Modern Taylor Series Method in Numerical Integration: Part 2

NEČASOVÁ Gabriela[(*)], **VEIGEND Petr**[(*)], **ŠÁTEK Václav**[(*),(**)]

[(*)]Brno University of Technology, Faculty of Information Technology
Božetěchova 2, 612 66, Brno, Czech Republic
E-mail: inecasova@fit.vut.cz; iveigend@fit.vut.cz; satek@fit.vut.cz
[(**)]IT4Innovations, VŠB Technical University of Ostrava
17. listopadu 15/2172, 708 33, Ostrava-Poruba, Czech Republic

Abstract: The paper deals with extremely exact, stable, and fast numerical solutions of systems of differential equations with initial condition – initial value problems. Systems of ordinary differential equations are solved using variable order, variable step-size Modern Taylor Series Method. The Modern Taylor Series Method is based on a recurrent calculation of the Taylor series terms for each time interval. Thus, the complicated calculation of higher order derivatives (much criticized in the literature) need not be performed but rather the value of each Taylor series term is numerically calculated.
The paper present the solution of linear and nonlinear problems. As a linear problem, the telegraph equation was chosen. As a nonlinear problem, the behavior of Lorenz system was analyzed. All experiments were performed using MATLAB software, the newly developed nonlinear solver that uses Modern Taylor Series Method was used. Both linear and nonlinear solvers were compared with state of the art solvers in MATLAB.

Keywords: Taylor series method, ordinary differential equations, technical initial value problems.

1 Introduction

The paper deals with the solution of technical initial value problems (IVPs) representing the problems which arise from common technical practice (especially from electrical and mechanical engineering). To solve technical IVPs means to find the numerical solution of the system of ordinary differential equations (ODEs).

The best-known and the most accurate method of calculating a new value of the numerical solution of ODE [7]

$$y' = f(t, y), \qquad y(t_0) = y_0\,, \tag{1}$$

is to construct the Taylor series in the form

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

where h is the integration step.

The Taylor series can be very effectively implemented as the variable-order, variable-step-size numerical method [20] – Modern Taylor Series Method (MTSM). The method is based on a recurrent calculation of the Taylor series terms for each integration step. Therefore, the complicated calculation of higher order derivatives (much criticized in the literature) does not need to be performed, but rather the value of each Taylor series term is numerically calculated [13]. Equation (2) can then be rewritten in the form (3)

$$y_{i+1} = DY_0 + DY_1 + DY_2 + \cdots + DY_n \,. \tag{3}$$

Theoretically, it is possible to compute the solution of homogeneous linear differential equations with constant coefficients with arbitrary order and with arbitrary accuracy. Let us denote the ORD as the function which changes during the computation and defines the number of Taylor series terms in the current integration step ($ORD_{i+1} = n$). The resulting system of linear equations can be effectively solved either sequentially or in parallel.

An important part of the method is an automatic integration order setting, i.e. using as many Taylor series terms as the defined accuracy requires. Thus it is common that the computation uses different numbers of Taylor series terms for different integration steps of constant length.

The following paper is divided into several sections, which consider concrete technical IVPs and usage of MTSM. In Section 2, the effective numerical solution of a system of linear ODEs using higher order MTSM is shown and the problem of Telegraph equation is analyzed. The Section 3 considers the solution of quadratic nonlinear ODEs and the nonlinear Lorenz attractor problem is discussed. All algorithms of MTSM are efficiently implemented in MATLAB software [15] using vectorization. Finally, the MTSM algorithms are compared with MATLAB ode solvers.

Several papers focus on computer implementations of the Taylor series method in a variable order and variable step context (see, for instance TIDES software [3], TAYLOR [10] including detailed description of a variable step size version, ATOMF [6], COSY INFINITY [4], DAETS [17]. The variable step-size variable-order scheme is also described in [2] and [1], where simulations on a parallel computer are shown. This paper follows the article [5].

