Implicit Extension of Taylor Series Method with Numerical Derivatives for Initial Value Problems

E. MILETICS AND G. MOLNÁRKA
Department of Mathematics, Széchenyi István University, Győr, Hungary
<miletics><molnarka>@sze.hu

Abstract—The Taylor series method is one of the earliest analytic-numeric algorithms for approximate solution of initial value problems for ordinary differential equations. The main idea of the rehabilitation of this algorithms is based on the approximate calculation of higher derivatives using well-known finite-difference technique for the partial differential equations. The approximate solution is given as a piecewise polynomial function defined on the subintervals of the whole interval integration. This property offers different facility for adaptive error control. This paper describes several explicit Taylor series algorithms with numerical derivatives and their implicit extension and examines its consistency and stability properties. The implicit extension based on a collocation term added to the explicit truncated Taylor series and the approximate solution obtained as a continuously differentiable piecewise polynomials function. Some numerical test results is presented to prove the efficiency of these new-old algorithm. © 2005 Elsevier Ltd. All rights reserved.

Keywords—Numerical derivatives, Finite differences, Implicit algorithms, Collocation methods.

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 described it in his work [2]. Since then, we can find many mentions of it, such as [3–5]. A lot of authors have further developed this algorithm, see, for example, [6,7]. 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. 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 conditions 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 evaluations are realized only at grid points. Therefore, these algorithms give the values of the approximate solution only at grid

This paper was partly supported by OTKA T 043258.
points. For such algorithms, main cost is the number of the function evaluation. These algorithms differ from others in their order, stability properties, and their cost of realization. The overview of the modern algorithms, we can find in the monograph of [8,9]. A possible implicit extension of the Taylor series algorithm is given in [10]. Actually, in the qualification of algorithms their quality properties such as conservativity, positivity preserving [11], and monotonity preserving become important.

While the obtained approximate solution by the proposed algorithm are piecewise continuously differentiable functions, the qualitative properties of the numerical method could be examined and controlled more easily than using classical methods.

Using parallel computers to measure the complexity of the algorithms must be defined new cost functions because in this case, the main goal is to minimize the execution time and not the number of function evaluations. We can see that for the explicit versions of Taylor series algorithm with numerical derivatives, the function evaluations and the calculation of the coefficients of the Taylor series can be performed fully parallel while for the explicit R-K algorithms, the function evaluations can be made only in sequential order. From this point of view, several variants of the method of the Taylor series with numerical derivatives could be an effective algorithm. To develop Taylor series algorithms is also important because most of the validated numerical methods for the solution of ODE initial-value problems are based on Taylor series, see [12].

One family of such algorithms can be derived from the classical method of the Taylor series by approximating the derivatives in Taylor coefficients with numerical derivatives which we have proposed in [13]. In this paper, we propose an implicit version of Taylor series algorithm. The implicit version has an advantage with respect to the explicit one, it has some $L$-stability region (see Section 3.4), but the calculation of the implicit correction is a sequential step in the algorithm.

1.1. Formulation of the Problem

The problems to be solved are as follows,

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0,$$

(1)

where

$$x \in [x_0, x_0 + T], \quad y(x) = [y_1(x), y_2(x), \ldots, y_n(x)]^T : R \to \mathbb{R}^n,$$

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

for a given $p$ and

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

Let us introduce the following notations,

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

(2)

By this notation, equation (1) is as follows,

$$Y'(x) = F(Y(x)), \quad Y(x_0) = [x_0, y_1(x_0), y_2(x_0), \ldots, y_n(x_0)]^T.$$

(3)

By this step, we deduced the well-known fact that every nonautonomous system can be transformed into autonomous one, but in this paper, from the point of view of the realization of the proposed algorithm, the notation has some practical significance. Using this notation, the Taylor series of the solution of (1) is

$$Y(x_0 + s) = Y(x_0) + Y'(x_0)s + \frac{1}{2}Y''(x_0)s^2 + \cdots + O(s^{p+1}),$$

(4)

where $O$ denotes well-known asymptotic order function, and value of $p$ might be chosen depending on the smoothness of the right side in (3) and the desired order of the method to be obtained.
2. NUMERICAL APPROXIMATION OF THE DERIVATIVES

The main idea of the construction of the method of Taylor algorithm with numerical derivatives is the numerical approximation of the derivatives \( Y^{(k)}(x_0), \ k = 2, 3, 4, \ldots \). For higher-order derivatives, we have to approximate several partial derivatives \([14]\). In the following, we construct some approximations of the derivatives of the right-hand side \( F(Y(x)) \) of (3) defined in (2) by the matrices (linear form for first derivatives) by bilinear, trilinear, etc., forms (for second, third, etc., derivatives). The first derivatives are given by the right-hand side of (1), its approximation is only an evaluation.

