Research Article

Petr Veigend*, Gabriela Nečasová, and Václav Šátek

Taylor Series Based Numerical Integration Method

<https://doi.org/10.1515/comp-2020-0163>

Received Feb 29, 2020; accepted Apr 27, 2020

Abstract: This article deals with a high order integration method based on the Taylor series. The paper shows many positive properties of this method on a set of technical initial value problems. These problems can be transformed into the autonomous systems of ordinary differential equations for both linear and nonlinear problems. The MATLAB implementation of the method is compared with state-of-the-art MATLAB solvers.

Keywords: ordinary differential equations, initial value problems, autonomous systems, Taylor series, MTSM, MATLAB

1 Introduction

The article deals with the solution of technical initial value problems (IVPs). The solution of the technical IVPs means to find the numerical solution of the system of ordinary differential equations (ODEs). In the article, ODEs are solved using a high order variable-step variable-order numerical integration method – Modern Taylor Series Method (abbreviated as MTSM in the article) [1].

The article consists of several sections. In Section 2, MTSM and its properties is introduced. Subsections 2.1 and 2.2 focus on the solution of linear and nonlinear problems using MTSM, respectively. Several examples of linear and nonlinear problems are introduced in Section 3; namely, movement of a charged particle (Section 3.1), calculation of Fourier coefficients (Section 3.2) and Kepler problem (Section 3.3). The results obtained by MTSM are compared with state-of-the-art MATLAB solvers. This article extends

the paper [2]. The new nonlinear problem (Kepler problem) has been added and other experiments were revised. Nonlinear problems and their solutions using the MTSM are analyzed in detail, including step-by-step simplification and transformations of the problem into several autonomous systems that are compared.

The article follows the previously published article [3], which focused on linear systems of ODEs and their solutions.

2 The Modern Taylor Series Method

This section of the article briefly introduces the developed method – MTSM. It compiles information from previously published works, mainly [1, 4, 5] and other works cited throughout.

An ODE

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

with initial condition

$$y(t_0) = y_0, \quad (2)$$

is called an initial value problem. The best-known and the most accurate method of calculating a new value of the numerical solution of an ODE is to construct the Taylor series in the form

$$y_{i+1} = y_i + h \cdot f(t_i, y_i) + \frac{h^2}{2!} \cdot f'(t_i, y_i) + \dots + \frac{h^n}{n!} \cdot f^{[n-1]}(t_i, y_i), \quad (3)$$

where h is the size of the integration step, $y_i = y(t_i)$ is the previous value and $y_{i+1} = y(t_i + h)$ is the next value of the function $y(t)$ [6].

The MTSM very effectively implements the variable-step-size, variable-order numerical calculation of differential equations using the Taylor series' [1]. It is based on a recurrent calculation of the Taylor series' terms for each integration step. Therefore, the complicated calculation of higher-order derivatives does not need to be performed, but rather the value of each Taylor series term can be numerically calculated [1].

***Corresponding Author: Petr Veigend:** Brno University of Technology, Faculty of Information Technology, Božetěchova 2, 612 66, Brno, Czech Republic; Email: iveigend@fit.vutbr.cz

Gabriela Nečasová: Brno University of Technology, Faculty of Information Technology, Božetěchova 2, 612 66, Brno, Czech Republic

Václav Šátek: Brno University of Technology, Faculty of Information Technology, Božetěchova 2, 612 66, Brno, Czech Republic; IT4Innovations, VŠB Technical University of Ostrava, 17. listopadu 15/2172, 708 33, Ostrava-Poruba, Czech Republic

Equation (3) can then be rewritten in the form

$$y_{i+1} = DY_0 + DY_1 + DY_2 + \dots + DY_n, \quad (4)$$

where DY denotes the Taylor series' terms.

The MTSM automatically transforms the input problem into a system of autonomous ODEs, which allows the recurrent calculation of terms of the Taylor series. Another important part of the method is an automatic integration order setting, *i.e.* using as many Taylor series terms as the defined accuracy requires. Let us denote as ORD the function which changes during the computation and defines the number of Taylor series terms used in the current integration step y_{i+1} ($ORD_{i+1} = n$). The MTSM allows for computation with arbitrary accuracy and step-size if variable-precision arithmetic and higher order of the method is used.

The first implementation of the MTSM was TKSL/386 (TKSL stands for Taylor-Kunovsky Simulation Language) [7]. Currently, the MTSM has been implemented and tested in MATLAB [8], in C/C++ languages (FOS [9] and TKSL/C software [10]). Additionally, the method can be effectively implemented in hardware [11]. Several other implementations of the Taylor series method in a variable order and variable step context were presented by different authors *e.g.* TIDES software [12] and TAYLOR [13], which includes a detailed description of a variable-step-size version. Other implementations based on Taylor series include ATOMF [14], COSY INFINITY [15], and DAETS [16]. The variable step-size variable-order scheme is also described in [17, 18], and [19], where simulations on a parallel computer are shown. An approach based on an approximate formulation of the Taylor methods can be found in [20].

2.1 Linear Problems

For linear systems of ODEs, Equation (1) is in the form

$$y' = Ay + b. \quad (5)$$

For a linear system of ODEs (5), the Taylor series (3) can be rewritten in matrix-vector notation as

$$y_{i+1} = y_i + h(Ay_i + b) + \frac{h^2}{2!}A(Ay_i + b) + \dots + \frac{h^n}{n!}A^{(n-1)}(Ay_i + b), \quad (6)$$

where A is the constant Jacobian matrix and b is the constant right-hand side. Moreover, Taylor series terms DY in (4) can be computed recurrently using

$$\begin{aligned} DY_0 &= y_i, \quad DY_1 = h(Ay_i + b), \\ DY_j &= \frac{h}{j}ADY_{j-1}, \quad j = 2, \dots, n. \end{aligned} \quad (7)$$

2.2 Nonlinear Problems

The effective solution of nonlinear systems of ODEs is described. For such a system, a Taylor series based numerical method can be implemented in a very effective way.

