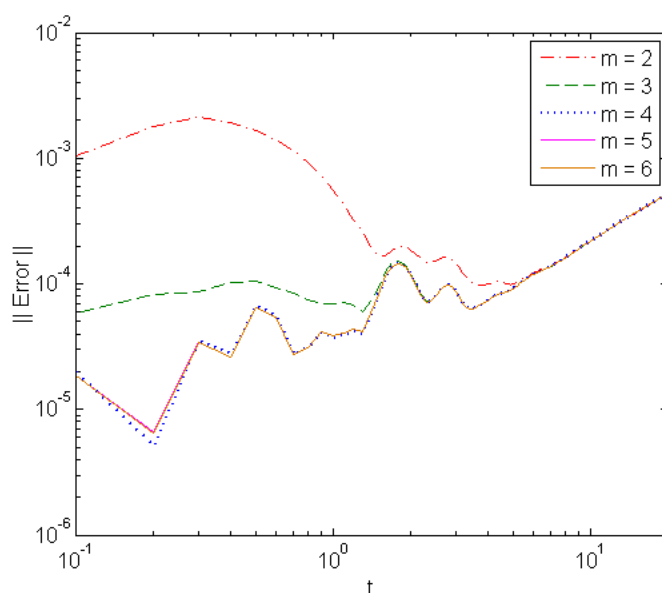
t	h	Y	Theoretical Results	Numerical Results			
				ETL6 [10]		GTL6	
				Solution	Error	Solution	Error
10	0.01	$y_1$	$2.061153622438558 \times 10^{-9}$	$2.061153624612472 \times 10^{-9}$	$2.17 \times 10^{-18}$	$2.061153624612640 \times 10^{-9}$	$2.17 \times 10^{-18}$
		$y_2$	$4.539992976248485 \times 10^{-5}$	$4.539992977820482 \times 10^{-5}$	$1.57 \times 10^{-14}$	$4.539992977820673 \times 10^{-5}$	$1.57 \times 10^{-14}$
		$y_3$	$4.539989723328859 \times 10^{-5}$	$4.539989724900820 \times 10^{-5}$	$1.57 \times 10^{-14}$	$4.539989724901008 \times 10^{-5}$	$1.57 \times 10^{-14}$

**Problem 6.** Consider the following nonlinear chemical reaction system involving eight reactants [1]:

$$\begin{aligned}
 y_1' &= -1.71y_1 + 0.43y_2 + 8.32y_3 + 0.0007, \\
 y_2' &= 1.71y_1 - 8.75y_2, \\
 y_3' &= -10.03y_3 + 0.43y_4 + 0.035y_5, \\
 y_4' &= 8.32y_2 + 1.71y_3 - 1.12y_4, \\
 y_5' &= -1.745y_5 + 0.43y_6 + 0.43y_7, \\
 y_6' &= -280y_6y_8 + 0.69y_4 + 1.71y_5 - 0.43y_6 + 0.69y_7, \\
 y_7' &= 280y_6y_8 - 1.81y_7, \\
 y_8' &= -y_7',
 \end{aligned} \tag{37}$$

with initial conditions  $y_1(0) = 1$ ,  $y_2(0) = y_3(0) = y_4(0) = y_5(0) = y_6(0) = y_7(0) = y_8(0) = 0$ .

This problem has no known analytical solution, and therefore, a numerical approximate solution using the built in BVP4C MATLAB solver [33–35] is taken as the reference solution for comparison. The problem is solved using different-order vector form GTL at  $h = 0.1$  and the solution error is represented in Figure 14 at  $h = 0.1$ . The same results are obtained using the ETL due to the absence of zero derivatives or zero vectors. Moreover, the exact and numerical solutions are illustrated in Figure 15.

**Figure 14.** Solution error of different-order GTL in solving Problem 6 at  $h = 0.1$ .

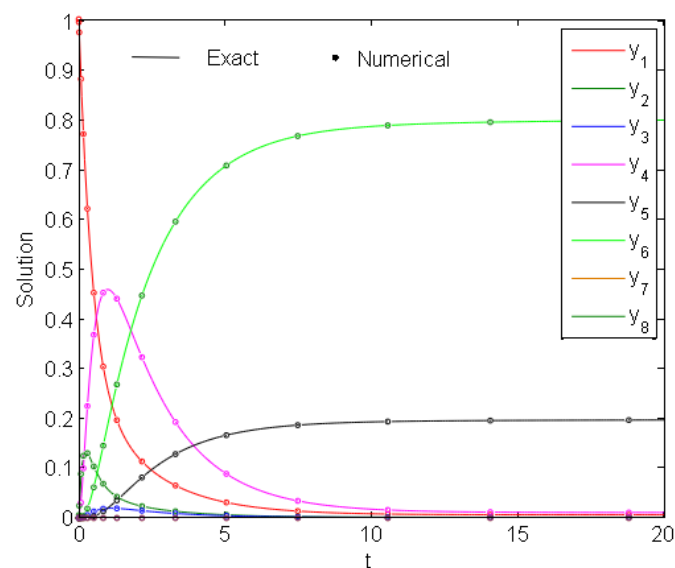


Figure 15. Exact and numerical solutions of Problem 6.