2.1. Numerical Approximation of the Second Derivatives

The second derivative of the left-hand side of (3) is called its Jacobian and is as follows,

\[
Y''(x_0) = F'(Y(x_0)) = [0, f'_1(Y(x_0)), \ldots, f'_n(Y(x_0))]^T
\]

\[
= \left[ \frac{\partial f_1(x_0)}{\partial Y}, F(x_0), \frac{\partial f_2(x_0)}{\partial Y}, F(x_0), \ldots, \frac{\partial f_n(x_0)}{\partial Y}, F(x_0) \right]^T
\]

\[
= \frac{\partial F(Y(x_0))}{\partial Y} F(Y(x_0)),
\]

where the derivatives \( \frac{\partial f_k(x_0)}{\partial Y}, k = 1, 2, \ldots, n \) are to be approximated numerically.

Let us introduce the following notations,

\[
F^{k,i} := F^{k,i}(Y(x_0)) = [1, f_1(\ldots, y_k(x_0) + ih, \ldots), \ldots, f_n(\ldots, y_k(x_0) + ih, \ldots)]^T,
\]

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 (3) at the given points. Using the finite-difference formulae to approximate the first derivatives, we get that the following \( D \) matrix approximates the generalized Jacobian matrix of the right-hand side of (3).

Let us define the \( D \) matrix by the following formula,

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

**Lemma 1. The Order of Accuracy of Approximation.** Assume that every component of the function \( F \) is five times continuously differentiable with respect to all its variables in some neighborhood of \( x \) and \( y_i(x) \), \( i = 1, 2, \ldots, n \). Then, matrix \( D \) approximates the generalized Jacobian matrix \( \frac{\partial F}{\partial Y} \) by fourth-order degree of accuracy at the point \( x \) in the following sense,

\[
\max_{0 \leq i, j \leq n} \left( \left| \frac{\partial F_i}{\partial Y_j} - D_{ij} \right| \right) \leq c_4 h^4,
\]

where \( c_4 \) is constants not depending on the parameter \( h \).

**Proof.** Directly can be obtained after simple calculations.

**Remark 1.** The different accuracy order approximations of the generalized Jacobian matrix is given in [13,14].

2.2. Numerical Approximation of the Third Derivatives

The third derivative of the left-hand side of (3), the coefficient of the third term in the Taylor’s expansion (4) is

\[
Y''' = F''' = \left( \frac{\partial F}{\partial Y} \right)' = \left( \frac{\partial^2 F}{\partial Y^2} F \right)' + \frac{\partial F}{\partial Y} \left( \frac{\partial F}{\partial Y} \right) \cdot \frac{\partial F}{\partial Y}.
\]

(6)
The second term in (6) can be approximated by using matrix $D$. 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,

\[
\begin{align*}
    f^{kl,ij} &= f(\ldots, y_k + ih, \ldots, y_l + jh, \ldots), & k, l = 1, 2, \ldots, n, & i, j = 0, \pm 1, \pm 2, \ldots, \\
    f^{kk,ii} &= f(\ldots, y_k + ih, \ldots), & k = 1, 2, \ldots, n, & i = 0, \pm 1, \pm 2, \ldots.
\end{align*}
\]

(7)  

(8)

Let us define matrix $H_i$ as follows,

\[
H_i = \frac{1}{4h^2} \left( \begin{array}{c} (F^{i0,11} - F^{i0,-11} - F^{i0,1-1} + F^{i0,-1-1}) \ldots, \\
4 (F^{i1,11} - 2F^{i0,00} + F^{i1,-1-1}) \ldots, \\
(F^{in,11} - F^{in,-11} - F^{in,1-1} + F^{in,-1-1}) \ldots, \\
\end{array} \right),
\]

\[
i = 1, 2, \ldots, n.
\]

