

Numerical Derivatives by Symbolic Tools in MATLAB

Ming-Gong Lee and Rei-Wei Song and Hsuang-Chi Chang*

mglee@chu.edu.tw*

Department of Applied Mathematics
Chung Hua University
Hsinchu, 30012 Taiwan

Abstract: Numerical approaches of ordinary differential equations (ODEs) usually require Jacobian evaluations and sometimes higher order derivatives of system of equations. The evaluation of these numerical derivatives can either use algebraic derivation or by the computer algebra system (CAS), such as MAPLE V, but these derivations may either cause debugging difficulty in programming or memory swell in computation. Finite difference formulas including forward difference, central difference, and backward difference are used to derive the Jacobian with different degree of accuracy. Symbolic evaluation capability of MATLAB is used to enhance fast derivation of numerical derivatives given in this paper. The accuracy of these numerical derivatives shows that the evaluation by symbolic computation from MATLAB is possible to be equipped into a numerical scheme of ordinary equations.

1. Introduction

The calculation of derivatives is necessary in the numerical solution of differential equations, such as the Jacobian matrix of Newton iterations in most of the ordinary equations solvers; and higher order coefficients in Taylor series methods, etc. A variety of software including symbolic programs [5] and automatic differentiation [1,2,4] have been introduced and used successfully in the application problems. In this paper, we will use the ideas given in [7] and to implement in the MATLAB environment [6]. We use the symbolic tool given in MATLAB and evaluate numerical derivatives of given functions with flexible order of accuracy. These numerical derivatives can further be used in the solutions of ODEs by Taylor series methods or other numerical schemes. Though, Miletics and Molnárka [7] have shown how to use numerical derivatives in a stiff ordinary equation by Taylor series methods, but we demonstrated how to implement with different combination to give accurate higher order derivatives.

In organization of this paper, Section 2 gives some notation of the differential equations, Section 3 gives some of the higher order numerical derivatives in matrix forms, Section 4 gives numerical results by implement Taylor series method to a stiff equation. Conclusion is given at Section 5.

2. Mathematical Formulations

Given an ordinary differential equation

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0, \quad x \in [x_0, x_0 + T], \quad (2.1)$$

where $y(x) = [y_1(x), y_2(x), \dots, y_n(x)]^T : \mathfrak{R} \rightarrow \mathfrak{R}^n$. Given p such that functions

$y_i(x) \in C^{p+1}([x_0, x_0 + T])$, for $i = 1, \dots, n$, and

$$f(x, y(x)) = [f_1(x, y(x)), f_2(x, y(x)), \dots, f_n(x, y(x))]^T.$$

The variable x may or may not be explicitly expressed in the right-hand side function. In the case that variable x is expressed explicitly in the Equation 2.1, we can define a new variable Y as

follows: Let $Y(x) = [x, y_1(x), y_2(x), \dots, y_n(x)] : \mathfrak{R} \rightarrow \mathfrak{R}^{n+1}$, then

$$Y'(x) = \frac{dY(x)}{dx} = [1, f_1(Y(x)), \dots, f_n(Y(x))]^T = F(Y(x)), \quad (2.2)$$

where $Y_0(x) = x, Y_1(x) = y_1, \dots, Y_n(x) = y_n$.

Assume $f_i(x, y(x)), i = 1, \dots, n$, is a $(p+1)$ times differentiable function, then the classical Taylor series expression [3] at $x_0 + s$ about x_0 can be written as,

$$y(x_0 + s) = y(x_0) + y'(x_0)s + \frac{1}{2!}y''(x_0)s^2 + \frac{1}{3!}y'''(x_0)s^3 + \dots + \frac{1}{p!}s^p y^{(p)}(x_0 + s) + \frac{1}{(p+1)!}s^{p+1}y^{(p+1)}(x_0 \oplus s), \quad (2.3)$$

function $y^{(p+1)}(x_0 \oplus s)$ is evaluated at some unknown point between x_0 and $x_0 \oplus s$.

3. Numerical Derivative by Finite Difference

When implementing Taylor series method, some derivatives of right hand side function, i.e., Eq. 2.2 must be evaluated. In this section, we will describe how to use finite difference to express higher order derivatives of Eq. 2.2.