## 5. Conclusions

This paper presented a new generalised Taylor-like explicit method for SODEs and stiff ODE systems. The new method was formulated in its general component and vector applicable forms. The error and stability analysis of the method were discussed showing that it has at least  $(m + 1)$  order of convergence and the L-stability property. It was shown that several other integration schemes are essentially special cases of the proposed method. The algorithm is applied to six stiff test problems and the results are analyzed in several figures and tables and compared with those obtained by means of the classical Taylor-like methods. The results show that the new method is not only stable and accurate but also overcomes the shortcomings of the classical Taylor-like explicit methods in their component and vector forms by adapting the value of  $k$  during the computations without losing the convergence order and the L-stability property.

**Author Contributions:** Conceptualization, E.R.E.-Z. and J.T.M.; methodology, E.R.E.-Z.; software, E.R.E.-Z.; validation, E.R.E.-Z., J.T.M. and A.E.; formal analysis, E.R.E.-Z.; investigation, E.R.E.-Z.; resources, J.T.M.; data curation, A.E.; writing—original draft preparation, E.R.E.-Z.; writing—review and editing, E.R.E.-Z. and J.T.M.; visualization, J.T.M.; supervision, E.R.E.-Z. and J.T.M.; project administration, E.R.E.-Z.; funding acquisition, E.R.E.-Z.

**Funding:** This project was supported by the Deanship of Scientific Research at Prince Sattam Bin Abdulaziz University under the research project No. 2019/1/10067.

**Conflicts of Interest:** The author declares no conflict of interest.

## References

1. Hairer, E.; Wanner, G. *Solving Ordinary Differential Equations II*, second revised ed.; Springer: Berlin/Heidelberg, Germany, 2002.
2. Lambert, J.D. *Computational Methods in Ordinary Differential Equation*; John Wiley and Sons: New York, NY, USA, 1991.
3. Costa, R.P., Jr.; de Oliveira, M.L. Analysis of the performance of numerical integration methods for the tracking of ultra-high energy cosmic rays. *J. Comput. Phys.* **2019**, *392*, 432–443. [[CrossRef](#)]
4. Machado, J.A.; Galhano, A.M. Evaluation of manipulator direct dynamics using customized Rung-Kutta Methods. *Syst. Anal. Model. Simuln.* **1995**, *17*, 229–239.
5. Wu, X.Y. Exact exponentially fitted method for solving stiff ordinary differential equations. *J. Nanjing Univ. (Natural Sci.)* **1997**, *33*, 1–6.



