Taylor Series Method with Numerical Derivatives for Numerical Solution of ODE Initial Value Problems

E. Miletics
G. Molnárka
Department of Mathematics, Széchenyi István University, Győr
miletics@sze.hu
molnarka@sze.hu

HU ISSN 1418-7108: HEJ Manuscript no.: ANM-030110-B

Abstract

The Taylor series method is one of the earliest analytic-numeric algorithms for approximate solution of initial value problems for ordinary differential equations. Currently this algorithm is not applied frequently. This is because when one solves systems of ordinary differential equations, calculating the higher order derivatives formally is an over-elaborate task, this is true even if one uses the computer algebraic systems such as MATHEMATICA or MAPLEV. The other reason is that only the explicit versions of this algorithm are known. The main idea of the rehabilitation of these algorithms is based on the approximate calculation of higher derivatives using well-known technique for the partial differential equations. In some cases such algorithms will be much more complicated than a R-K methods, because it will require more function evaluation than well-known classical algorithms. However these evaluations can be accomplished fully parallel and the coefficients of truncated Taylor series can be calculated with matrix-vector operations. For large systems these operations suit for the parallel computers. These algorithms have several advantageous properties over the widely used classical methods. The approximate solution is given as a piecewise polynomial function defined on the subintervals of the whole interval and the local error of this solution at the interior points of the subinterval is less than that one at the end point. This property offers different facility for adaptive error control. We remark that for the explicit Taylor series methods is possible to give its implicit extension. Using the fact, that the approximate solution is a continuous function, (in the case of implicit version it is a continuously differentiable function), adaptive examination and control of some qualitative properties of algorithms will be more simple than the case when the approximate solution are given only at discrete grid points. This paper describes several above-mentioned algorithms and examines its consistency and stability properties. It demonstrates some numerical test results for systems of equations herewith we attempt to prove the efficiency of these new-old algorithms.

1 Introduction

The Taylor series algorithm is one of the earliest algorithms for the approximate solution for initial value problems for ordinary differential equations. Newton used it in his calculation [1] and Euler describe it in his work [2]. Since then one can find many mentions of it such as J. Liouville [3], G. Peano [4], E. Picard [5]. Many authors have furter developed this algorithm, see for example A. Gibbons [12] and R. E. Moore [13]. The basic idea of these developments was the recursive calculation of the coefficients of the Taylor series.

Modern numerical algorithms for the solution of ordinary differential equations are also based on the method of the Taylor series. Each algorithm, such as the Runge-Kutta or the multistep methods are constructed so that they give an expression depending on a parameter (h) called step size as an approximate solution and the first terms of the Taylor series of this expression must be identical with the terms of the Taylor series of the exact solution. These are the consistency and the order conditions for the algorithms. These expressions potentially can be evaluated at any value of the parameter (h), but practically the evaluation is realized only at grid points. Such algorithms give the value of the approximate solution at grid points. For such algorithm main cost is the number of the function evaluation. (Remark: some algorithm contains more than one parameter.) These algorithms differ from others in its order, stability properties and its cost of realization. The overview of the modern algorithms on can find in the monograph of E. Harier, S. P. Norsett and G. Wanner [6] - [7]. A possible implicit extension of the Taylor series algorithm is given in [9]. Actually, in the qualification of the algorithm become important their quality properties such as conservativity, pozitivity preserving [8], monotonity preserving, detecting and following the bifurcation etc.

By appearing the parallel computers as the complexity of algorithm new cost functions must be defined because in this case the main goal is to minimize of the execution time and not the number of function evaluations. From this point of view several variant of the method of the Taylor series could be an effective algorithm.

One family of such algorithm can be derived from the classical method of the Taylor series by approximating the derivatives in Taylor coefficients with numerical derivatives.

1.1 Formulation of the problem

The problems to be solved are as follows:

$$y^{'}(x) = f(x, y(x)), \quad y(x_0) = y_0, \tag{1}$$

where $x \in [x_{0,}x_0 + T]$, $\quad y(x) = [y_1(x), y_2(x), ..., y_n(x)] : R \to R^n$,
$y_i(x) \in C^{p+1}([x_0, x_0 + T])$, $\quad i = 1, ..., n$ for a given p and
$f(x, y(x)) = [f_1(x, y(x)), f_2(x, y(x)), ..., f_n(x, y(x))]^T$.
Let us introduce the following notations:

$$Y(x) = [x, y_1(x), ..., y_n(x)]^T, \quad F(Y(x)) = [1, f_1(Y(x)), ..., f_n(Y(x))]^T.$$

By this notation the equation (1) is as follows:

$$Y(x) = F(Y(x)), \quad Y(x_0) = [x_0, y_1(x_0), y_2(x_0), ..., y_n(x_0)]^T. \tag{2}$$

Assume, that the functions $f_i(x,(y(x)), \quad i = 1,...,n$ are (p+1)-times continuously differentiable functions with respect x. Then the method of the Taylor series is based on the following Taylor's series expansion of the solution. Using our compact notation the Taylor series of the solution of (2) is:

$$\begin{gathered} Y(x_0+s) = Y(x_0) + Y^{'}(x_0)s + \frac{1}{2!}Y^{''}(x_0)s^2 + \\ +\frac{1}{3!}Y^{'''}(x_0)s^3 + ... + \frac{1}{(p+1)!}Y^{(p+1)}(x_0+\Theta s), \end{gathered} \tag{3}$$

where Θ could be different for the components and the value of p might be chosen depending on the smoothness of the right hand side in (1) and the desired order of the method to be obtained.