3.1 Finite Difference Expressions of Second Order Derivatives

From Eq. 2.1 or Eq. 2.2, a second order derivative of y can be expressed as:

$$y''(x) = f'(x) = [f_1'(Y(x)), \dots, f_n'(Y(x))]^T = [(\frac{\partial f_1(Y(x))}{\partial Y} \cdot \frac{dY(x)}{dx}), \dots, (\frac{\partial f_n(Y(x))}{\partial Y} \cdot \frac{dY(x)}{dx})]^T = \frac{\partial f}{\partial Y} F. \quad (3.1.1)$$

Since the first component of function F is 1, the higher order derivatives of the first component of this function are all zero. As a result, we will only use the notation of Eq. 2.1 for simplicity. By implementing forward differencing, backward differencing, and central differencing to approximate the partial derivatives, $\frac{\partial f_k(x)}{\partial Y}$, for $k = 1, 2, \dots, n$, various formulas can be obtained. Matrix A is

obtained by forward difference, matrix C is obtained by central difference, matrix D is obtained by an extrapolation formula, and these partial derivatives are evaluated at $x = x_0$.

$$A = \frac{1}{h} \{ [[f_1(x+h, y_1(x), \dots, y_n(x)), \dots, f_n(x+h, y_1(x), \dots, y_n(x))] - [f_1(x, y_1(x), \dots, y_n(x)), \dots, f_n(x, y_1(x), \dots, y_n(x))]]^T, \dots, [[f_1(x, y_1(x), \dots, y_n(x)+h), \dots, f_n(x, y_1(x), \dots, y_n(x)+h)] - [f_1(x, y_1(x), \dots, y_n(x)), \dots, f_n(x, y_1(x), \dots, y_n(x))]]^T \} \quad (3.1.2)$$

$$C = \frac{1}{2h} \{ [[f_1(x_0+h, y_1(x_0), \dots, y_n(x_0)), \dots, f_n(x_0+h, y_1(x_0), \dots, y_n(x_0))]^T - [f_1(x_0-h, y_1(x_0), \dots, y_n(x_0)), \dots, f_n(x_0-h, y_1(x_0), \dots, y_n(x_0))]^T, \dots, [[f_1(x_0, y_1(x_0), \dots, y_n(x_0)+h), \dots, f_n(x_0, y_1(x_0)+h, \dots, y_n(x_0))]^T - [f_1(x_0, y_1(x_0), \dots, y_n(x_0)-h), \dots, f_n(x_0, y_1(x_0), \dots, y_n(x_0)-h)]^T \} \quad (3.1.3)$$

$$\begin{aligned}
D = & \frac{1}{6h} \{ [f_1(x-2h, y_1(x), \dots, y_n(x)), \dots, f_n(x-2h, y_1(x), \dots, y_n(x))] - \\
& - 6[f_1(x-h, y_1(x), \dots, y_n(x)), \dots, f_n(x-h, y_1(x), \dots, y_n(x))] + \\
& + 3[f_1(x, y_1(x), \dots, y_n(x)), \dots, f_n(x, y_1(x), \dots, y_n(x))] + \\
& + 2[f_1(x+h, y_1(x), \dots, y_n(x)), \dots, f_n(x+h, y_1(x), \dots, y_n(x))] \}, \dots, \\
& [f_1(x, y_1(x), \dots, y_n(x)-2h), \dots, f_n(x, y_1(x), \dots, y_n(x)-2h)] - \\
& - 6[f_1(x, y_1(x), \dots, y_n(x)-h), \dots, f_n(x-h, y_1(x), \dots, y_n(x)-h)] + \\
& + 3[f_1(x, y_1(x), \dots, y_n(x)), \dots, f_n(x, y_1(x), \dots, y_n(x))] + \\
& + 2[f_1(x, y_1(x), \dots, y_n(x)+h), \dots, f_n(x, y_1(x), \dots, y_n(x)+h)] \}
\end{aligned} \tag{3.1.4}$$

Because of the complex formulation of the expressions given at Equations 3.1.2-3.1.4, we use notations given at [7] to simply, let

$$F^{k,i}(x_0, y(x_0)) = [f_1(x_0, y_1(x_0), \dots, y_k(x_0) + ih, \dots), \dots, f_n(x_0, y_1(x_0), \dots, y_k(x_0) + ih, \dots)]^T, \tag{3.1.5}$$

$k = 0, \dots, n, i = 0, \pm 1, \pm 2, \dots, \pm n$, when $k=0, y_0(x) = x$.