Equation (1) for nonlinear systems of ODEs can be rewritten as

$$y' = Ay + B_1 y_{jk} + B_2 y_{jkl} + \dots + b, \quad y(0) = y_0, \quad (8)$$

where $A \in \mathbb{R}^{ne \times ne}$ is the matrix representing the linear part of the system and $B_1 \in \mathbb{R}^{ne \times nm_{jk}}$, $B_2 \in \mathbb{R}^{ne \times nm_{jkl}}$ are the matrices representing the nonlinear part of the system, $b \in \mathbb{R}^{ne}$ is the right-hand side for the forces incoming to the system, y_0 is a vector of the initial conditions, the symbol ne stands for the number of equations of the system of ODEs and the symbols nm_{jk} , nm_{jkl} mean the number of multiplications. The unknown function $y_{jk} \in \mathbb{R}^{nm_{jk}}$ represents the vector of multiplications $y_j \cdot y_k$ and similarly $y_{jkl} \in \mathbb{R}^{nm_{jkl}}$ represent the vector of multiplications $y_{jj} \cdot y_{kk} \cdot y_{ll}$, where indexes $j, k, jj, kk, ll \in (1, \dots, ne)$ come from multiplications terms in the system (8). The operation ' $\cdot$ ' stands for *element-by-element* multiplication, *i.e.* $y_j \cdot y_k$ is a vector $(y_{j_1}y_{k_1}, y_{j_2}y_{k_2}, \dots, y_{j_{nm_{jk}}}y_{k_{nm_{jk}}})^T$.

For simplification, the matrices $A, B_1, B_2, \dots$ and the vector b are constant. The higher derivatives of the terms $B_1 y_{jk}, B_2 y_{jkl}$ in (8) can be included in a recurrent calculation of Taylor series' terms DY_{B1} and DY_{B2}

$$\begin{aligned} DY_{B1} &= h(B_1 y_{jk}), \quad DY_{B2} = h(B_2 y_{jkl}), \\ DY_{B1r} &= \frac{h}{r} \left(B_1 \sum_{p=1}^r DY_{j,r-p} \cdot DY_{k,p-1} \right), \\ DY_{B2r} &= \frac{h}{r} B_2 \sum_{q=0}^{r-1} DY_{jj,q} \cdot \left(\sum_{p=1}^r DY_{kk,r-p-q} \cdot DY_{ll,p-1} \right), \\ DY_r &= DY_{A_{r-1}} + DY_{B1_{r-1}} + DY_{B2_{r-1}}, \quad r = 2, \dots, n, \end{aligned} \quad (9)$$

where the linear term $DY_{A_{r-1}}$ is computed using (7). The Taylor series higher order multiplication terms $DY_{B3}, DY_{B4}, \dots$ can be calculated recurrently in a similar way.

3 Selected examples

In this section, several examples of linear and nonlinear problems are presented and solved using the MTSM. The MTSM was implemented in MATLAB software [8] using vectorization. The results are compared with state-of-the-art non-stiff solvers in MATLAB, specifically ode23, ode45 and ode113 [21].

All experiments were performed using MATLAB 2015a on the Salomon supercomputer [22]. Note that columns labeled as *Time of calculation* and *Ratio* are taken as a median value from 100 runs. Ratios of computation times $ratio = \text{ode}/\text{MTSM} \gg 1$ indicate significantly faster computation using the MTSM.

3.1 Movement of a charged particle

Movement of a charged particle in a electromagnetic and electrostatic field is given by Lorenz (electromagnetic) force [1, 23]

$$\mathbf{f} = q(\mathbf{e} + \mathbf{v}\mathbf{b}), \quad (10)$$

where $\mathbf{e} = (e_x, e_y, e_z)^T$ is the electric field, $\mathbf{b} = (b_x, b_y, b_z)^T$ is the magnetic field, vector $\mathbf{v}$ is instantaneous velocity of the charged particle with charge q . The force vector $\mathbf{f}$ can be substituted as $m\mathbf{a}$,

$$m\mathbf{a} = q(\mathbf{e} + \mathbf{v}\mathbf{b}). \quad (11)$$

We can further suppose that acceleration $\mathbf{a}$ in x , y and z axis (x'' , y'' , z'') can be calculated as

$$\begin{aligned} x'' &= \frac{q}{m} (e_x + y' b_z - z' b_y), \\ y'' &= \frac{q}{m} (e_y + z' b_x - x' b_z), \\ z'' &= \frac{q}{m} (e_z + x' b_y - y' b_x). \end{aligned} \quad (12)$$

By substituting $\mathbf{y} = (x', y', z', x, y, z)^T$, the problem can be transformed into a system of linear ODEs in a matrix-vector representation (5),

$$\mathbf{A} = \begin{pmatrix} 0 & \frac{q}{m} b_z & -\frac{q}{m} b_y & 0 & 0 & 0 \\ -\frac{q}{m} b_z & 0 & \frac{q}{m} b_x & 0 & 0 & 0 \\ \frac{q}{m} b_y & -\frac{q}{m} b_x & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \quad (13)$$

where initial conditions are $\mathbf{y}_0 = (-8 \cdot 10^7, 0, 0, 0, 0, 0)^T$. Constants are set to $m = 9.10938356 \cdot 10^{-31}$ [kg], $q = -1.6 \cdot 10^{-19}$ [C], $b_x = b_y = 0$, $b_z = 0.5$ [T] and $e_x = e_y = e_z = 0$ [T]. Further, the right-hand side $\mathbf{b} = (\frac{q}{m} e_x, \frac{q}{m} e_y, \frac{q}{m} e_z, 0, 0, 0)^T = \mathbf{0}$. The behavior of the electron is in Figure 1.