6. Ebady, A.M.; Habib, H.M.; El-Zahar, E.R. A Fourth Order A-Stable Explicit One-Step Method for Solving Stiff Differential Systems Arising in Chemical Reactions. *Int. J. Pure Appl. Math.* **2012**, *81*, 803–812.
7. Wu, X.Y. A sixth-order A-stable explicit one-step method for stiff systems. *Comput. Math. Appl.* **1998**, *35*, 59–64. [[CrossRef](#)]
8. Ahmad, R.R.; Yaacob, N.; Mohd Murid, A.H. Explicit methods in solving stiff ordinary differential equations. *Int. J. Comput. Math.* **2004**, *81*, 1407–1415. [[CrossRef](#)]
9. Wu, X.Y.; Xia, J.L. Two low accuracy methods for stiff systems. *Appl. Math. Comput.* **2001**, *123*, 141–153. [[CrossRef](#)]
10. Wu, X.Y.; Xia, J.L. The vector form of a sixth-order A-stable explicit one-step method for stiff problems. *Comput. Math. Appl.* **2001**, *39*, 247–257. [[CrossRef](#)]
11. Wu, X.Y.; Xia, J.L. Study of general Taylor-like explicit methods in solving stiff ordinary differential equations. *Int. J. Comput. Math.* **2007**, *84*, 1795–1803. [[CrossRef](#)]
12. Wu, X.Y.; Xia, J.L. New vector forms of elemental functions with Taylor series. *Appl. Math. Comput.* **2003**, *141*, 307–312. [[CrossRef](#)]
13. Álvarez, J.; Rojo, J. An improved class of generalized Runge–Kutta methods for stiff problems. Part I: The scalar case. *Appl. Math. Comput.* **2002**, *130*, 537–560. [[CrossRef](#)]
14. Lepik, Ü.; Hein, H. *Haar Wavelets: With Applications*; Springer Science and Business Media: Berlin, Germany, 2014.
15. Yatim, S.A. M.; Ibrahim, Z.B.; Othman, K.I.; Suleiman, M.B. A numerical algorithm for solving stiff ordinary differential equations. *Math. Probl. Eng.* **2013**, *2013*. [[CrossRef](#)]
16. Wu, X.Y.; and Xia, J.L. An explicit two-step method exact for the scalar test equation  $y' = \lambda y$ . *Comput. Math. Appl.* **2000**, *39*, 249–257. [[CrossRef](#)]
17. Kushnir, D.; Rokhlin, V. A highly accurate solver for stiff ordinary differential equations. *SIAM J. Sci. Comput.* **2012**, *34*, A1296–A1315. [[CrossRef](#)]
18. Butusov, D.; Karimov, A.; Tutueva, A.; Kaplun, D.; Nepomuceno, E.G. The effects of Padé numerical integration in simulation of conservative chaotic systems. *Entropy* **2019**, *21*, 362. [[CrossRef](#)]
19. Miletics, E.; Molnarka, G. Taylor series method with numerical derivatives for initial value problems. *J. Comput. Methods Sci. Eng.* **2004**, *4*, 105–114. [[CrossRef](#)]
20. Jorba, A.; Zou, M. A software package for the numerical integration of ODEs by means of high-order Taylor methods. *Exp. Math.* **2005**, *14*, 99–117. [[CrossRef](#)]
21. Miletics, E.; Molnarka, G. Taylor Series Method with Numerical Derivatives for Numerical Solution of ODE Initial Values Problems. 2003. pp. 1–16. Available online: <http://heja.szif.hu/ANM/ANM-030110-B/anm030110b.pdf> (accessed on 2 October 2019).
22. Miletics, E.; Molnarka, G. Implicit extension of Taylor series method with numerical derivatives for initial value problems. *Comput. Math. Appl.* **2005**, *50*, 1167–1177. [[CrossRef](#)]
23. Schwarz, D.E.; Lamour, E. Projector based integration of DAEs with the Taylor series method using automatic differentiation. *J. Comput. Math. Appl.* **2014**, *262*, 62–72. [[CrossRef](#)]
24. El-Zahar, E.R. Piecewise approximate analytical solutions of high order singular perturbation problems with a discontinuous source term. *Int. J. Differ. Equ.* **2016**. [[CrossRef](#)]
25. El-Zahar, E.R.; Habib, H.M.; Rashidi, M.M.; El-Desoky, I.M. A Comparison of Explicit Semi-Analytical Numerical Methods for Solving Stiff ODE Systems. *Am. J. Appl. Sci.* **2015**, *12*, 304–320. [[CrossRef](#)]
26. Momani, S.; Odibat, Z. A novel method for nonlinear fractional partial differential equations: Combination of DTM and generalized Taylor’s formula. *J. Comput. Appl. Math.* **2008**, *220*, 85–95. [[CrossRef](#)]
27. Amodio, P.; Iavernaro, F.; Mazzia, F.; Mukhametzhonov, M.S.; Sergeyev Y.D. A generalized Taylor method of order three for the solution of initial value problems in standard and infinity floating-point arithmetic. *Math. Comput. Simul.* **2017**, *141*, 24–39. [[CrossRef](#)]
28. Toutounian, F.; Nasabzadeh, H. A new method based on generalized Taylor expansion for computing a series solution of the linear systems. *Appl. Math. Comput.* **2014**, *248*, 602–609. [[CrossRef](#)]
29. Sergeyev, Y.D. *Arithmetic of Infinity*, 2nd ed.; Edizioni Orizzonti Meridionali: Cosenza, Italy, 2003.
30. Sergeyev, Y.D. Infinity computer and calculus. *AIP Conf. Proc.* **2007**, *936*, 23–26.
31. Sergeyev, Y.D. Lagrange Lecture: Methodology of numerical computations with infinities and infinitesimals. *Rend. Semin. Mat. Univ. Politec. Torino* **2010**, *68*, 95–113.

32. Sergeyev, Y.D. Numerical computations with infinite and infinitesimal numbers: Theory and applications. In *Dynamics of Information Systems: Algorithmic Approaches*; Springer: New York, NY, USA, 2013; pp. 1–66.
33. Shampine, L.F.; Kierzenka, J.; Reichelt, M.W. Solving Boundary Value Problems for Ordinary Differential Equations in MATLAB with bvp4c. Tutorial Notes; 2000; pp. 1–27. Available online: [https://classes.engineering.wustl.edu/che512/bvp\\_paper.pdf](https://classes.engineering.wustl.edu/che512/bvp_paper.pdf) (accessed on 3 October 2019).
34. Shampine, L. (Southern Methodist University, Dallas, TX, USA). Solving a hard BVP with bvp4c. Private communication, 2004.
35. Gokhan, F.S.; Yilmaz, G. Numerical solution of Brillouin and Raman fiber amplifiers using bvp6c. *COMPEL Int. J. Comput. Math. Electr. Electron. Eng.* **2010**, *29*, 824–839. [[CrossRef](#)]



© 2019 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 (<http://creativecommons.org/licenses/by/4.0/>).