By this notation, we can rewrite Eqs. 3.1.2~3.1.4 as follows:

$$A = \frac{1}{h} [(F^{0,1} - F^{0,0}), (F^{1,1} - F^{1,0}), \dots, (F^{n,1} - F^{n,0})], \tag{3.1.6}$$

$$C = \frac{1}{h} [(F^{0,0} - F^{0,-1}), (F^{1,0} - F^{1,-1}), \dots, (F^{n,0} - F^{n,-1})], \tag{3.1.7}$$

$$\begin{aligned}
D = & \frac{1}{6h} [(F^{0,-2} - 6F^{0,-1} + 3F^{0,0} + 2F^{0,1}), \dots, \\
& (F^{n,-2} - 6F^{n,-1} + 3F^{n,0} + 2F^{n,1})].
\end{aligned} \tag{3.1.8}$$

The following matrix E is obtained by a special type of finite difference, and is given as,

$$\begin{aligned}
E = & \frac{1}{12h} [(F^{0,-2} - 8F^{0,-1} + 8F^{0,1} - F^{0,2}), \dots, \\
& (F^{n,-2} - 8F^{n,-1} + 8F^{n,1} - F^{n,2})].
\end{aligned} \tag{3.1.9}$$

3.2 Third Order Derivatives

As in Section 3.1, we can derive third order derivative of variable y or the second derivative of Eq. 2.1 as:

$$Y''' = F'' = \left(\frac{\partial f}{\partial Y} F \right)' = \left(\frac{\partial^2 f}{\partial Y^2} FF \right) + \frac{\partial f}{\partial Y} \left(\frac{\partial f}{\partial Y} F \right). \tag{3.2.1}$$

Similarly we also define a new function so as to simplify notations of the derived derivative functions :

$$F^{kl,ij} = f(\dots, y_k + ih, \dots, y_l + jh, \dots), \quad k, l = 0, 1, \dots, n, \quad i, j = 0, \pm 1, \pm 2, \dots, \tag{3.2.2}$$

$$F^{kk,ii} = f(\dots, y_k + ih, \dots), \quad k, l = 0, 1, \dots, n, \quad i = 0, \pm 1, \pm 2, \dots$$

Higher order derivatives were obtained similarly as the second order derivatives. The following matrix G_m is formulated by first applying the central difference scheme to have a second order

derivative function, and then applying a forward difference scheme to have a third order derivative function. The induced formula is given as:

$$G_m = \frac{1}{2h^2} [(F^{m0,10} - F^{m0,-10} - F^{m0,1-1} + F^{m0,-1-1}), \dots, \\ 2(F^{mm,11} - 2F^{mm,00} + F^{mm,-1-1}), \dots, \\ (F^{mn,10} - F^{mn,-10} - F^{mn,1-1} + F^{mn,-1-1})], m = 0, 1, \dots, n. \quad (3.2.3)$$

Another third order derivative formula H_m can be obtained by implementing central difference approximation consecutively as:

$$H_m = \frac{1}{4h^2} [(F^{m0,11} - F^{m0,-11} - F^{m0,1-1} + F^{m0,-1-1}), \dots, \\ 4(F^{mm,11} - 2F^{mm,00} + F^{mm,-1-1}), \dots, \\ (F^{mn,11} - F^{mn,-11} - F^{mn,1-1} + F^{mn,-1-1})], m = 1, 2, \dots, n \quad (3.2.4)$$

By using formulas G_m or H_m , the matrix of $\frac{\partial^2 f}{\partial Y^2}$ can be approximated as:

$$[G_1, G_2, \dots, G_n]^T \approx [H_1, H_2, \dots, H_n]^T \approx \frac{\partial^2 f}{\partial Y^2}. \quad (3.2.5)$$

3.3 The fourth Order Derivative

By Eq. 3.2.1, the fourth derivative of function y or the third order derivative matrix of Eq. 2.1 can be written as:

$$Y^{IV} = F''' = \left(\frac{\partial^2 f}{\partial Y^2} FF\right)' + \left(\frac{\partial f}{\partial Y} \left(\frac{\partial f}{\partial Y} F\right)\right)' = \frac{\partial^3 f}{\partial Y^3} FFF + \frac{\partial^2 f}{\partial Y^2} \left(\frac{\partial f}{\partial Y} F\right)F + \\ + 2\left(\frac{\partial^2 f}{\partial Y^2} F \left(\frac{\partial f}{\partial Y} F\right)\right) + \frac{\partial f}{\partial Y} \left(\frac{\partial^2 f}{\partial Y^2} FF\right) + \left(\frac{\partial f}{\partial Y}\right)^3 F \quad (3.3.1)$$