2 Solution of linear ODEs

Equation (2) for linear systems of ODEs in the form $\boldsymbol{y}' = \mathbf{A}\boldsymbol{y} + \mathbf{b}$ could be rewritten

$$\boldsymbol{y}_{i+1} \quad = \quad \boldsymbol{y}_i + h(\mathbf{A}\boldsymbol{y}_i + \mathbf{b}) + \tfrac{h^2}{2!}\mathbf{A}(\mathbf{A}\boldsymbol{y}_i + \mathbf{b}) + \tfrac{h^n}{n!}\mathbf{A}^{(n-1)}(\mathbf{A}\boldsymbol{y}_i + \mathbf{b}) \,, \tag{4}$$

where $\mathbf{A}$ is the constant Jacobian matrix and $\mathbf{b}$ is the constant right-hand side.

Moreover, (4) can be rewritten in the form (3) where Taylor series terms could be computed recurrently

$$DY_0 = \boldsymbol{y}_i \,, \quad DY_1 = h(\mathbf{A}\boldsymbol{y}_i + \mathbf{b}), \quad DY_{j+1} = \frac{h}{j}\mathbf{A}DY_j \,, \quad j = 2, \ldots, n-1 \,. \tag{5}$$

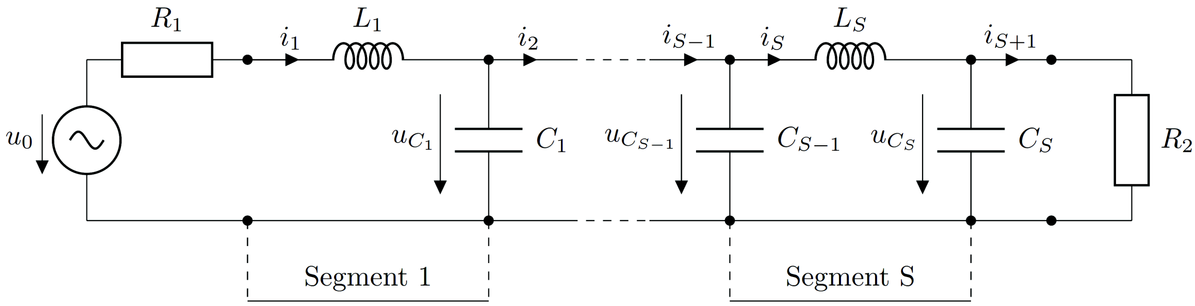


Figure 1: Model of the line−series of S segments

2.1 Telegraph equation

Let us solve the following electric circuit Figure 1, which represents a telegraph line [12,21]. The solution leads to the linear IVP

$$\boldsymbol{y}' = \mathbf{A}\boldsymbol{y} + \mathbf{b}, \quad \boldsymbol{y}(0) = \mathbf{y}_0\,, \tag{6}$$

where $\mathbf{A}$ is a matrix of constants (R, L, C parameters of circuit), $\boldsymbol{y}$ is a vector of variables (voltages and currents), $\mathbf{b}$ is a vector of constants and $\mathbf{y}_0$ is a vector of initial conditions. The block structure of matrix $\mathbf{A}$ and vectors $\boldsymbol{y}$ and $\mathbf{b}$ follows

$$\mathbf{A} = \left(\begin{array}{c|c} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \hline \mathbf{A}_{21} & \mathbf{A}_{22} \end{array}\right), \quad \boldsymbol{y} = \left(\begin{array}{c} u_{C_1} \\ \vdots \\ u_{C_S} \\ \hline i_1 \\ \vdots \\ i_S \end{array}\right), \quad \mathbf{b} = \left(\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline \frac{u_0}{L_1} \\ \vdots \\ 0 \end{array}\right), \tag{7}$$

where $\mathbf{A}_{11}$, $\mathbf{A}_{12}$, $\mathbf{A}_{21}$ and $\mathbf{A}_{22}$ are individual block matrices with size $S \times S$