(9)

**Lemma 2.** The Order of the Approximation of Third-Order Derivatives. Assume that the conditions of the Lemma 1 are fulfilled. Then, the matrices $H_i$, $i = 1, 2, \ldots, n$ approximate the generalized bilinear formula $\frac{\partial^3 F}{\partial Y^2}$ by second-order degree of accuracy at the point $x$ in the following sense,

\[
(H_i)_{kl} \approx \frac{\partial^3 f_k}{\partial Y_k \partial Y_l}, \quad i, k, l = 1, 2, \ldots, n, \quad \text{and} \quad \max_{0 \leq i, k, l \leq n} \left| (H_i)_{kl} - \frac{\partial^3 f_k}{\partial Y_k \partial Y_l} \right| \leq c_2 h^2,
\]

where $c_2 > 0$ is independent of $h$.

**Proof.** This proof is similar as it was the proof of Lemma 1.

2.3. Numerical Approximation of the Fourth Derivatives

The fourth-order derivative of the solution vector $Y$ is the third derivative of the right-hand side of (3). So, we get

\[
Y^{IV} = F''' = \left( \frac{\partial^2 F}{\partial Y^2} F F \right)' + \left( \frac{\partial F}{\partial Y} \right)^2 F
\]

\[
= \frac{\partial^3 F}{\partial Y^3} F F F + \frac{\partial^2 F}{\partial Y^2} \left( \frac{\partial F}{\partial Y} F \right) F
\]

\[
+ 2 \left( \frac{\partial^2 F}{\partial Y^2} \left( \frac{\partial F}{\partial Y} F \right) \right) + \frac{\partial F}{\partial Y} \left( \frac{\partial^2 F}{\partial Y^2} F F \right) + \left( \frac{\partial F}{\partial Y} \right)^3 F.
\]

(10)

The derivatives in the four last terms can be approximated by matrices $D$ and $H_i$. Therefore, we have to approximate the third derivative $\frac{\partial^2 F}{\partial Y^2}$ which is a trilinear formula consisting of the elements $\frac{\partial f_k}{\partial Y_k \partial Y_l \partial Y_m}$, $i, j, k, l = 1, 2, \ldots, n$. In order to approximate these terms by finite-difference method let us introduce the following notations,

\[
f^{ij,mmmp}_{k} = f_k(\ldots, y_i + mh, \ldots, y_j + nh, \ldots, y_l + ph, \ldots),
\]

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

\[
f^{ii,mmmp}_{k} = f_k(\ldots, y_i + mh, \ldots, y_l + ph, \ldots) \quad \text{and} \quad f^{ii,mmmp}_{k}(\ldots, y_i + mh, \ldots).
\]
Using these notations, the approximation of the trilinear formula in (10) can be built up by the following matrices,

\[
\begin{align*}
T_{ij} &= \frac{1}{h^3} \left[ \left( F^{ij}0,000 - F^{ij}0,001 - F^{ij}0,0-1 + F^{ij}0,1-10 \right. \\
&\quad - F^{ij}0,1-11 + F^{ij}0,101 - F^{ij}0,100 \right) + \left( F^{ij}n,000 - F^{ij}n,001 \right. \\
&\quad - F^{ij}n,0-10 + F^{ij}n,0-11 + F^{ij}n,1-10 - F^{ij}n,1-11 \\
&\quad + F^{ij}n,101 - F^{ij}n,100 \right], \\
& \quad i, j = 0, 1, \ldots, n,
\end{align*}
\]

(11)

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

\[
(T_{ij})_j = \frac{1}{h^3} \left( 2F^{ijj,000} - F^{ijj,001} - F^{ijj,011} - 2F^{ijj,100} + F^{ijj,101} + F^{ijj,111} \right),
\]

and for three repeated indices, we get

\[
(T_{jj})_j = \frac{1}{2h^3} \left( -2F^{jjj,111} + F^{jjj,222} + 2F^{jjj,11-1} - F^{jjj,11-2} \right).
\]

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

**LEMMA 3.** Let the function \( F \), the right-hand side of (3) 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 \( T_{ij} \), \( i, j = 1, 2, \ldots, n \), and \( \max_{0 \leq i, j, k \leq n} |(T_{ij})_{kl} - \frac{\partial^3 F}{\partial Y^3}| \leq c_3 h \), where \( c_3 \) is not dependent on \( h \).