In Eq. 3.3.1, $\frac{\partial^3 f}{\partial Y^3}$ is derived similarly as in section 3.2 by lower order derivative functions, and its

formula can be written as $\frac{\partial^3 f}{\partial Y_i \partial Y_j \partial Y_k}$, $i, j, k, l = 1, 2, \dots, n$.

Similarly, we also define new functions to simplify the notation [7],

$$f_k^{ijl,mez}(Y) = f_k(\dots, y_i + mh, \dots, y_j + eh, \dots, y_l + zh, \dots), \\ f_k^{iil,mnz}(Y) = f_k(\dots, y_i + mh, \dots, y_l + zh, \dots), \\ f_k^{iii,mmm}(Y) = f_k(\dots, y_i + mh, \dots), k, l, i, j = 0, 1, \dots, n, m, e, z = 0, \pm 1, \pm 2 \dots \quad (3.3.2)$$

When the partial derivatives were implemented with three different directions, a formula is given as:

$$\begin{aligned}
T_{ij} = & \frac{1}{h^3} [(F^{ij0,000} - F^{ij0,001} - F^{ij0,0-11} + F^{ij0,1-10} - \\
& - F^{ij0,1-11} + F^{ij0,101} - F^{ij0,100})], \dots, \\
& \frac{1}{h^3} [(F^{ijn,000} - F^{ijn,001} - \\
& - F^{ijn,0-11} + F^{ijn,1-10} - F^{ijn,1-11} + F^{ijn,101} - F^{ijn,100})], \quad i, j = 0, 1, \dots, n
\end{aligned} \tag{3.3.3}$$

If the partial derivatives were implemented with respect to two same directions, a formula is given as:

$$\begin{aligned}
(T_{ij})_j = & \frac{1}{h^3} [2F^{ijj,000} - F^{ijj,0-1-1} - F^{ijj,011} - 2F^{ijj,100} + F^{ijj,1-1-1} + F^{ijj,111}], \\
& i, j = 0, 1, \dots, n.
\end{aligned} \tag{3.3.4}$$

If the partial derivative were implemented with respect to only one direction, a formula is given as:

$$(T_{jj})_j = \frac{1}{2h^3} [-2F^{jjj,111} + F^{jjj,222} + 2F^{jjj,-1-1-1} - F^{jjj,-2-2-2}], \quad j = 0, 1, \dots, n. \tag{3.3.5}$$

As for these derivatives matrices for $\frac{\partial^3 f}{\partial Y^3}$, it can be obtained by combing different formulas, such as forward and central difference and the four-point formula E. We omit the procedures here.

4. Numerical Derivatives and Its Accuracy

In this section we will implement numerical derivatives by finite difference given in Sec. 3 in a Taylor series method to a stiff ordinary equation. In Section 4.1, we will show how to acquire these numerical derivatives in a flexible way, i.e., we can calculate with different combination from the central differencing, forward differencing, and other formulas to obtain more accurate derivatives. Some errors between exact derivatives and numerical derivatives are given to show their accuracy. In Section 4.2, we will compare the numerical solutions obtained by implementing numerical derivative and by exact derivatives to a stiff ordinary differential equation.

4.1 Accuracy of the Numerical Derivatives

The second order derivatives was given at Equation 3.1.1, the numerical derivative can be obtained by applying Eqs. 3.1.6~3.1.9. For the third order numerical derivative of Equation 2.1, it will be given as follows. During the implementation, a central difference is first used to get the second order derivative, and a forward difference scheme is applied further to get the third order derivatives. The order of applying finite difference schemes is not the critical issue, since we can still change freely with these ordering to get accurate numerical derivatives. The above procedure was given only to demonstrate one possibility of our procedure. As a result, for the above procedure, the third order numerical derivative can be written as:

$$\begin{aligned}
G1_m = & \frac{1}{2h^2} [(F^{m0,11} - F^{m0,1-1} - F^{m0,01} + F^{m0,0-1}), \dots, \\
& F^{mm,22} - F^{mm,00} - F^{mm,11} + F^{mm,-1-1}, \dots, \\
& (F^{mn,11} - F^{mn,1-1} - F^{mn,01} + F^{mn,0-1})], \quad m = 1, 2, \dots, n.
\end{aligned} \tag{4.1.1}$$