$$\mathbf{A}_{11} = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{-1}{R_2 C_S} \end{pmatrix} \qquad \mathbf{A}_{12} = \begin{pmatrix} \frac{1}{C_1} & \frac{-1}{C_1} & 0 & \cdots & \cdots & 0 \\ 0 & \frac{1}{C_2} & \frac{-1}{C_2} & 0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \frac{1}{C_S} \end{pmatrix}$$

$$\mathbf{A}_{21} = \begin{pmatrix} \frac{-1}{L_1} & 0 & 0 & \cdots & \cdots & 0 \\ \frac{1}{L_2} & \frac{-1}{L_2} & 0 & 0 & \cdots & \vdots \\ 0 & \frac{1}{L_3} & \frac{-1}{L_3} & 0 & \cdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \frac{1}{L_S} & \frac{-1}{L_S} \end{pmatrix} \quad \mathbf{A}_{22} = \begin{pmatrix} \frac{-R_1}{L_1} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}.$$

For our experiments the capacitances and inductances are the same, $C = C_1 = \cdots = C_S = 1\,\mathrm{pF}$ and $L = L_1 = \cdots = L_S = 10\,\mathrm{nH}$. Moreover the transmission line is adjusted if $R_1 = R_2 = \sqrt{L/C} = 100\,\Omega$. The angular velocity is set $\omega = 3 \cdot 10^9 \mathrm{rad/s}$. The input voltage u_0 should be generally constant (DC circuit) or harmonic (AC circuit) signal. In the case of DC circuit the input voltage u_0 is hidden in constant right hand side $\mathbf{b}$, see (7). In the case of AC circuit the input voltage $u_0 = U_0 \sin(\omega t)$ can be computed using auxiliary system of coupled linear ODEs

$$\begin{aligned} u_0' &= \omega x\,, & u_0(0) &= 0 \\ x' &= -\omega u_0\,, & x(0) &= U_0\,. \end{aligned} \tag{8}$$

In our example we use AC circuit with input voltage $u_0 = \sin(\omega t)$. The propagation constant per unit length of one segment for simple model of transmission line Figure 1 is known $t_{LC} = \sqrt{LC}$. Then the total delay of input signal could be computed as $t_{delay} = S\,t_{LC}$. The delay of output voltage $u_{C_{100}}$ for 100 segments is shown in Figure 2. The time of simulation was set $t_{max} = 2\,t_{delay}$ for all experiments.

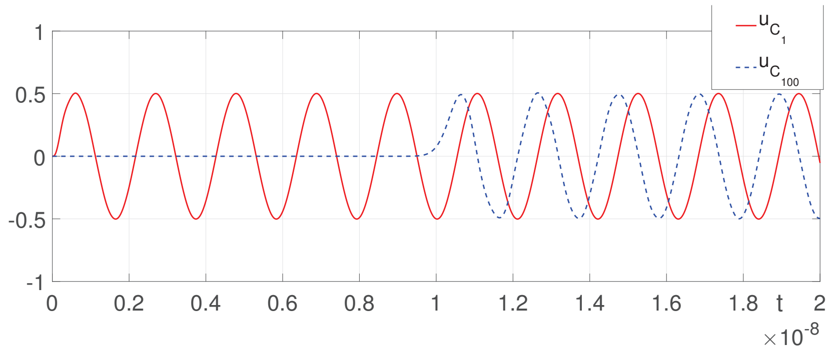


Figure 2: Delay of the signal on the transmission line with $S = 100$ segments

Vectorized MATLAB code of explicit Taylor series **expTay** with a variable order and variable step size scheme for linear systems of ODEs (6) has been implemented. This algorithm was tested on a set of examples of telegraph line with different number of segments S. The MTSM was compared with vectorized MATLAB explicit **ode** solvers. Both relative and absolute tolerances for all solvers were set to 10^{-7}. Benchmark results for MTSM with fixed number of integration steps $t_{max}/h = 200$ are shown in Table 1 and the results for MTSM with fixed h are shown in Table 2. Ratios of computation times *ratio* =**ode**/**expTay** > 1 indicate faster computation of the MTSM in all cases. Each reported runtime is taken as a median value of 100 computations. The MTSM average order ($mean(ORD)$) could be seen in the last columns of Table 1, 2. For the linearity and non-stiffness of the problem the ORD function was oscillating $mean(ORD) \pm 2$ during the computation.