2 Numerical approximation of the derivatives

The main idea of the construction if the method of Taylor algorithm with numerical derivatives is the numerical approximation of the derivatives $Y^{(k)}(x_0)$,
$k = 2,3,4,...$. The first derivatives are given by the right hand side of (1), so its approximation is only an evaluation

2.1 Numerical approximation of the second derivatives

The second derivative of the right hand side of (2) called its Jacobian is:

$$\begin{gathered} Y^{''}(x_0) = F^{'}(x_0) = [0, f_1^{'}(Y(x_0)),..., f_n^{'}(Y(x_0))]^T = \\ = [0, (\frac{\partial f_1(x_0)}{\partial Y}, F(x_0)),..., (\frac{\partial f_n(x_0)}{\partial Y}, F(x_0))]^T = \frac{\partial F}{\partial Y}F, \end{gathered} \tag{4}$$

where the derivatives $\frac{\partial f_k(x_0)}{\partial Y}, \quad k = 1,2,...,n$ are to be approximated by numerical derivatives.
Let us introduce the following notations:

$$\begin{gathered} F^{k,i}(Y(x_0)) = [1, f_1(..., y_k(x_0)+ih,...),..., f_n(..., y_k(x_0)+ih,...)]^T, \\ k = 1,...,n, \quad i = 0, \pm 1, \pm 2,... \end{gathered} \tag{5}$$

where $k = 0$ means the first coordinate of the vector Y which is simply x and $h > 0$ is a given small value.

The calculation of the values $F^{k,i}(Y(x_0))$ means the evaluation of the right hand side of (2) at the given points. Using the finite difference formulae to approximate the first derivatives we get that the following A, B, C, and D matrices approximate the generalized Jacobi matrix of the right hand side of (2) with different degree of accuracy.
Let us define the A, B, C and D matrices by the following formulae:

$$A = \frac{1}{h}\left[\left(F^{0,1} - F^{0,0}\right), \left(F^{1,1} - F^{1,0}\right), ..., \left(F^{n,1} - F^{n,0}\right)\right], \tag{6}$$

$$B = \frac{1}{2h}\left[\left(F^{0,1} - F^{0,-1}\right), \left(F^{1,1} - F^{1,-1}\right), ..., \left(F^{n,1} - F^{n,-1}\right)\right], \tag{7}$$

$$C = \frac{1}{6h}[\left(F^{0,-2} - 6F^{0,-1} + 3F^{0,0} + 2F^{0,1}\right), ..., \\ \left(F^{n,-2} - 6F^{n,-1} + 3F^{n,0} + 2F^{n,1}\right)], \tag{8}$$

$$D = \frac{1}{12h}[\left(F^{0,-2} - 8F^{0,-1} + 8F^{0,1} - F^{0,2}\right), ..., \\ \left(F^{n,-2} - 8F^{n,-1} + 8F^{n,1} - F^{n,2}\right)]. \tag{9}$$

Lemma 1 *(The order of accuracy of approximation.) Assume that every components of the function F is five-times continuously differentiable with respect to all its variables in some neighbourhood of x and $y_i(x), \quad i = 1, 2, ..., n$. Then the matrix A approximates the generalized Jacobi matrix $\frac{\partial F}{\partial Y}$ by first order degree of accuracy while the matrix B by second order degree of accuracy, the matrix C by third order degree of accuracy and the matrix D by fourth order degree of accuracy at the point x in the following sense*

$$\max_{0\le i,j\le n}(|\frac{\partial F_i}{\partial Y_j} - A_{ij}|) \le c_1 h, \quad \max_{0\le i,j\le n}(|\frac{\partial F_i}{\partial Y_j} - B_{ij}|) \le c_2 h^2, ..., \\ \max_{0\le i,j\le n}(|\frac{\partial F_i}{\partial Y_j} - D_{ij}|) \le c_4 h^4,$$

where $c_1, c_2, ..., c_4$ are constants not depending on the parameter h.