There is a minor difference between $G1_m$ and G_m which is given at Eq. 3.2.3. Matrix $G1_m$ is used in the implementation of this paper. Similarly, in the fourth derivative of Equation 2.1, we use central differencing scheme and backward differencing scheme for easy implementation in programming, a formula is given when three directions of partial differentiation are all different and is given as,

$$T1_{ij} = \frac{1}{2h^3} [(F^{ij0,111} - F^{ij0,110} - F^{ij0,1-11} + F^{ij0,1-10} - F^{ij0,011} + F^{ij0,010} + F^{ij0,0-11} - F^{ij0,010}), \dots, (F^{ijn,111} - F^{ijn,110} - F^{ijn,1-11} + F^{ijn,1-10} - F^{ijn,011} + F^{ijn,010} + F^{ijn,0-11} - F^{ijn,010})], \quad (4.1.2)$$

$i, j = 0, 1, \dots, n$.

When there are two same directions of partial differentiation and a formula is given as,

$$(T1_{ij})_j = \frac{1}{2h^3} [F^{ijj,122} - F^{ijj,011} - F^{ijj,100} + F^{ijj,1-1-1} - F^{ijj,022} + F^{ijj,011} + F^{ijj,000} - F^{ijj,00-1}], \quad (4.1.3)$$

When the directions of partial differentiation are all the same, a formula is given as,

$$(T1_{jj})_j = \frac{1}{2h^3} [F^{jjj,333} - 2F^{jjj,222} + 2F^{jjj,000} - F^{jjj,-1-1-1}]. \quad (4.1.4)$$

Equations 4.1.2~4.1.4 are the formulas for $\frac{\partial^3 f}{\partial Y^3}$ shown in the fourth derivatives. These formulas can be implemented with the second order formulas A, B, C, D and E, in this paper we only adapt matrices A and C for convenience.

A stiff differential equation [6] is given to test the accuracy of the numerical derivatives of different orders. We will compare the accuracy of numerical derivatives of right hand side function with the exact derivatives.

$$\begin{aligned} \frac{dy_1(x)}{dt} &= \frac{1}{\alpha} (y_1(x) + y_2(x) - y_1(x)y_2(x) - qy_1^2(x)) \\ \frac{dy_2(x)}{dt} &= 2my_3(x) - y_2(x) - y_1(x)y_2(x) \\ \frac{dy_3(x)}{dt} &= \frac{1}{\gamma} (y_1(x) - y_3(x)), \quad y_1(0) = a, y_2(0) = b, y_3(0) = d, \end{aligned} \quad (4.1.5)$$

where α, m, q , and γ are parameters, a, b and d are initial values. Let $\alpha = 0.1, m=0.5, \gamma = 1, q=0.01, a=5, b=7$, and $d=8$. Evaluate Eqs. 3.1.6~3.1.9, the following error estimates of the first derivative are estimated and are given at Table 1. Matrices A, C, D, and E are all have good accuracy.

Table 1. When $h = 0.05$, error of matrices A, C, D, E

f_i	$\max_{0 \leq j \leq n} \left \frac{\partial f}{\partial Y_j} - A_j \right $	$\max_{0 \leq j \leq n} \left \frac{\partial f}{\partial Y_j} - C_j \right $	$\max_{0 \leq j \leq n} \left \frac{\partial f}{\partial Y_j} - D_j \right $	$\max_{0 \leq j \leq n} \left \frac{\partial f}{\partial Y_j} - E_j \right $
-------	---	---	---	---

f_1	0.005	$4.97379915 \times 10^{-14}$	$5.61328761 \times 10^{-13}$	$5.6843418 \times 10^{-14}$
f_2	$2.84217094 \times 10^{-14}$	$1.42108547 \times 10^{-14}$	$3.73034936 \times 10^{-14}$	$6.3948846 \times 10^{-14}$
f_3	0	0	0	0

Evaluate matrices of Eqs.4.1.1, the following error estimates of the second order derivative are estimated and are given at Table 2. It shows that matrix $G1$ has good accuracy.