Table 1: Time of solutions: explicit Taylor **expTay** and MATLAB explicit **ode** solver comparison; **expTay** with fixed number of steps $t_{max}/h = 200$

	ode23	**ode45**	**ode113**	**expTay**	**expTay**
S	*ratio*	*ratio*	*ratio*	[s]	$mean(ORD)$
200	18.9	7.1	6.2	0.056	20
600	18.6	9.2	6.9	0.199	41
1000	15.7	7.3	4.6	0.538	62
1400	15.9	7.5	4.1	1.005	83
1800	8.2	5.3	2.6	2.738	113

Table 2: Time of solutions: explicit Taylor **expTay** and MATLAB explicit **ode** solver comparison; **expTay** with fixed integration time step $h = 8 \cdot 10^{-10}$

S	**ode23** *ratio*	**ode45** *ratio*	**ode113** *ratio*	**expTay** [s]	**expTay** *mean(ORD)*
200	30.7	11.9	10.4	0.033	52
600	20.3	10.1	7.5	0.181	51
1000	13.6	6.4	4.1	0.606	51
1400	11.7	5.6	3.1	1.334	51
1800	8.8	5.5	2.6	2.498	51

More comparisons of MTSM numerical solutions of linear ODEs systems could be found in [16, 18].

3 Solution of nonlinear (quadratic) ODEs

In this section, the effective solution of special case of nonlinear quadratic systems of ODEs is described. The nonlinear quadratic system of ODEs is any first-order ODE that is quadratic in the unknown function. For such system Taylor series based numerical method can be implemented in very effective way.

Equation (1) for nonlinear-quadratic systems of ODEs can be rewritten in the form

$$\boldsymbol{y}' = \mathbf{A}\boldsymbol{y}^2 + \mathbf{B}\boldsymbol{y_{jk}} + \mathbf{C}\boldsymbol{y} + \mathbf{b}\,, \quad \boldsymbol{y}(0) = \mathbf{y}_0\,, \tag{9}$$

where $\mathbf{A} \in \mathbb{R}^{ne \times ne}$ is the matrix for pure quadratic term, $\mathbf{B} \in \mathbb{R}^{ne \times ne(ne-1)/2}$ is the matrix for mixed quadratic term, $\mathbf{C} \in \mathbb{R}^{ne \times ne}$ is the Jacobian matrix for linear part of the system, $\mathbf{b} \in \mathbb{R}^{ne}$ is the right-hand side for the forces incoming to the system and $\mathbf{y}_0$ is a vector of initial conditions and symbol ne stands for the number of equations in system of ODEs. The unknown function $\boldsymbol{y}^2$ represents the vector of multiplications $(y_1y_1, y_2y_2, \ldots, y_{ne}y_{ne})^T$; the unknown function $\boldsymbol{y_{jk}}$ represents the vector of mixed terms multiplications $(y_{j_1}y_{k_1}, y_{j_2}y_{k_2} \ldots, y_{j_{ne(ne-1)/2}}y_{k_{ne(ne-1)/2}})^T$. The indexes j, k comes from combinatorics $C(ne, 2)$. For simplification we suppose that the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and the vector $\mathbf{b}$ are constant.