Proof. We only outline for the matrix A. For the other B, C and D matrices the proof is similar. The expressions $\frac{\partial F_i}{\partial Y_j}$ denote the partial derivatives

$$\frac{\partial f_i(x, y_1, y_2, ..., y_n)}{\partial y_j}, \quad i, j = 1, 2, ..., n \quad \textit{and} \\ A_{ij} = \frac{1}{h}(f_i(..., y_j + h, ...) - f_i(..., y_j, ...)).$$

From this by the (6) we get that $\quad A_{ij} = \frac{\partial f_i(x,y_1,y_2,...,y_n)}{\partial y_j} + c_{ij} h, \ i, j = 1, 2, ..., n.$
Now we choose $\max_{0\le i,j\le n} |c_{ij}| = c_1$, which gives the appropriate constant for the first inequality. ■

Remark 2 *The different accuracy order approximations of the generalized Jacobi matrix we need for construction of the different accuracy order numerical algorithm but for higher order methods we have to approximate the higher order derivatives of the right hand side of (2).*

2.2 Numerical approximation of the third derivatives

The third derivative of the right hand side of (2) is the coefficient of the third term in the Taylor' expansion (3) is:

$$Y^{'''} = F^{''} = (\frac{\partial F}{\partial Y} F) = (\frac{\partial^2 F}{\partial Y^2} FF) + \frac{\partial F}{\partial Y}(\frac{\partial F}{\partial Y} F). \tag{10}$$

The second term in (10) can be approximated by using the matrices A, B, C and D. Our task now is the construction of the approximation formulae for the first term. So we have to give the approximate formulae for the bilinear form $\frac{\partial^2 F}{\partial Y^2}$.
Let us introduce the following notations:

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

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

Let us define the matrices G_i and H_i as follows:

$$\begin{aligned} G_i = \frac{1}{2h^2}[(&F^{i0,10} - F^{i0,-10} - F^{i0,1-1} + F^{i0,-1-1}), ..., \\ &2(F^{ii,11} - 2F^{ii,11} + F^{ii,-1-1}), ..., \\ (F^{i0,10} - F^{i0,-10} - &F^{i0,1-1} + F^{i0,-1-1})], \quad i = 1, 2, ..., n, \end{aligned} \tag{11}$$

$$\begin{aligned} H_i = \frac{1}{4h^2}[(&F^{i0,11} - F^{i0,-11} - F^{i0,1-1} + F^{i0,-1-1}), ..., \\ &4(F^{ii,11} - 2F^{ii,00} + F^{ii,-1-1}), ..., \\ (F^{in,11} - F^{in,-11} - &F^{in,1-1} + F^{in,-1-1})], \quad i = 1, 2, ...n. \end{aligned} \tag{12}$$

Lemma 3 *(The order of the approximation of the third order derivatives.) Assume, that the conditions of the Lemma 1. are fulfilled. Then the matrices G_i, $i = 1, 2, ..., n$ approximate the generalized bilinear formula $\frac{\partial^2 F}{\partial Y^2}$ by first order degree of accuracy while the matrix H_i, $i = 1, 2, ..., n$ by second order degree of accuracy at the point x in the following sense:*

$$(G_i)_{kl} \approx \frac{\partial^2 f_k}{\partial Y_i \partial Y_l}, \; i, k, l = 1, 2, ..., n \text{ and } \max_{0 \le i,k,l \le n} |(G_i)_{kl} - \frac{\partial^2 f_k}{\partial Y_i \partial Y_l}| \le c_2 h, \tag{13}$$

where $c_2 > 0$ is independent of h.

Proof. (schema) As we have seen before the elements $(G_i)_{kl}(Y)$ approximate the derivatives $\frac{\partial^2 f_k(Y)}{\partial Y_i \partial Y_l}$.

So describing the Taylor series of $(G_i)_{kl}(Y)$ around the point Y we get: $(G_i)_{kl} = \frac{\partial^2 f_k(Y)}{\partial Y_i \partial Y_l} + c_{ikl} h$, where the values c_{ikl} are constants consisting from the sum of values of third derivatives of f_k taken from the h distance neighbourhood of Y. The appropriate value for the Lemma 2 is $c_1 = \max_{i,k,l} \|c_{ikl}\|$. ■

2.3 Numerical approximation of the fourth derivatives

The forth derivative of the solution vector Y is the third derivative of the right hand side of (2). So we get:

$$\begin{aligned} Y^{IV} = F^{'''} = (\frac{\partial^2 F}{\partial Y^2} F F)^{'} + ((\frac{\partial F}{\partial Y})^2 F)^{'} = \frac{\partial^3 F}{\partial Y^3} F F F + \frac{\partial^2 F}{\partial Y^2} (\frac{\partial F}{\partial Y} F) F + \\ +2(\frac{\partial^2 F}{\partial Y^2} F (\frac{\partial F}{\partial Y} F)) + \frac{\partial F}{\partial Y} (\frac{\partial^2 F}{\partial Y^2} F F) + (\frac{\partial F}{\partial Y})^3 F. \end{aligned} \tag{14}$$

The derivatives in three last term of (14) can be approximated by the matrices A, B, C, D, G_i and H_i. Therefore we have to approximate the third derivative $\frac{\partial^3 F}{\partial Y^3}$ which is a trilinear formula consisting of the elements $\frac{\partial f_l}{\partial Y_i \partial Y_j \partial Y_k}$, $i, j, k, l = 1, 2, ..., n$. In order to approximate these terms by finite difference method let us introduce the following notations:

$$f_k^{ijl,mnp}(Y) = f_k(..., Y_i + mh, ..., Y_j + nh, ..., Y_l + ph),$$

but with the convention that in the case of repeated indices is no multiple shift as it is described by the next expression:

$$f_k^{iil,mmp}(Y) = f_k(..., Y_i + mh, ..., Y_l + ph, ...) \;\; or \;\; f_k^{iii,mmm}(..., Y_i + mh, ...).$$