Figure 1: Position of a particle (upper) and trajectory of the particle (lower).

Results of simulations using IVP (13) for the given parameters of the electromagnetic and electrostatic fields and time of simulation $t \in (0, 10^{-8})$ are in Table 1. The MTSM (with fixed integration step size $h = 10^{-10}$) calculates the problem faster than the state-of-the-art MATLAB ode solvers. The order of the method is $ORD = 51 \pm 1$.

Table 1: Results of calculations for movement of a particle.

Solver	Time of calculation [s]	Ratio	Number of steps
MTSM	0.017836	–	100
ode23	193.097	10826.2	2235540
ode45	4.83689	2712.2	213925
ode113	0.78659	44.1	9446

3.2 Fourier coefficients

A Fourier series is the limit of the sequence of trigonometric polynomials that have a cosine part and a sine part. They are mainly used in the study of phenomena with a periodic character. The advantage of the Fourier series is that the requirements for its convergence to the developed function is weaker than in the case of the Taylor series. For example, the existence of derivatives of all orders of a given function at a given point is not required. The calculation of coefficients can be (especially when using numerical methods) easier than for the Taylor series.

Any periodic function $f(t)$ can be written as a sum of harmonics (14) called the Fourier series

$$f(t) = \frac{a_0}{2} + a_1 \cos(\omega t) + a_2 \cos(2\omega t) + \dots + b_1 \sin(\omega t) + b_2 \sin(2\omega t) + \dots, \quad (14)$$

where

$$a_0 = \frac{2}{T} \int_0^T f(t) dt, \quad (15)$$

$$a_k = \frac{2}{T} \int_0^T f(t) \cos(k\omega t) dt, \quad k = 1, 2, 3, \dots, n, \quad (16)$$

$$b_k = \frac{2}{T} \int_0^T f(t) \sin(k\omega t) dt, \quad k = 1, 2, 3, \dots, n, \quad (17)$$

where T is the period of the function $f(t)$. The calculation of the Fourier series' coefficients requires the calculation of definite integrals. The definite integral

$$Y = \int_0^T f(x) dx \quad (18)$$

can be transformed into the differential equation

$$y' = f(x) \quad y(0) = 0, \quad (19)$$

where $y(T) = Y$ and T denotes the maximum time of simulation $t \in (0, T)$. Note that the same approach can be used for the calculation of multiple definite integrals. More details about transformation and numerical solution of definite integrals using the MTSM can be found in [24].

For example, the analytic calculation of coefficient A_2 using the definite integral

$$A_2 = \frac{2}{T} \int_0^T f(t) \cos(2\omega t) dt \quad (20)$$

can be transformed into the IVP

$$a'_2 = \frac{2}{T} f(t) \cos(2\omega t), \quad a_2(0) = 0. \quad (21)$$

The solution of the IVP (21) at the maximum time T represents the calculated value of the definite integral (20) ($A_0 \approx a_0(T)$, $A_2 \approx a_2(T)$).

As an example, consider the simple harmonic function

$$f(t) = \sin^2(\omega t) \quad (22)$$

with parameters $\omega = \frac{2\pi}{T}$ [rad · s], $T = 2$ [s]. The coefficients of the Fourier series of (22) given by

$$\sin^2(\omega t) = \frac{a_0}{2} + a_2 \cos(2\omega t) \quad (23)$$

can be calculated analytically

$$\sin^2(\omega t) = \frac{1}{2} - \frac{1}{2} \cos(2\omega t) \quad (24)$$

as $A_0 = 1$ and $A_2 = -\frac{1}{2}$. The coefficients can be calculated numerically using the following system of ODEs

$$\begin{aligned} a'_0 &= f(t) & a_0(0) &= 0 \\ a'_2 &= \frac{2}{T} f(t) \cos(2\omega t) & a_2(0) &= 0. \end{aligned} \quad (25)$$

The solution of the IVP in $t \in (0, T)$, where $T = 2$ [s] is in Figure 2. The final values of functions $a_0(t)$ and $a_2(t)$ show the calculated values of Fourier coefficients $a_0(T) = 1$, $a_2(T) = -0.5$.

Figure 2: Fourier coefficients of $f(t) = \sin^2(\omega t)$.

The IVP (25) can be transformed into different autonomous systems with auxiliary generating equations. The first simple substitution $\mathbf{y} = (a_0, a_2, \sin(\omega t), \cos(\omega t), \sin(2\omega t), \cos(2\omega t))^T$ leads to the following nonlinear sys-

tem of ODEs

$$\begin{aligned}
 y_1' &= y_3^2 & y_1(0) &= 0 \\
 y_2' &= \frac{2}{T} y_3^2 y_6 & y_2(0) &= 0 \\
 y_3' &= \omega y_4 & y_3(0) &= 0 \\
 y_4' &= -\omega y_3 & y_4(0) &= 1 \\
 y_5' &= 2\omega y_6 & y_5(0) &= 0 \\
 y_6' &= -2\omega y_5 & y_6(0) &= 1.
 \end{aligned} \tag{26}$$