Higher derivatives of such systems (9) can be effectively computed in MATLAB software [15] using matrix-vector multiplication, e.g. higher derivative $\boldsymbol{y}^{[p]}$ for pure quadratic term with matrix $\mathbf{A}$ can be expressed as

$$\boldsymbol{y}^{[p]} = \mathbf{A}\left(\sum_{i=0}^{p-2} \boldsymbol{y}^{[p-1-i]}.*\boldsymbol{y}^{[i]}\binom{p-1}{i} + \boldsymbol{y}.*\boldsymbol{y}^{[p-1]}\right), \tag{10}$$

where the operation '.∗' stands for *element-by-element* multiplication, i.e. $\boldsymbol{y}^{[p_1]}.*\boldsymbol{y}^{[p_2]}$ is a vector $(y_1^{[p_1]}y_1^{[p_2]}, \ldots, y_{ne}^{[p_1]}y_{ne}^{[p_2]})^T$. The binomial coefficients $\binom{p-1}{i}$ can be effectively precomputed using Pascal triangle, for more information see *pascal* function in MATLAB software [15].

3.1 Lorenz system

Lorenz system explains some of the unpredictable behavior of the weather. The Lorenz model supposes, that a planet atmosphere consists of a two-dimensional fluid cell which is heated from below and cooled from above [9]. The fluid motion can be described by three-dimensional system of ODEs (11)

$$\begin{aligned} x' &= \sigma(y - x) \\ y' &= \rho x - y - xz \\ z' &= xy - \beta z\,, \end{aligned} \tag{11}$$

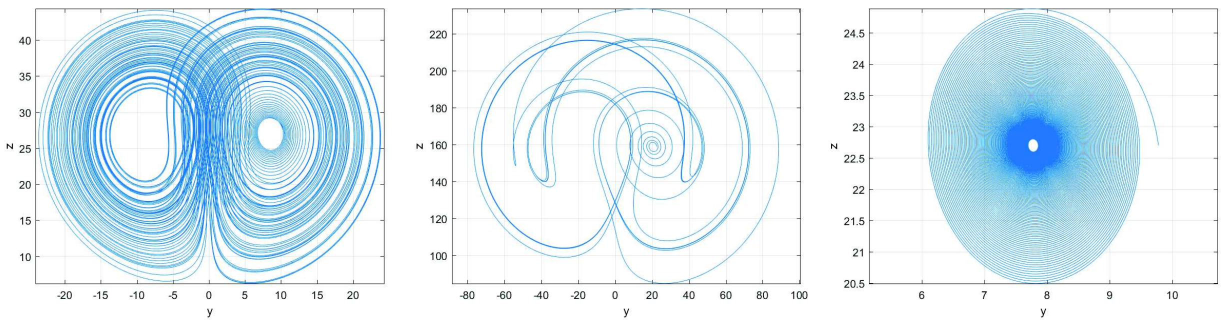
where σ is the Prandtl number, ρ is the Rayleigh number and β is the parameter related to the physical size of the system. The behavior of the system depends on the values of the parameters and initial conditions. Small changes in the initial conditions have a significant effect on the solution. The system (11) could be rewritten in the matrix form (9) where

$$\boldsymbol{y} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \quad \mathbf{A} = \mathbf{0}, \quad \mathbf{B} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} -\sigma & \sigma & 0 \\ \rho & -1 & 0 \\ 0 & 0 & -\beta \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

For the experiments, the parameters $\sigma = 10$, $\beta = 8/3$ were fixed. We change the parameter ρ to obtain different behavior of the system (11). For $\rho = 28$, the chaotic behavior could be observed (originally used by Lorenz [14]). For large ρ, e.g. $\rho = 160$, the solution is periodical (for more information see [11]). For $\rho = 23.7$, the solution is stable. Two equilibrium points can be calculated using (12). Initial conditions were then calculated by adding the constant vector $\vec{v} = (0, 2, 0)$ to the equilibrium point Q^+. For more information see [9], [8].

$$Q^{\pm} = (\pm\sqrt{\beta(\rho - 1)}, \pm\sqrt{\beta(\rho - 1)}, \rho - 1) \tag{12}$$