Using these notations the approximation of the trilinear formula in (15) can be built up by the following matrices:

$$\begin{gathered} T_{ij} = \frac{1}{h^3}[(F^{ij0,000} - F^{ij0,001} - F^{ij0,0-10} + F^{ij0,0-11} + F^{ij0,1-10} - \\ -F^{ij0,1-11} + F^{ij0,101} - F^{ij0,100})], ..., \frac{1}{h^3}[(F^{ijn,000} - F^{ijn,001} - \\ -F^{ijn,0-10} + F^{ijn,0-11} + F^{ijn,1-10} - F^{ijn,1-11} + \\ +F^{ijn,101} - F^{ijn,100})], \quad i, j = 0, 1, ...n, \end{gathered} \tag{15}$$

but in the case of two repeated indices for example the term $(T_{ij})_j$ takes the form as follows:

$$(T_{ij})_j = \frac{1}{h^3}(2F^{ijj,000} - F^{ijj,0-1-1} - F^{ijj,011} - 2F^{ijj,100} + F^{ijj,1-1-1} + F^{ijj,111}),$$

and for three repeated indices we get:

$$(T_{jj})_j = \frac{1}{2h^3}(-2F^{jjj,111} + F^{jjj,222} + 2F^{jjj,-1-1-1} - F^{jjj,-2-2-2}).$$

We remark, that this formula approximate the appropriate terms of derivatives by second order accuracy.

Lemma 4 *Let the function F the right hand side of (2) be four times continuously differentiable with respect Y. Then the trilinear form $\frac{\partial^3 F}{\partial Y^3}$ can be approximated with first order accuracy by the bilinear forms T_{ij}, $ij = 1, 2, ..., n$ and $\max\limits_{0 \le i,j,k,l \le n} |(T_{ij})_{kl} - \frac{\partial^3 F}{\partial Y^3}| \le c_3 h$, where c_3 is not dependent on h.*

Proof. (schema) The elements $(T_{ij})_{kl}$ approximate the derivative $\frac{\partial^3 f_l}{\partial Y_i \partial Y_j \partial Y_k}$. So, describing the Taylor's series expansion of $(T_{ij})_{kl}$ at the neighbourhood of Y we get: $(T_{ij})_{kl} Y = \frac{\partial^3 f_l}{\partial Y_i \partial Y_j \partial Y_k} + c_{ijkl} h$, where constants c_{ijkl} consist of some values of fourth order derivatives of f_l from the h radius neighbourhood of. The maximum of absolute value of them will be the appropriate constant for c_3. ■

3 The methods of the Taylor series based on numerical derivatives

By summarizing the results explained above we can construct some truncation of Taylor's series of the Y solution of (2) as an approximate solution at a given subinterval. The basic idea of the construction is that the expressions (4), (10), (14) of the derivatives in the truncation of Taylor's expansion (3) are replaced by its approximations which will be constructed by some combinations of the matrices (6)-(9), (11), (12), (15). This rich collection of approximate matrices allows us to construct a big choice of explicit methods of Taylor's series such as first-, second-, third- and fourth order methods. The first order method is the well known and fully examined Euler algorithm. Therefore we start with the second order algorithm.

3.1 Methods of second order accuracy

Theorem 5 *Let us assume, that the third derivatives of the solution of (1) are bounded in the following sense: there exists such constant $c_1 > 0$ that $\left\|\frac{\partial^2 F}{\partial Y^2}\right\| \leq c_1$ and let h and s be parameters satisfying the condition $h \leq s$. Then the following expression gives a second order explicit algorithm*

$$\hat{Y}(x_k + s) = Y(x_k) + F(x_k)s + \frac{1}{2}A(x_k)F(x_k)s^2. \tag{16}$$

Another possible second order algorithms can be constructed by replacing matrix $A(x_n)$ in (16) by the matrices $B(x_k)$, $C(x_k)$, or $D(x_k)$ defined by the expressions (7)-(9).

Proof. We have to prove, that the polynomial on the right hand side of (21) up to second order is equal with the Taylor's expansion given in (3). So if s is small enough then $\left\|Y(x_i + s) - \hat{Y}(x_i + s)\right\| \leq cs^3$, where $c > 0$ does not depend on s. By using the Lemma 1. we get:

$$\left\|Y(x_k + s) - \hat{Y}(x_k + s)\right\| \leqq$$
$$\leqq \left\|\frac{1}{2!}Y''(x_k)s^2 + \frac{1}{3!}Y''(x_k)s^3 + O(s^4) - \frac{1}{2!}\frac{\partial F}{\partial Y}(x_k)F(x_k)s^2 + O(s^2 h)\right\| \leqq$$
$$\leqq \left\|\frac{1}{3!}Y''(x_k)s^3 + O(s^4) + O(s^2 h)\right\| = \left\|O(s^3)\right\| \leqq cs^3.$$

■

Remark 6 *For the algorithms obtained by replacing the matrix $A(x_k)$ with the matrices given in (7)-(9) sufficient conditions are $h^2 \leq s$, $h^3 \leq s$ or $h^4 \leq s$ respectively because the matrices $B(x_i)$, $C(x_i)$ or $D(x_i)$ have higher order approximation property for the second derivatives of F. These algorithms will differ from the algorithm (16) in its main term of error.*

3.2 Methods of third order accuracy

The third order Taylor's series algorithms can be constructed by the matrices given in (7)-(9) and (11), (12).