The first experiments with (26) with multiplication of 3 terms ($y_3^2 y_6$) in the second equation were performed in [2]. The more effective way of solving the IVP (25) is to transform it into the autonomous system with fewer multiplications or even better to the linear system of ODEs. The next substitution $\mathbf{y} = (a_0, a_2, \sin^2(\omega t), \sin(\omega t) \cos(\omega t), \cos^2(\omega t), \cos(2\omega t), \sin(2\omega t))^T$ leads to the following non-linear system of ODEs with at most two term multiplication

$$\begin{aligned}
 y_1' &= y_3 & y_1(0) &= 0 \\
 y_2' &= \frac{2}{T} y_3 y_6 & y_2(0) &= 0 \\
 y_3' &= 2\omega y_4 & y_3(0) &= 0 \\
 y_4' &= \omega(y_5 - y_3) & y_4(0) &= 0 \\
 y_5' &= -2\omega y_4 & y_5(0) &= 1 \\
 y_6' &= -2\omega y_7 & y_6(0) &= 1 \\
 y_7' &= 2\omega y_6 & y_7(0) &= 0.
 \end{aligned} \tag{27}$$

The last transformation of the IVP (25) using substitution $\mathbf{y} = (a_0, a_2, \sin^2(\omega t), \sin(\omega t) \cos(\omega t), \cos^2(\omega t), \sin^2(\omega t) \cos(2\omega t), \sin^2(\omega t) \sin(2\omega t), \sin(\omega t) \cos(\omega t) \cos(2\omega t), \sin(\omega t) \cos(\omega t) \sin(2\omega t), \cos^2(\omega t) \cos(2\omega t), \cos^2(\omega t) \sin(2\omega t))^T$ leads to the linear IVP

$$\begin{aligned}
 y_1' &= y_3 & y_1(0) &= 0 \\
 y_2' &= \frac{2}{T} y_6 & y_2(0) &= 0 \\
 y_3' &= 2\omega y_4 & y_3(0) &= 0 \\
 y_4' &= \omega(y_5 - y_3) & y_4(0) &= 0 \\
 y_5' &= -2\omega y_4 & y_5(0) &= 1 \\
 y_6' &= 2\omega(y_8 - y_7) & y_6(0) &= 0 \\
 y_7' &= 2\omega(y_9 + y_6) & y_7(0) &= 0 \\
 y_8' &= \omega(y_{10} - y_6 - 2y_9) & y_8(0) &= 0 \\
 y_9' &= \omega(y_{11} - y_7 + 2y_8) & y_9(0) &= 0 \\
 y_{10}' &= -2\omega(y_8 + y_{11}) & y_{10}(0) &= 1 \\
 y_{11}' &= -2\omega(y_9 - y_{10}) & y_{11}(0) &= 0.
 \end{aligned} \tag{28}$$

Results of the calculation of the Fourier coefficients using IVPs (27) and (28) are in Tables 2 and Table 3, respectively. The tolerance for all solvers was set to 10^{-10}

Table 2: Results of calculations for Fourier coefficients, Eq. (27).

Solver	Time of calculation [s]	Ratio	$\ error\ $
MTSM	0.00135245	–	2.7e–15
ode23	0.392118	289.9	7.2e–10
ode45	0.0294324	21.8	8.3e–11
ode113	0.0122999	9.1	1.4e–11

Table 3: Results of calculations for Fourier coefficients, Eq. (28).

Solver	Time of calculation [s]	Ratio	$\ error\ $
MTSM	0.0006809	–	1.1e–11
ode23	0.533741	783.9	2.9e–13
ode45	0.0456895	67.1	1.8e–12
ode113	0.0159332	23.4	1.6e–10

with fixed integration step size for the MTSM solver set at $h = 0.4$ in all cases. The column labeled $\|error\|$ shows the norm of the error between analytic (A_0, A_2) and numerical solution ($a_0(T), a_2(T)$). The results show that the MTSM is faster and more accurate than all tested state-of-the-art solvers. The average order used by the MTSM method is $ORD = 25 \pm 1$.

3.3 Kepler problem

The Kepler problem describes the motion of a single planet around a heavy sun [25]. The problem is simplified so that the planet does not gravitationally influence the sun and the sun is treated as stationary. The motion of the planet is also limited to the plane (it does not move up or down).

Let $y_1(t)$ and $y_2(t)$ denote rectangular coordinates centered at the sun, specifying at time t the position of the planet. Also let $y_3(t)$ and $y_4(t)$ denote a components of velocity in the y_1 and y_2 directions. If M denotes the mass of the sun, γ the gravitational constant and m mass of the planet, then the attractive force will have the magnitude

$$\frac{\gamma M m}{y_1^2 + y_2^2}.$$

The accelerations of the planet in y_1 and y_2 directions can be calculated as follows:

$$\begin{aligned}
 &\frac{-\gamma M y_1}{(y_1^2 + y_2^2)^{\frac{3}{2}}} \\
 &\frac{-\gamma M y_2}{(y_1^2 + y_2^2)^{\frac{3}{2}}},
 \end{aligned} \tag{29}$$

where the negative sign denotes the inward direction of the acceleration. By adjusting the scales of the variables, the

factor γM can be removed. The system of ODEs representing the motion is as follows:

$$\begin{aligned} y_1' &= y_3 & y_1(0) &= 1 - e \\ y_2' &= y_4 & y_2(0) &= 0 \\ y_3' &= \frac{-y_1}{r^3} & y_3(0) &= 0 \\ y_4' &= \frac{-y_2}{r^3} & y_4(0) &= \sqrt{\frac{1+e}{1-e}}, \end{aligned} \quad (30)$$

where e is an eccentricity of a rotating body, and $r = \sqrt{y_1^2 + y_2^2}$. The solution of this system is known and can be either ellipses, parabolas or hyperbolas.

The analytical solution of the Kepler problem (30) is defined by the following ellipse

$$(y_1 + e)^2 + \frac{y_2^2}{1 - e^2} - 1 = 0. \quad (31)$$

The trajectory for $e = 0.75$ is in Figure 6 (right). The change in y_1 and y_2 coordinates in time for different values of e is in Figure 3. The first experiments with solution of the system (30) using the MTSM were performed in [26].