**PROOF.** The elements \((T_{ij})_{kl}\) approximate the derivative \( \frac{\partial^3 F}{\partial Y^3} \). So, describing the Taylor's series expansion of \((T_{ij})_{kl}\) in the neighbourhood of \( Y \), we get \((T_{ij})_{kl} Y = \frac{\partial^3 f}{\partial Y^3} + c_{ijkl} h\), where constants \( c_{ijkl} \) consist of some values of fourth-order derivatives of \( f_i \) from the neighbourhood of \( h \) radius. The maximum of absolute value of them will be the appropriate constant for \( c_3 \).

### 3. THE FOURTH-ORDER METHODS OF TAYLOR SERIES BASED ON NUMERICAL DERIVATIVES

By summarizing the results explained above, we can construct some truncations of Taylor's series of the \( Y \) solution of (1) as an approximate solution at a given subinterval. The basic idea of the construction is that the expressions (5), (9), and (11) of the derivatives in the truncation of Taylor's expansion (4) are replaced by its approximations which will be constructed by some combinations of the matrices (5), (9), and (11). This 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. These methods are explained and examined in [13].

#### 3.1. Explicit Methods of Fourth-Order Accuracy

**THEOREM 1.** Let assume, that the fifth-order derivatives of the solution of (1) are bounded, there exists such \( c_3 \) constant that \( \left\| \frac{\partial^5 F}{\partial Y^5} \right\| \leq c_3 \) and let \( h \) and \( s \) be parameters satisfying the condition \( h \leq t \). Let \( x_k \), \( k = 0, 1, \ldots, \), denote the grid points in \([x_0, x_0 + T]\) and \( t_k = x_{k+1} - x_k \), \( t \in (0, t_k] \). Then, the following expression gives a fourth-order explicit algorithm,

\[
\hat{Y}(x_k + t) = Y_0(x_k) + Y_1(x_k) t + \frac{1}{2!} Y_2(x_k) t^2 + \frac{1}{3!} Y_3(x_k) t^3 + \frac{1}{4!} Y_4(x_k) t^4, \tag{12}
\]
Let us introduce the following notation,
\[
\begin{align*}
    p^{-1} & \sum_{i=0}^{1} a(P) s^i, \quad p = 2, 3, 4, 5.
\end{align*}
\]

The simple iteration algorithm for determining the unknown vector \( X_k \) is as follows,
\[
    s_{\text{coll}, y(0)} = D T_{p-1}(x_k, s_{\text{coll})} = 0, 1, 2. \quad (16)
\]

where \( s_{\text{coll}} \) is given. The following theorem states that for small enough \( s_{\text{coll}} \), the iteration (16) for every \( X_k(0) \) value is convergent.

### 3.2. Implicit Taylor Series Methods up to Fifth Order

The explicit Taylor series methods can be used for the construction of implicit algorithm. The idea of the construction is that the explicit Taylor series truncations could be augmented by one or more extra terms see [10]. Here, we formulate the implicit extensions with one extra term as follows. Let us define this extension of the Taylor series truncation given in (12) for the \([x_k, x_k + s]\) interval where \( p \) denote the desired order of the implicit method and \( a_k^{(p)} \) denote the unknown correction term to be determined for the \( p \)th method in the given interval. Then,
\[
    \tilde{Y}'(x_k, a_k^{(p)}, s_{\text{coll}}) - F(\tilde{Y}(x_k, a_k^{(p)}, s_{\text{coll}})).
\]

On the left-hand side of (14), the derivation can be performed explicitly. So, we get
\[
    \tilde{Y}'(x_k, a_k^{(p)}, s_{\text{coll}}) = F(\tilde{Y}(x_k, a_k^{(p)}, s_{\text{coll}})).
\]

Let us introduce the following notation,
\[
    T_{p-1}(x_k, s) = \sum_{i=0}^{p-1} \frac{1}{i!} Y_i(x_k) s^i, \quad DT_{p-1}(x_k, s) = \sum_{i=1}^{p-2} \frac{1}{(i-1)!} Y_i(x_k) s^{p-1},
\]

and
\[
    X_k = \frac{1}{(p-1)!} a_k^{(p)} s_{\text{coll}}.
\]