Theorem 7 *Let assume, that the fourth order derivatives of the solution of (2) are bounded, there exists such constant $c_2 > 0$ that $\left\|\frac{\partial^3 F}{\partial Y^3}\right\| \leq c_2$ and let h and s be parameters satisfying the condition $s \leq h$. Then the following expression gives a third order explicit algorithm.*

$$\begin{gathered}\hat{Y}(x_k+s) = Y(x_k) + F(x_k)s + \frac{1}{2}B(x_k)F(x_k)s^2 + \\ +\frac{1}{3!}\{[(G_0(x_k)F(x_k), F(x_k)), (G_1(x_k)F(x_k), F(x_k)), ..., \\ (G_n(x_k)F(x_k), F(x_k))]^T + B^2(x_k)F(x_k)\}s^3,\end{gathered} \tag{17}$$

where for the bilinear form G we gave the explicit form for calculation its value at the point $F(x_k)$.

Remark 8 *One could get further third order methods if in the expression (17) the matrix $B(x_k)$ would be replaced by the matrices $C(x_k)$, or defined in (8) or the matrix $B^2(x_k)$ would be replaced by matrices $C^2(x_k)$ or $D^2(x_k)$ and the matrices $G_i(x_k), i = 1, 2, ..., n$ would be replaced by matrices $H_i(x_k), i = 1, 2, ..., n$ defined in (12).*

Proof. We have to prove, that the polynomial on the right hand side of (17) up to third order is equal with the Taylor's expansion given in (3). So if s is small enough then: $\left\|Y(x_k+s) - \widetilde{Y}(x_k+s)\right\| \leq cs^2$, where the constant $c > 0$ is independent from s. Using the results of Lemma 1., Lemma 2. and Theorem 1. we get:

$$\begin{gathered}||Y(x_k+s) - \widetilde{Y}(x_k+s)|| \leq \\ \leq ||\frac{1}{3!}Y^{III}(y_k)s^3 + \frac{1}{4!}Y^{IV}(x_k)s^4 + O(s^5) - \\ -\frac{1}{3!}[(\frac{\partial^2 F}{\partial Y^2}FF) + \frac{\partial F}{\partial Y}(\frac{\partial F}{\partial Y}F)]s^3 + O(s^2h^2) + O(s^3h)|| \leq \\ \leq ||\frac{1}{3!}Y^{IV}(x_k)s^4 + O(s^5) + O(s^2h^2) + O(s^3h)|| \leq \|O(s^4)\| \leq cs^4.\end{gathered} \tag{18}$$

■

Remark 9 *If we consider the algorithms mentioned in Remark 2. in the Theorem 2. The condition $h^2 < s$ will be the sufficient one. These algorithms differ from each other in the main term of error.*

3.3 Methods of fourth order accuracy

The fourth order Taylor's series algorithms can be constructed by the matrices given in (8),(9),(11),(12) and (15).

Theorem 10 *Let assume, that the fifth order derivatives of the solution of (1) are bounded, there exists such c_3 constant that $\left\|\frac{\partial^4 F}{\partial Y^4}\right\| \leq c_3$ and let h and s be parameters satisfying the condition $h \leq s$. Then the following expression gives a fourth order explicit algorithm.*

$$\begin{gathered}\widetilde{Y}(x_k+s) = Y(x_k) + F(x_k)s + \frac{1}{2!}C(x_k)F(x_k)s^2 + \\ +\frac{1}{3!}[H(x_k)F(x_k)F(x_k) + C^2(x_k)F(x_k)]s^3 + \\ +\frac{1}{4!}[T(x_k)F(x_k)F(x_k)F(x_k) + H(x_k)(C(x_k)F(x_k))F(x_k) + \\ +2H(x_k)F(x_k)(C(x_k)F(x_k)) + C(x_k)(H(x_k)F(x_k)F(x_k)) + \\ +C^3(x_k)F(x_k)]s^4\end{gathered} \tag{19}$$

Proof. We have to prove, that the polynomial on the right hand side of (19) up to fourth order is equal with the Taylor's expansion given in (3). So if s is small enough then: $\|Y(x_i+s) - \tilde{Y}(x_i+s)\| \leq cs^3$, where the constant c is independent from s. Using the results of Lemma 1-3. and Theorem 1-2. we get:

$$\begin{gathered}||Y(x_i+s) - \tilde{Y}(x_i+s)|| \leq ||\frac{Y^{IV}(x_i)}{4!} + \frac{Y^V(x_i)}{5!} + O(s^6) - \\ -\frac{1}{4!}[T(x_i)F(x_i)F(x_i)F(x_i) + 2H(x_i)(C^2(x_i)F(x_i)F(x_i)) + \\ +2C(x_i)(H(x_i)F(x_i)F(x_i)) + C^3(x_i)F(x_i)]s^4 + O(s^2h^3 + s^3h^2 + s^4h)|| \leq \\ \leq ||\frac{Y^V(x_i)}{5!} + O(s^6 + s^2h^3 + s^3h^2 + s^4h)|| = ||O(s^5)|| \leq cs^5,\end{gathered}$$