To solve the system (30) using the MTSM with recurrent calculation of the Taylor series' terms (9) the system of ODEs has to be transformed to the new autonomous equivalent system (8).

The term $\frac{1}{r^3}$ has to be replaced by the equivalent generating ODEs that are added to (30). First, the term r^3 is substituted to give

$$\begin{aligned} y_5 &= \left(\sqrt{y_1^2 + y_2^2} \right)^3 = (y_1^2 + y_2^2)^{\frac{3}{2}} \\ y_5' &= \frac{3}{2} (y_1^2 + y_2^2)^{\frac{1}{2}} (2y_1 y_1' + 2y_2 y_2') \\ &= 3(y_1^2 + y_2^2)^{\frac{1}{2}} (y_1 y_3 + y_2 y_4). \end{aligned}$$

Then the substitution of the term $(y_1^2 + y_2^2)^{\frac{1}{2}}$ is introduced to give

$$\begin{aligned} y_6 &= (y_1^2 + y_2^2)^{\frac{1}{2}} \\ y_6' &= \frac{1}{2} (y_1^2 + y_2^2)^{-\frac{1}{2}} (2y_1 y_1' + 2y_2 y_2') \\ &= (y_1^2 + y_2^2)^{-\frac{1}{2}} (y_1 y_3 + y_2 y_4) \\ &= \frac{(y_1 y_3 + y_2 y_4)}{(y_1^2 + y_2^2)^{\frac{1}{2}}} \\ &= \frac{(y_1 y_3 + y_2 y_4)}{y_6}. \end{aligned}$$

After the previous substitution, the terms that contain operation division (y_5^{-1} and y_6^{-1}) can be removed from the system using the following auxiliary ODEs

$$y_7 = \frac{1}{y_5} = y_5^{-1}$$

Figure 3: Kepler problem, solution: $e = 0.25$ (left), $e = 0.5$ (middle), $e = 0.75$ (right).

Figure 4: Kepler problem, MTSM ORD function: $e = 0.25$ (left), $e = 0.5$ (middle), $e = 0.75$ (right).

Figure 5: Kepler problem, stiffness indicator σ : $e = 0.25$ (left), $e = 0.5$ (middle), $e = 0.75$ (right).

$$\begin{aligned}
 y_7' &= -y_5^{-1}y_5' \\
 &= -y_7^2y_5' = -y_7^2(3y_6(y_1y_3 + y_2y_4)) \\
 &= -3y_6y_7^2(y_1y_3 + y_2y_4) \\
 y_8 &= \frac{1}{y_6} = y_6^{-1} \\
 y_8' &= -y_6^{-2}y_6' \\
 &= -y_8^2y_6' = -y_8^2(y_8(y_1y_3 + y_2y_4)) \\
 &= -y_8^3(y_1y_3 + y_2y_4).
 \end{aligned}$$

The new autonomous system of ODEs with the substituted equations without operation division is obtained to give

$$\begin{aligned}
 y_1' &= y_3 \\
 y_2' &= y_4 \\
 y_3' &= -y_7y_1 \\
 y_4' &= -y_7y_2 \\
 y_5' &= 3y_6(y_1y_3 + y_2y_4) \\
 y_6' &= y_8(y_1y_3 + y_2y_4) \\
 y_7' &= -3y_6y_7^2(y_1y_3 + y_2y_4) \\
 y_8' &= -y_8^3(y_1y_3 + y_2y_4).
 \end{aligned}$$

The correct initial values for all auxiliary ODEs follow: $\mathbf{y}(0) = (1 - e, 0, 0, \sqrt{\frac{1+e}{1-e}}, (\sqrt{y_1(0)^2 + y_2(0)^2})^3, \sqrt{y_1(0)^2 + y_2(0)^2}, \frac{1}{y_5(0)}, \frac{1}{y_6(0)})^T$. The new IVP (32) is equivalent to the original Kepler problem (30).

Note that the new IVP (32) is an autonomous system of ODEs with operation multiplication in the form (8), and such system can be solved using the MTSM with recurrent calculation of the Taylor series' terms using (9).

The matrix-vector representation (8) of the system (32) is as follows:

- the 8×8 matrix $\mathbf{A}$ has two non-zero elements $A(1, 3) = 1, A(2, 4) = 1$,
- the 2×8 matrix $\mathbf{B}_1$ has two nonzero elements $B_1(3, 1) = -1, B_1(4, 2) = -1$,
- the 8×4 matrix $\mathbf{B}_2$ has four nonzero elements $B_2(5, 1) = 3, B_2(5, 2) = 3, B_2(6, 3) = 1, B_2(6, 4) = 1$,

- the 8×4 matrix $\mathbf{B}_4$ is defined with four nonzero elements $B_4(7, 1) = -3, B_4(7, 2) = -3, B_4(8, 3) = -1, B_4(8, 4) = -1$.