Table 2. When $h = 0.05$, error in calculating $G1$

$\max_{0 \leq j, k \leq n} \left(\left \frac{\partial^2 f_1}{\partial Y_j \partial Y_k} - (G1_1)_{jk} \right \right)$	$\max_{0 \leq j, k \leq n} \left(\left \frac{\partial^2 f_2}{\partial Y_j \partial Y_k} - (G1_2)_{jk} \right \right)$	$\max_{0 \leq j, k \leq n} \left(\left \frac{\partial^2 f_3}{\partial Y_j \partial Y_k} - (G1_3)_{jk} \right \right)$
$9.09494701 \times 10^{-13}$	$5.11590769 \times 10^{-13}$	0

Evaluate matrices of Eqs. 4.1.2~4.1.4, the following error estimates of the third order derivatives are estimated and are given at Table 3. . It shows that matrix $T1$ has good accuracy.

Table 3. When $h = 0.05$, error in calculating $T1$

$\max_{0 \leq i, j, k \leq n} \left(\left \frac{\partial^3 f_1}{\partial Y_i \partial Y_k \partial Y_j} - (T1_{ij})_{k1} \right \right)$	$\max_{0 \leq i, j, k \leq n} \left(\left \frac{\partial^3 f_2}{\partial Y_i \partial Y_k \partial Y_j} - (T1_{ij})_{k2} \right \right)$	$\max_{0 \leq i, j, k \leq n} \left(\left \frac{\partial^3 f_3}{\partial Y_i \partial Y_k \partial Y_j} - (T1_{ij})_{k3} \right \right)$
$4.54747350 \times 10^{-10}$	$5.68434188 \times 10^{-11}$	0

These results show that the accuracy of the numerical derivatives all satisfies the theoretical results given in [7].

4.2 The Numerical Solution by Taylor Series Method

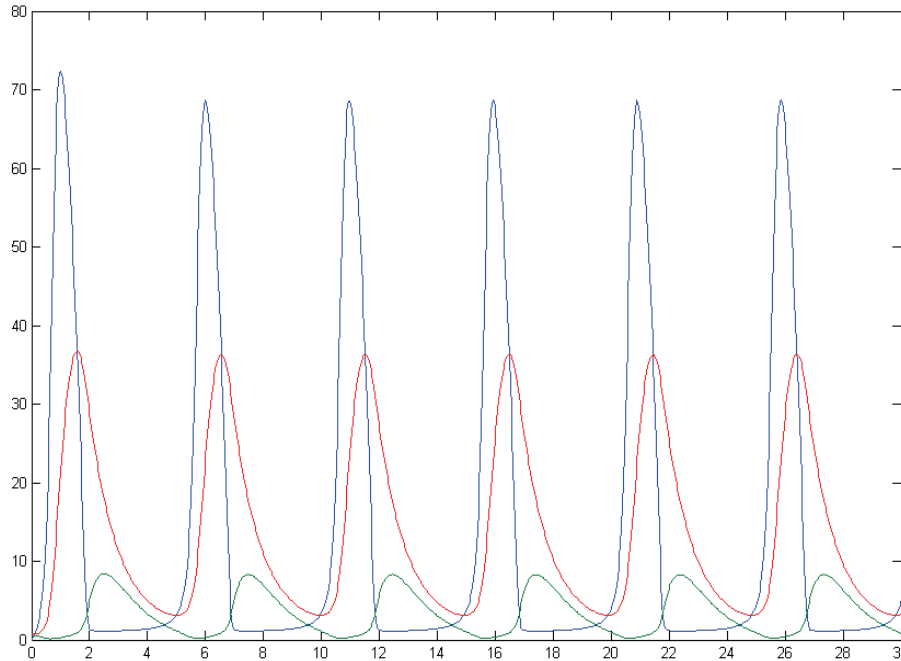


Figure. 1 Numerical Solutions of Equation 4.1.5 given by Taylor Series Method with Numerical Derivatives

Following the same algorithm given at [7], let the error control $\varepsilon = 0.01$ and the interval of integration is $[0, 30]$. The numerical solutions of Taylor series method by implementing numerical derivatives given in Sec. 4 are shown in Figure 1 above. The traces are solutions of three components of the system of Equation 4.1.5.