which prove our theorem. ■

3.4 The stability analysis of the algorithms

The methods of Taylor's series defined above belong to the family of the explicit one step algorithms. General form of these algorithms can be given by the following formula

$$\begin{gathered}\widetilde{Y}(x_n+s) = \widetilde{Y}_n + s\Phi(x_n, \widetilde{Y}_n; s), \quad n = 1, 2, ..., \\ Y(x_0) = [x_0, y_1(x_0), ..., y_n(x_0)]^T,\end{gathered} \tag{20}$$

where

$$\Phi(x_n, \widetilde{Y}_n; s) = Y'(x_n) + \frac{1}{2}Y''(x_n)s + ... + \frac{1}{p!}Y^{(p)}(x_n)s^{(p-1)} + O(s^p).$$

This is true, because we have proved that the algorithms defined above are p order consistent methods for $p = 4$. We recall the well known theorem for the stability of the methods given by formula (20) see [10]. This theorem says: the $(p-1)$ order algorithm given by (20) is stable on the interval $x \in [x_0, x_0+T]$ if $F \in C^p([x_0, x_0+T] \times R^n)$. So we can formulate the following theorem.

Theorem 11 *The algorithm given by formula (20) is consistent stable one step explicit algorithm.*

Proof. The conditions sufficient for the theorem cited above are fulfilled, because at the construction of approximate calculation of matrices we have supposed its fulfillment. ■

4 Computer implementation and test results

The algorithms proposed above can be regarded as analytic-numeric algorithms because they define the approximate solution as a piecewise polynomial. Therefore we have decided to realize these algorithms by using computer algebraic system, namely the MapleV. The realization of the Taylor's series algorithm for an equation by MapleV is not a complicated task, therefore we have worked out an implementation for the system of equations.

4.1 The test problem

To test the efficiency of the proposed algorithm we used the following stiff initial value problem arising from the biochemistry see [11].

$$\frac{dy_1(t)}{dt} = \frac{1}{\alpha}(y_1(t) + y_2(t) - y_1(t)y_2(t) - qy_1^2(t)),$$

$$\frac{dy_2(t)}{dt} = 2my_3(t) - y_2(t) - y_1(t)y_2(t),$$

$$\frac{dy_3(t)}{dt} = \frac{1}{r}(y_1(t) - y_3(t)), \quad y_1(0) = a, \quad y_2(0) = b, \quad y_3(0) = d.$$

Here α, m, q and r are some parameters, a, b and d are the initial values. For some values of parameters this model has a periodic solution very sensitive for the parameter values.

4.2 Automatic error control and numerical complexity

Usually the arithmetic complexity of the algorithms for numerical solution of the initial value problems of ordinary differential equations measured by the number of function evaluation by step. One of the advantage of the above proposed algorithms is that the truncated Taylor's series is obtained in explicit form in every subinterval. This is the main point from which one can profit using MapleV. This property facilitate to investigate several properties of the approximate solution such as the local error of approximate solution for each component because the main term of the local error is known for example in the form of "- $1050.796332s^3$", where s is the local time variable in a subinterval $[x_k, x]$. So without any further calculation one can estimate the admissible step size to ensure the prescribed local error for each component of the solution. Using this technique while we apply these algorithms we get an automatic step size control algorithm too. One simple step size control algorithm could be the following.

i) We chose an appropriate value of h.

ii) One give the admissible local error $\varepsilon > 0$ and choose the order p of the algorithm.

iii) From calculation the values $e_i = \left|y_i^{(p)}(x_k)\right|$, $i = 1, 2, ..., n-1$, are known for every k.

iv) In the interval $[x_k, x]$ we calculate the value $me = \max\limits_{1 \le i \le n}(\left| y_i^{(p)}(x_n) \right|)$.

v) We chose such step size s_k for which $s_k < \sqrt[p]{\frac{\varepsilon}{me_k}}$ and $x_{k+1} = x_k + s_k$.

vi) If $h > s_k$ than $h := \frac{s_k}{2}$, go to i).

So we can conclude, that for the adaptive step size control is not necessary further function evaluation contrary to the Richardson extrapolation or embedded R-K algorithms. For the algorithm (19) the number of function evaluations are $13n$, where n is the number of equations. These evaluations can be made fully parallel. We have to remark, that the error control by Richardson extrapolation require 12 function evaluation for a fourth order R-K method. In the case these extra evaluations are not needed.