The vectors on the right hand side of (8) are defined as follows:

$$\mathbf{y}_{jk} = \begin{pmatrix} y_1y_7 \\ y_2y_7 \end{pmatrix} \quad \mathbf{y}_{jkl} = \begin{pmatrix} y_6y_1y_3 \\ y_6y_2y_4 \\ y_1y_3y_8 \\ y_2y_4y_8 \end{pmatrix}$$

$$\mathbf{y}_{jklmn} = \begin{pmatrix} y_6y_1y_3y_7y_7 \\ y_6y_2y_4y_7y_7 \\ y_1y_3y_8y_8y_8 \\ y_2y_4y_8y_8y_8 \end{pmatrix}.$$

The system of ODEs (32) can be further simplified by reducing the number of multiplications. By substituting $(y_1y_3 + y_2y_4)$ we obtain another auxiliary ODE:

$$\begin{aligned}
 y_9 &= y_1y_3 + y_2y_4 \\
 y_9' &= y_1' y_3 + y_1 y_3' + y_2' y_4 + y_2 y_4' \\
 &= y_3^2 + y_4^2 - y_1^2 y_7 - y_2^2 y_7.
 \end{aligned}$$

Adding the new equation into (32) decreases the number of multiplications to four. We obtain the following:

$$\begin{aligned}
 y_1' &= y_3 \\
 y_2' &= y_4 \\
 y_3' &= -y_7y_1 \\
 y_4' &= -y_7y_2 \\
 y_5' &= 3y_6y_9 \\
 y_6' &= y_8y_9 \\
 y_7' &= -3y_6y_7^2y_9 \\
 y_8' &= -y_8^3y_9 \\
 y_9' &= y_3^2 + y_4^2 - y_7(y_1^2 + y_2^2).
 \end{aligned} \tag{33}$$

Again, the correct initial values for all auxiliary ODEs are added: $\mathbf{y}(0) = (1 - e, 0, 0, \sqrt{\frac{1+e}{1-e}}, (\sqrt{y_1(0)^2 + y_2(0)^2})^3, \sqrt{y_1(0)^2 + y_2(0)^2}, \frac{1}{y_5(0)}, \frac{1}{y_6(0)}, \frac{1}{y_9(0)})^T$.

$y_1(0)y_3(0) + y_2(0)y_4(0))^T$. The new IVP (33) is equivalent to the original Kepler problem (30) and with (32).

The matrix-vector representation of the ODEs system (33) using the notation of (8) is as follows:

- the 9×9 matrix $\mathbf{A}$ has two non-zero elements $A(1, 3) = 1, A(2, 4) = 1$,
- the 9×9 matrix $\mathbf{B}_1$ has eight nonzero elements $B_1(3, 1) = -1, B_1(4, 2) = -1, B_1(5, 3) = 3, B_1(6, 4) = 1, B_1(9, 5) = 1, B_1(9, 6) = 1, B_1(9, 7) = -1, B_1(9, 8) = -1$,
- the 9×2 matrix $\mathbf{B}_2$ has two nonzero elements $B_2(9, 1) = -1, B_2(9, 2) = -1$,
- the 9×2 matrix $\mathbf{B}_3$ has two nonzero elements $B_3(7, 1) = -3, B_3(8, 2) = -1$.

The vectors on the right hand side of (8) are defined as follows:

$$\mathbf{y}_{jk} = \begin{pmatrix} y_1 y_7 \\ y_2 y_7 \\ y_6 y_9 \\ y_8 y_9 \\ y_3 y_3 \\ y_4 y_4 \\ y_1 y_7 \\ y_2 y_7 \end{pmatrix} \quad \mathbf{y}_{jkl} = \begin{pmatrix} y_7 y_1 y_1 \\ y_7 y_2 y_2 \end{pmatrix} \quad \mathbf{y}_{jklm} = \begin{pmatrix} y_6 y_7 y_7 y_9 \\ y_8 y_8 y_8 y_9 \end{pmatrix}.$$

The number of multiplications can be decreased even further by simplifying the remaining ODEs with four multiplications. The multiplication $y_6 y_9$ can be substituted by an auxiliary ODE:

$$\begin{aligned} y_{10} &= y_6 y_9 \\ y'_{10} &= y'_6 y_9 + y_6 y'_9 \\ &= y_8 y_9 y_9 + y_6 y_3^2 + y_6 y_4^2 - y_6 y_7 (y_1^2 + y_2^2). \end{aligned}$$

A new multiplication ($y_8 y_9$) in the auxiliary ODE can also be substituted, to give

$$\begin{aligned} y_{11} &= y_8 y_9 \\ y'_{11} &= y'_8 y_9 + y_8 y'_9 \\ &= -y_8^3 y_9^2 + y_8 (y_3^2 + y_4^2 - y_7 (y_1^2 + y_2^2)) \\ &= -y_8 y_{11}^2 + y_8 (y_3^2 + y_4^2 - y_7 (y_1^2 + y_2^2)). \end{aligned}$$

Finally, the bracket ($y_1^2 + y_2^2$) can be substituted to give

$$\begin{aligned} y_{12} &= y_1^2 + y_2^2 \\ y'_{12} &= 2y_1 y'_1 + 2y_2 y'_2 = 2y_1 y_3 + 2y_2 y_4 \\ &= 2y_1 y_3 + 2y_2 y_4. \end{aligned}$$

Adding the three new auxiliary ODEs into (33) decreases the number of multiplications to three

$$\begin{aligned} y'_1 &= y_3 \\ y'_2 &= y_4 \\ y'_3 &= -y_7 y_1 \\ y'_4 &= -y_7 y_2 \\ y'_5 &= 3y_{10} \\ y'_6 &= y_{11} \\ y'_7 &= -3y_7^2 y_{10} \\ y'_8 &= -y_8^2 y_{11} \\ y'_9 &= y_3^2 + y_4^2 - y_7 y_{12} \\ y'_{10} &= y_9 y_{11} + y_6 y_3^2 + y_6 y_4^2 - y_6 y_7 y_{12} \\ y'_{11} &= -y_8 y_{11}^2 + y_8 y_3^2 + y_8 y_4^2 - y_8 y_7 y_{12} \\ y'_{12} &= 2y_1 y_3 + 2y_2 y_4. \end{aligned} \quad (34)$$

The correct initial values to all auxiliary ODEs are added as follows: $\mathbf{y}(0) = (1 - e, 0, 0, \sqrt{\frac{1+e}{1-e}}, (\sqrt{y_1(0)^2 + y_2(0)^2})^3, \sqrt{y_1(0)^2 + y_2(0)^2}, \frac{1}{y_5(0)}, \frac{1}{y_6(0)}, y_1(0)y_3(0) + y_2(0)y_4(0), y_6(0)y_9(0), y_8(0)y_9(0), y_1(0)^2 + y_2(0)^2)^T$. The new IVP (34) is equivalent to all previous systems ((30), (32) and (33)).