The simple iteration algorithm for determining the unknown vector \( X_k \) is as follows,
\[
    X_k^{(l+1)} = F(T_{p-1}(x_k, s_{\text{coll}}) + \frac{s_{\text{coll}}}{k} X_k^{(l)}) - DT_{p-1}(x_k, s_{\text{coll}}), \quad l = 0, 1, 2, \ldots
\]

where \( X_k^{(0)} \) is given. The following theorem states that for small enough \( s_{\text{coll}} \), the iteration (16) for every \( X_k^{(0)} \) value is convergent.
THEOREM 2. Let us assume, that the right-hand side of (1) satisfy the following Lipschitz condition,
\[ \|F(Z_1) - F(Z_2)\| \leq L\|Z_1 - Z_2\|. \] (17)

Then, if \((\text{scoll}/p)L < 1\), equation (14) has an unique solution and iteration (16) converges to this solution from every initial value \(X_k^{(0)}\), \(k = 1, 2, \ldots\).

PROOF. Subtracting the \((l + 1)\)th and \(l\)th iteration steps from each other, we get
\[
\left\|X_k^{(l+1)} - X_k^{(l)}\right\| = \left\|F\left(T_{p-1}(x_k, s_{\text{coll}}) + \frac{s_{\text{coll}}}{p}X_k^{(l)}\right) - F\left(T_{p-1}(x_k, s_{\text{coll}}) + \frac{s_{\text{coll}}}{p}X_k^{(l-1)}\right)\right\| \\
\leq L\frac{s_{\text{coll}}}{p}\left\|X_k^{(l)} - X_k^{(l-1)}\right\|
\]
which proves our assertion.

REMARK 2. We see that the iteration is convergent for small enough \(s_{\text{coll}}\) value and by comparison with the classical implicit algorithm this \(s_{\text{coll}}\) value can be \(k\) times greater, see (16). The other advantage of this iteration is that initial values for iteration \(X_k^{(0)} = 0\) for every \(k\) in most cases is a good choice because the solution must be near zero.

3.3. The Consistency Order of the Implicit Methods

The exact solution of the problem (1) in the interval \([x_k, x_{k+1}]\) is \(Y(x_k + s)\) while the approximate solution obtained by the proposed implicit methods (14) is \(\tilde{Y}(x_k, a_k^{(p)}, s)\).

THEOREM 3. If the right-hand side of (1) has continuous derivatives up to order \((p + 1)\), then consistency order of the proposed implicit algorithms is \(p + 1\), that there is such constant \(M > 0\), that for every \(k = 1, 2, \ldots\),
\[
\left\|Y(x_k + s_{\text{coll}}) - \tilde{Y}\left(x_k, a_k^{(p)}, s_{\text{coll}}\right)\right\| \leq M \frac{s^{p+1}}{p_{\text{coll}}}.
\] (18)

while for the \(s \in [0, s_{\text{coll}}]\), the following estimation holds. There is such \(N > 0\), that for every subinterval,
\[
\left\|Y(x_k + s) - \tilde{Y}\left(x_k, a_k^{(p)}, s\right)\right\| \leq N s^p.
\] (19)

PROOF. Let us substitute the exact and the approximate solution into the original equation and subtract them from each other. We get
\[
\left\|\tilde{Y}'\left(x_k, a_k^{(p)}, s\right) - Y'\left(x_k + s\right)\right\| = \left\|F\left(\tilde{Y}\left(x_k, a_k^{(p)}, s\right)\right) - F\left(Y\left(x_k + s\right)\right)\right\| \\
\leq L \left\|\tilde{Y}\left(x_k, a_k^{(p)}, s\right) - Y\left(x_k + s\right)\right\| = L \left\|\frac{1}{(p+1)!} a_k^{(p+1)} s^{p+1} \\
- \left(\frac{1}{(p+1)!}\right) Y^{(p+1)}(x_k) s^{p+1} + O(s^{p+2})\right\|, \quad k = 2, 3, 4.
\]

but the collocation condition says that at the point of \(x_k + s_{\text{coll}}\),
\[
a_k^{(p+1)} = Y^{(p+1)}(x_k) + O(s_{\text{coll}}),
\]
so we get that the estimation (18) holds. The other point for \(s \in (0, s_{\text{coll}})\), only estimation (19) is valid.
Table 1. Linear transfer functions for the implicit Taylor series methods.