Applying the Taylor's series algorithm one can define another error indicator different from above defined which can be used for the above step size control algorithm. This error indicator can be constructed in the following way: the idea of the construction is based on the fact, that the truncation of Taylor's series of the solution, for example $\widetilde{y}(x_k + s) = \tilde{y}(x_k) + \tilde{y}'(x_k)s + \frac{1}{2!}\tilde{y}''(x_k)s^2 + \frac{1}{3!}\tilde{y}'''(x_k)s^3 + \frac{1}{4!}\tilde{y}^{IV}(x_k)s^4$ is defined on some neighborhood of x_k, so not only for the interval $[x_k, x]$ but for the $[x_{k-1}, x_k]$ too. So the truncated Taylor series can be evaluated easily at the grid points x_{k-1}. If the last admitted step size was not too large then $|\widetilde{y}(x_k - s_{k-1}) - y(x_{k-1})| = \delta$ must be such small as the tolerance ε. If $\varepsilon \ll \delta$, then the last admitted step s_{k-1} probably was too large.

Let the parameter values be as follows: $\alpha := 0.1$; $q := 0.01$; $m := 0.5$; $r := 1$ and the initial conditions are $a = 0$, $b = 0.5$ and $d = 0.8$. The test problem was solved on the interval $[0, 30]$.

The following results are obtained by the algorithm (19) which is a fourth order variant of above explained algorithms. We remark, that applying the error control value $\varepsilon = 0.01$ the number of the steps on this the interval $[0, 30]$ was nearly 500. In the Figure 1. one can see the numerical solution of our test problem.

For the reader who uses the conventional numerical algorithms for the solution of ordinary differential equation the result obtained by the algorithm (19) are curious. In the Table 1. we have shown the truncated Taylor's series for some set of subintervals generated by the automatic error control. We have to remark, that the coefficients of the truncated Taylor series for this test problem are exact, because, the right hand side of the equation is second order polynomial and its derivatives the matrices D, H and T approximate exactly.

Table 1. The approximate solution in some subintervals

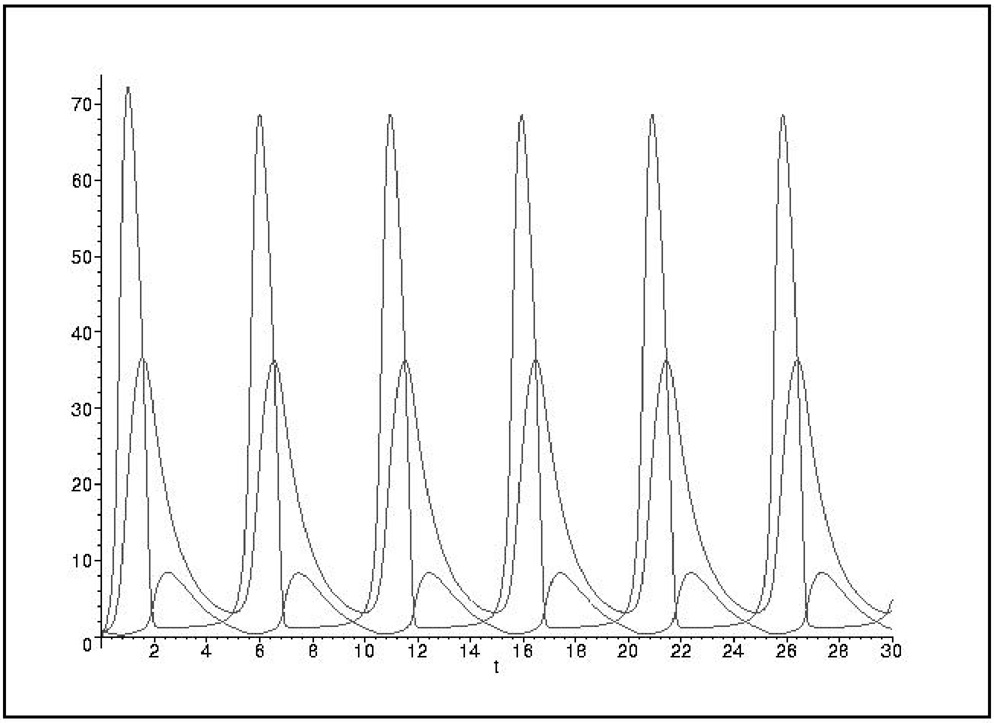

Figure 1: The numerical solutions obtained by algorithm (19)

Subintervals $[x_k, x_{k+1}]$	Calculted truncated Taylor series of solution $\tilde{y}_i(x_k+s), \quad i=1,2,3$
[2.523273367, 2.560628954]	$\tilde{y}_1 = 1.133904235 - 0.0593182614s + 2.452116961s^2$ $-57.48999655s^3 + 1069.883545s^4$ $\tilde{y}_2 = 8.416303435 - 0.367097903s - 7.587995590s^2$ $+1.244050777s^3 + 119.9332005s^4$ $\tilde{y}_3 = 17.59248764 - 16.45858341s + 8.199632575s^2$ $-1.915838538s^3 - 13.89353950s^4$
[2.560628954, 2.598028225]	$\tilde{y}_1 = 1.134196674 - 0.0488610395s + 2.430056057s^2$ $-57.19861627s^3 + 1064.893576s^4$ $\tilde{y}_2 = 8.392300074 - 0.921836135s - 6.738674970s^2$ $+0.6152729010s^3 + 119.7003332s^4$ $\tilde{y}_3 = 16.98898277 - 15.85478610s + 7.902962530s^2$ $-1.824302158s^3 - 13.84357853s^4$
[2.598028225, 2.635578062]	$\tilde{y}_1 = 1.134859481 - 0.0391621042s + 2.398377540s^2$ $-56.43639293s^3 + 1047.916127s^4$ $\tilde{y}_2 = 8.348665016 - 1.416269923s - 5.960804335s^2$ $0.08774420324s^3 + 118.1012914s^4$ $\tilde{y}_3 = 16.40695674 - 15.27209726s + 7.616467580s^2$ $1.739363347s^3 - 13.67425740s^4$