The matrix-vector representation of the ODEs system (34) in the (8) notation follows:

- the 12×12 matrix $\mathbf{A}$ has four non-zero elements $A(1, 3) = 1, A(2, 4) = 1, A(5, 10) = 3, A(6, 11) = 1$,
- the 12×8 matrix $\mathbf{B}_1$ has eight nonzero elements $B_1(3, 1) = -1, B_1(4, 2) = -1, B_1(9, 3) = 1, B_1(9, 4) = 1, B_1(9, 5) = -1, B_1(10, 6) = 1, B_1(12, 7) = 2, B_1(12, 8) = 2$,
- the 12×9 matrix $\mathbf{B}_2$ has nine nonzero elements $B_2(7, 1) = -3, B_2(8, 2) = -1, B_2(10, 3) = 1, B_2(10, 4) = 1, B_2(10, 5) = -1, B_2(11, 6) = -1, B_2(11, 7) = 1, B_2(11, 8) = 1, B_2(11, 9) = -1$.

The vectors on the right hand side of (8) are defined as follows:

$$\mathbf{y}_{jk} = \begin{pmatrix} y_1 y_7 \\ y_2 y_7 \\ y_3 y_3 \\ y_4 y_4 \\ y_7 y_{12} \\ y_7 y_{11} \\ y_1 y_3 \\ y_2 y_4 \end{pmatrix} \quad \mathbf{y}_{jkl} = \begin{pmatrix} y_7 y_7 y_{10} \\ y_8 y_8 y_{11} \\ y_6 y_3 y_3 \\ y_6 y_4 y_4 \\ y_6 y_7 y_{12} \\ y_8 y_{11} y_{11} \\ y_8 y_3 y_3 \\ y_8 y_4 y_4 \\ y_8 y_7 y_{12} \end{pmatrix}.$$

Figure 6: Kepler problem, circuit test for the solvers (from the left): ode23, ode45, ode113, MTSM.

Table 4: Results for Kepler problem, Eq. (32).

	ode23	ode45	ode113	expTay
e	ratio	ratio	ratio	[s]
0.25	146.4	4.7	1.3	0.0293
0.5	122.8	4.1	1.0	0.0536
0.75	90.4	3.0	0.7	0.1158

Table 5: Results for Kepler problem, Eq. (33).

	ode23	ode45	ode113	expTay
e	ratio	ratio	ratio	[s]
0.25	213.9	6.8	1.8	0.0205
0.5	174.4	5.7	1.5	0.0377
0.75	145.1	4.8	1.2	0.073

Table 6: Results for Kepler problem, Eq. (34).

	ode23	ode45	ode113	expTay
e	ratio	ratio	ratio	[s]
0.25	312.2	12.1	2.5	0.0144
0.5	262.5	9.1	2.1	0.0263
0.75	221.3	7.4	1.7	0.0509

The benchmark results of (32), (33), (34) for the MTSM with fixed integration step size $h_{0.25} = \pi/25$, $h_{0.5} = \pi/50$, $h_{0.75} = \pi/100$ and $e = 0.25$, $e = 0.5$, $e = 0.75$ are in Tables 4, 5 and 6, respectively. The tolerances for all solvers were set to obtain the $\|error\| = 10^{-8}$, where the $\|error\|$ is computed as the norm of (31). The maximum time of the simulation is set to 2 cycles, $T = 4 \cdot \pi$ [s]. The obtained ratios indicate faster computation of the MTSM in most cases ($ratio > 1$).

The autonomous system with low number of multiplications (34) is the fastest one, see Table 6. The greater number of multiplications slows down the calculation considerably. The ratios between the slowest system (32) and the systems of ODEs (33) and (34) for different values of e are in Table 7. The table shows that the system with three

Table 7: Comparison between times of calculation of autonomous system (32) and systems (33), (34).

	expTay (32)/(33)	expTay (32)/(34)
e	ratio	ratio
0.25	1.43	2.03
0.5	1.42	2.04
0.75	1.59	2.28

Table 8: Number of integration steps, Kepler problem (34).

	number of steps			
e	ode23	ode45	ode113	expTay
0.25	105180	8916	419	100
0.5	161323	12148	654	200
0.75	262631	19288	955	400

term multiplication (34) is approximately two times faster than the system with five term multiplication (32). The difference is caused by the large number of multiplications of higher order terms in the Taylor series. Currently, the Taylor series' terms are computed directly using (9) without any optimization to decrease the number of multiplications.

Table 8 shows the number of integration steps for different values of e . The number of integration steps grows with increasing values of the constant e . The Kepler problem becomes *stiff* for a larger constant e . Explicit numerical methods have difficulties with such systems. The stiffness indicator σ for the time normalized Kepler problem with different values of constant e is in Figure 5 [27].

The MTSM automatically detects the stiffness in the system and uses larger order (more terms of the Taylor series) in rapidly changing parts of solutions, see Figures 3 and Figure 4.

The last experiment shows that the solution of the Kepler problem becomes unstable using the state-of-the-art solvers. Results for $T = 200 \cdot \pi$ [s] and $e = 0.75$ are in Figure 6. The default error control tolerance $Tol = 10^{-6}$ was

set for all solvers. Only the MTSM provides a stable solution, see Figure 6 (right).