Table 4. Numerical Values by Taylor Series Method with Three ways to Obtain Derivatives

Subinterval	numerical values	Taylor Series Method		
		MATLAB	MAPLE	Exact Derivative
[2.523273367, 2.560628954]	\hat{y}_1	1.134198723	1.134196674	1.134199594
	\hat{y}_2	8.392300277	8.392300074	8.392299952
	\hat{y}_3	16.98898275	16.98898277	16.98898276
[2.560628954, 2.598028225]	\hat{y}_1	1.134864277	1.134859481	1.134864562
	\hat{y}_2	8.348665503	8.348665016	8.348664582
	\hat{y}_3	16.40695669	16.40695674	16.40695672
[2.598028225, 2.635578062]	\hat{y}_1	1.135874152	1.135876124	1.135872603
	\hat{y}_2	8.287319921	8.393618957	8.287318165
	\hat{y}_3	15.84411177	15.84344461	15.84411182

During the integration, we also examine the numerical solutions obtained by applying numerical derivatives and exact derivatives in several subintervals to show that these numerical derivatives do behave similarly with the solutions obtained by exact derivatives. These numerical values even exact at least up to the fourth digits and are given at Table 4. In the following, we examine the numerical solutions by applying Taylor series method and by subroutine given in MATLAB, the error between these solutions match well and are given at Table 5.

Table 5. Numerical Values by Taylor Series Method with Three ways to Obtain Derivatives (Compared with solutions given by implementing Matlab)

Subinterval		Solutions Error between MATLAB and Taylor Series Methods
		$\ y(x_k + s) - \hat{y}(x_k + s)\ $
[2.523273367, 2.560628954]	y_1	8.710000001244111e-007
	y_2	3.250000002452680e-007
	y_3	9.999997274690031e-009
[2.560628954, 2.598028225]	y_1	2.850000000442776e-007
	y_2	9.209999998205376e-007
	y_3	2.999999892949745e-008
[2.598028225, 2.635578062]	y_1	1.549000000045098e-006
	y_2	1.755999999630831e-006
	y_3	5.000000058430487e-008

From Tables 4 and 5, at several subintervals of integration, numerical solutions obtained by numerical derivative given in this paper and the exact derivative both match well. It means that the numerical derivatives derived from applying finite difference schemes can be applied during numerical solutions of differential equation even it is stiff.

5. Conclusion

In this paper, we have demonstrated how to calculate numerical derivatives with finite difference scheme and use the numeric to a Taylor series method to solve a stiff ordinary equation. These numerical derivatives also have good accuracy compared with exact derivatives. Though alternative way of finding derivatives are also possible, by results from this paper, numerical derivatives by finite differencing is still capable of rendering derivatives with good accuracy. Taylor series method may not be suitable for fast computational aspect, it requires many higher orders derivatives, but a possibility of parallel computation may be a resolution for the speed concern. In addition, some results show that the Taylor series method is not suitable for very stiff problems, since it is basically an explicit method. In order to solve this stability requirement, an implicit type of Taylor series method may be approached in the future.

References

- [1] Bishof Christian, Carle Alan, Havland Paul, Khademi Peyvand, and Mauer Andrew, *ADIFOR 2.0 User's Guide (Revision D)*, Technical Report ANL/MCS-TM-192, Mathematics and Computer Science Division, Argonne National Laboratory, USA, (1994).
- [2] Griewank Andreas, *Evaluating Derivatives-Principles and Techniques of Algorithmic Differentiation*, SIAM, Philadelphia, (2000).
- [3] Kendall E. Atkinson, *Introduction to Numerical Analysis*, John Wiley & Sons Ltd., New York (1978).
- [4] Lee, M.G., *Application of Automatic Differentiation in Numerical Solution of a Flexible Mechanism*, ICCMSE 2003 International Conference of Computational Methods in Sciences and Engineering, Kastoria, Greece, pp350-359.
- [5] MAPLE V, User Manual, The Waterloo Maple Inc., Canada (1997)
- [6] MATLAB, User Manual, The Mathworks Inc., USA (2006).
- [7] Miletics, E. and Molnárka G., *Taylor Series Method with Numerical Derivatives for Initial Value Problems*, Journal of Computational Methods in Sciences and Engineering ,vol. 3 ,no .3 (2003), pp. 319-329.