<table>
<thead>
<tr>
<th>The Method Order</th>
<th>The Transfer Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$\frac{2 + z}{2 - z}$</td>
</tr>
<tr>
<td>3</td>
<td>$\frac{6 + 4z + z^2}{6 - 2z}$</td>
</tr>
<tr>
<td>4</td>
<td>$\frac{24 + 18z + 6z^2 + z^3}{24 - z}$</td>
</tr>
<tr>
<td>5</td>
<td>$\frac{120 + 96z + 36z^2 + 8z^3 + z^4}{120 - z}$</td>
</tr>
</tbody>
</table>

Figure 1. The stability regions of the methods; (3) third-order, (4) fourth-order, (5) fifth-order methods.

3.4. The Stability Analysis of the Algorithms

The linear stability analysis of the implicit Taylor series algorithm is similar as the analysis of the other implicit algorithms. The transfer or stability function for the above proposed algorithms can be calculated easily and we give them in Table 1, where we use the $\lambda s = z$ notation.

From Figure 1, it is clear that the implicit methods of order 3, 4, and 5 are conditionally stable algorithms and its stability conditions are very near to each other. (On Figure 1, the curves are almost similar.) The stability function of the second-order implicit Taylor series algorithm is equal with the stability function of the well-known implicit trapezoidal formula, therefore, it is unconditionally linear stable algorithm. We have to remark that the trapezoidal formula and the proposed second-order implicit Taylor series algorithm seem to be the same. For the linear equation, $y' = \lambda y$, they give the same results, but for other equations these methods are different.

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 MAPLE V. The realization of the Taylor series algorithm for an equation by MAPLE V 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 well-known two-body problem [6],

\[
y'_1(t) = y_3(t), \quad y'_2(t) = y_4(t),
\]

\[
y'_3 = \frac{y_1}{(y_1^2 + y_2^2)^{3/2}}, \quad y'_4 = \frac{y_2}{(y_1^2 + y_2^2)^{3/2}},
\]

\[
y_1(0) = 0.5, \quad y_2(0) = 0, \quad y_3 = 0, \quad y_4 = \sqrt{3}.
\]

This model has a periodic solution.

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 advantages of the above proposed algorithms is that the truncated Taylor series is obtained in explicit form in every subinterval. This is the main point from which one can profit using MAPLE V. This property facilitates 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 \(-10^{50.796332s^3}\), 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 [15].

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 gives the admissible local error \(e > 0\) and choose the order \(p\) of the algorithm.

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

(iv) In the interval \([x_k, x]\), we calculate the value \(m_{ek} = \max_{1 \leq i \leq n}(|y^{(p)}(x_k)|)\).

(v) We chose such step size \(s_k\), for which \(s_k < p\sqrt{e/m_{ek}}\) and \(x_{k+1} = x_k + s_k\).

(vi) If \(h > s_k\), then \(h := s_k/2\), go to (ii).

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 (12) 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 \(12n\) sequential function evaluation for a fourth-order R-K method. But we have to remark that by using embedded R-K methods less sequential function evaluation is enough.

The following results are obtained by the algorithm (12) with implicit extension (16) fifth-order variant of above explained algorithms. We remark that applying the error control value \(e = 0.02\) the number of the steps on this the interval \([0, 20]\) was nearly 200. In Figure 2, one can see the numerical solution of our test problem (solid curve) while the circles show the solutions calculated by RKF45 built-in method.

For the reader who uses the conventional numerical algorithms for the solution of ordinary differential equation, the result obtained by the algorithm (16) are curious. In Table 2, we show the truncated Taylor series for some set of subintervals generated by the automatic error control.

5. CONCLUSIONS

The implicit 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 use only matrix-vector operations. These methods can be regarded as numerical-analytical algorithms. The obtained results contain more information about solution and this is a promising feature for the construction 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 MAPLE V system show that the proposed algorithms work well.

There are a lot of 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 [16]. Each of these investigations could be the subject of forthcoming papers.

REFERENCES

2. L. Euler, Institutionum Calculi Integralis, Volumen Primum, Opera Omnia, Volume XI, (1768).