4 Conclusion

The article dealt with the numerical solution of both linear and nonlinear IVPs represented by a system of ODEs. The results showed that the MTSM overperforms state-of-the-art MATLAB ode solvers in most cases and that the method is more stable in some cases. Future research will focus on further improvements and optimizations of the nonlinear MTSM solver.

Acknowledgement: This research was financially supported by the Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project “IT4Innovations excellence in science - LQ1602”. The paper also includes the results of the internal BUT FIT project FIT-S-20-6427.

References

- [1] Kunovský J., Modern Taylor Series Method, Habilitation work, Faculty of Electrical Engineering and Computer Science, Brno University of Technology, 1994
- [2] Veigend P., Nečasová G., Šátek V., High Order Numerical Integration Method and its Applications – The First 36 Years of MTSM, in 2019 IEEE 15th International Scientific Conference on Informatics, 2019, 288–293
- [3] Veigend P., Nečasová G., Šátek V., Model of the telegraph line and its numerical solution, Open Computer Science, 8, in press, 2018, 1–8, <https://doi.org/10.1515/comp-2018-0002>
- [4] Kocina F., Nečasová G., Veigend P., Šátek V., Kunovský J., Parallel Solution of Higher Order Differential Equations, in 14th International Conference on High Performance Computing & Simulation, Institute of Electrical and Electronics Engineers, 2016, ISBN 978-1-5090-2088-1
- [5] Veigend P., Šátek V., Nečasová G., Model of the Telegraph line and its Numerical Solution, Open Computer Science, 8(1), 2018, 10–17, [10.1515/comp-2018-0002](https://doi.org/10.1515/comp-2018-0002)
- [6] Hairer E., Nørsett S.P., Wanner G., Solving Ordinary Differential Equations I, vol. Nonstiff Problems. Springer-Verlag Berlin Heidelberg, 1987, ISBN 3-540-56670-8
- [7] Kunovský J., TKSL, <https://www.fit.vut.cz/research/product/51/en>
- [8] MATLAB and Simulink software, <http://www.mathworks.com>
- [9] Kocina F., FOS WEBGUI, http://www.fit.vutbr.cz/~iveigend/fos/index_en.php
- [10] Šátek V., High Performance Computing Research Group, <https://www.fit.vut.cz/research/group/hpc/en>
- [11] Kocina F., Kunovský J., Nečasová G., Šátek V., Veigend P., Parallel solution of higher order differential equations, in Proceedings of the 2016 International Conference on High Performance Computing & Simulation (HPCS 2016), Institute of Electrical and Electronics Engineers, 2016, 302–309
- [12] Barrio R., Rodríguez M., Abad A., Blesa F., TIDES: A free software based on the Taylor series method, Monografías de la Real Academia de Ciencias de Zaragoza, 35, 2011, 83–95
- [13] Jorba A., Zou M., A software package for the numerical integration of ODE by means of high-order Taylor methods, in Exp. Math., volume 14, 2005, 99–117
- [14] Chang Y.F., Corliss G., ATOMF: solving ODEs and DAEs using Taylor series, Computers Math. Applic., 28, 1994, 209–233
- [15] Berz M., COSY INFINITY version 8 reference manual, Technical Report MSUCL-1088, National Superconducting Cyclotron Lab., Michigan State University, East Lansing, Mich., 1997
- [16] Nedialkov N.S., Pryce J., Solving differential algebraic equations by Taylor series III. The DAETS code, JNAIAM J. Numer. Anal. Ind. Appl. Math., 3, 2008, 61–80
- [17] Barrio R., Blesa F., Lara M., VSVO Formulation of the Taylor Method for the Numerical Solution of ODEs, in Computers and Mathematics with Applications, volume 50, 2005, 93–111
- [18] Barrio R., Performance of the Taylor series method for ODEs/DAEs, in Applied Mathematics and Computation, volume 163, 2005, 525–545, ISSN 00963003
- [19] Mohazzabi P., L. Becker J., Numerical Solution of Differential Equations by Direct Taylor Expansion, Journal of Applied Mathematics and Physics, 05(03), 2017, 623–630, [10.4236/jamp.2017.53053](https://doi.org/10.4236/jamp.2017.53053)
- [20] Baeza A., Boscarino S., Mulet P., Russo G., Zorío D., Approximate Taylor methods for ODEs, Computers & Fluids, 159, 2017, 156 – 166, <https://doi.org/10.1016/j.compfluid.2017.10.001>
- [21] MATLAB: Choose an ODE Solver, <https://www.mathworks.com/help/matlab/math/choose-an-ode-solver.html>
- [22] IT4I, SOLOMON Supercomputer, <https://docs.it4i.cz/salomon/introduction/>
- [23] Richard P., Feynman L., Robert B., Sands M., Feynman Lectures on Physics: The New Millennium Edition, Basic Books, 2015
- [24] Chaloupka J., Kocina F., Veigend P., Nečasová G., Šátek V., Kunovský J., Multiple Integral Computations, in 14th International Conference of Numerical Analysis and Applied Mathematics, 1863, American Institute of Physics, 2017, 1–4, [10.1063/1.4992650](https://doi.org/10.1063/1.4992650)
- [25] Butcher J.C., Numerical methods for ordinary differential equations, John Wiley & Sons, 2016
- [26] Sehnalová P., Šátek V., 35 Years of Taylor-Kunovsky Simulation Language, in 16th International Conference of Numerical Analysis and Applied Mathematics, volume 2116, American Institute of Physics, 2019, [10.1063/1.5114335](https://doi.org/10.1063/1.5114335)
- [27] Söderlind G., Jay L., Calvo M., Stiffness 1952–2012: Sixty years in search of a definition, Bit Numer Math, 55, 2015, 531–558, [10.1007/s10543-014-0503-3](https://doi.org/10.1007/s10543-014-0503-3)