One can see, that the results contain much more information about the approximate solution than the classical numerical solution. This is the main reason, that the proposed algorithm can be useful in some cases in spite of its complexity. However we have to remark, that the automatic error control for this algorithm is much simpler than dose of the classical one step methods.

5 Conclusions

The Taylor series method with numerical derivatives proposed in this article is such algorithm which can be competitive with the classical algorithm for numerical solution of initial value problems for ordinary differential equations for parallel computers.

The complexity of these algorithms for parallel computers can be better than the classical ones because the function evaluations can be performed fully-parallel and they only use matrix-vector operations. These methods can be regarded as numerical-analytical algorithms. The obtained results contain more information about solution and these is a promising feature for the construction of further algorithms having some "quality properties" such as energy preserving, positivity preserving, etc.

The analytic-numeric structure of the approximate solution offers some new, easy error control algorithms which could be useful for the stiff systems as well. The numerical experiments performed by the MapleV system show that the proposed algorithms work well.

There are a lot off way to develop the basic algorithm, for example the automatic step size control could be improved, the information obtained by the approximate solution could be used more intensively. The realization and test of the proposed algorithm for parallel computers is a perspective topic to develop because the approximation of the bilinear and trilinear forms given in (11), (12) and (15) have special data structure especially if the elements of them are mostly zeros. Therefore realization require special data handling technique and can be solved effectively by the new approach given in [17]. Each of these investigation could be the subject of forthcoming paper.

References

[1] I. Newton, *Methodus Fluxionum et Serierum Infinatarum*, edita Londini 1736, Opuscula mathematica vol. I., 1761.

[2] L. Euler, *Institutionum Calculi Integralis*, Volumen Primum, Opera Omnia vol. XI., 1768.

[3] J. Liouville, *Sur le développement des fonctions ou parties de fonctions en series don't les divers termes sont assujetis a satisfaire a une meme equation differentielle du second ordre, contenat une parametre variable*, Journal De Math. Pures et appl.., vol.1, 253-265, 1836.

[4] G. Peano, *Intégration par series des équations différentielles linéaires*, Math. Anallen, vol.32, 450-456, 1888.

[5] E. Picard, *Mémoire sur la théorie des équations aux dérivées partiellee et la méthode des approximations successives*, J. de Math. Pures et appl. 4e série, vol. 6, 145-210, 1899.

[6] E. Harier, S. P. Norsett, G. Wanner, *Solving Ordinary Differential Equations, I. Nonstiff Problems*, Springer Verlag, 1987.

[7] E. Harier, S. P. Norsett, G. Wanner, *Solving Ordinary Differential Equations, II. Stiff and Differential-Algebraic Problems*, Springer Verlag, 1991.

[8] Z. Horváth, *Consistency and stability for some nonnegativity-conserving methods*, Applied Numerical Mathematics, 13, 371-381, 1993.

[9] G. Molnárka, B. Ráczkevi, *Implicit single step method by spline-like functions for solutions of ordinary differential equations*, Comput. Math. Applic., vol. 16, No. 9., 701-704, 1988.

[10] M. Crouzeix, A.L. Mignot, *Analyse numerique des equations differentielles*, MASSON, Paris, 1984.

[11] J.B. Murray, *Lectures on nonlinear-differential-equation models in biology*, Clarendon Press, Oxford, 1977.

[12] A. Gibbons, *A program for the automatic integration of differential equations using the method of Taylor series*, Computer J. Vol. 3., 108-111, 1960.

[13] R.E. Moore, *Methods and applications of interval analysis*, SIAM studies in Appl. Math, 1979.

[14] À. Jorba and M. Zou, *A software package for the numerical integration of ODE by means of high-order Taylor methods*, http://www.maia.ub.es/~angel/, 2001.

[15] P. E. Miletics and G. Molnárka, *Taylor Series Methods with Numerical Derivatives for Initial Value Problems*, in Computational and Mathematical Methods on Science and Engineering, Vol. I. eds.: J. Vigo Aguiar and B. A. Wade. Proc. of CMMSE-2002, Alicante, Spain, pp.258-270.

[16] R. Neidinger, *An efficient method for the numerical evalution of partial derivatives of arbitrary order*, ACM Trans. Mathm Software, Vol. 18., 159-173, 1992.

[17] B.Kiss and A. Krebsz, *On the Relational Database Type Numerical Programming*, Proceeding of the Third International Conference on Engineering Computational Technology, B.H.V. Topping and Z. Bittnar (Editors), Civil-Comp Press, Stirling, Scotland. Paper 48 (2002)