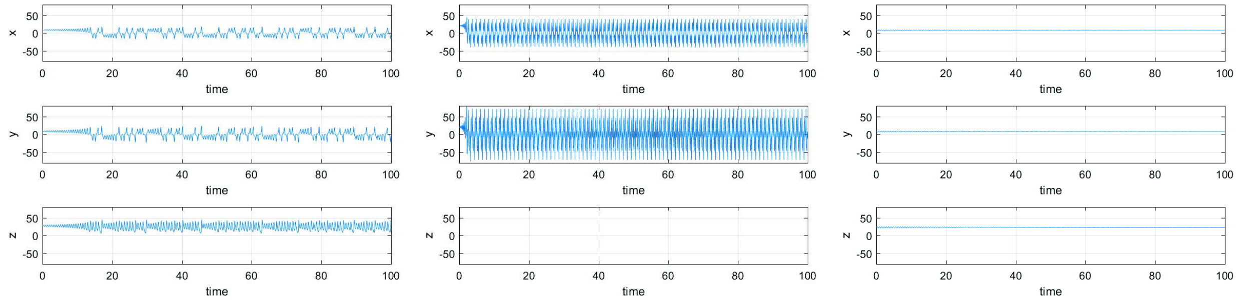
Figure 3 shows the solution of Lorenz system for different values of parameter ρ in yz-plane.



(a) $\rho = 28$ (b) $\rho = 160$ (c) $\rho = 23.7$

Figure 3: Behavior of Lorenz system, yz-plane

Solution in time domain could be seen in Figure 4. The maximum simulation time was set to $t_{MAX} = 100$ for all experiments.



(a) $\rho = 28$ (b) $\rho = 160$ (c) $\rho = 23.7$

Figure 4: Time domain

The MATLAB code of explicit Taylor series **expTay** with a variable order and variable step size scheme for nonlinear quadratic systems of ODEs (6) has been implemented. This algorithm was tested on a set of examples of Lorenz system with different ρ parameter. The MTSM was again compared with vectorized MATLAB explicit **ode** solvers. Both relative and absolute tolerances for all solvers were set to 10^{-10}. Results for comparisons MTSM with MATLAB ode solvers could be found in Table 3. Ratios of computation times $ratio =$**ode**/**expTay** > 1 indicate faster computation of the MTSM in all cases. The number of integration steps could be found in Table 4. The MTSM order (ORD) is shown in Figure 5.

Table 3: Time of solutions: explicit Taylor **expTay** and MATLAB explicit **ode** solver comparison

ρ	**ode23** *ratio*	**ode45** *ratio*	**ode113** *ratio*	**expTay** [s]
28	200.8	7.3	1.9	0.933
160	196.3	6.9	1.9	2.363
23.7	15.7	7.3	4.6	0.538

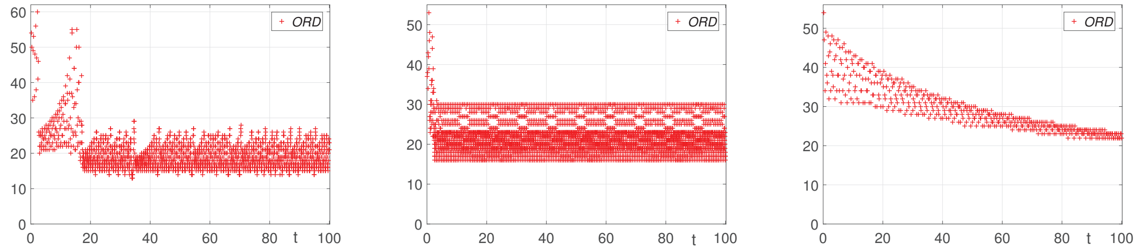
Table 4: Number of steps

ρ	**ode23**	**ode45**	**ode113**	**expTay**
28	3993135	322496	19794	2000
160	9907302	774340	48896	4000
23.7	1111529	147928	11029	500

More comparisons of MTSM numerical solutions of non-linear ODEs systems could be found in [19].

Conclusion

This article dealt with the numerical solution of linear and non-linear systems of ODEs. The model of the telegraph line was chosen as the example of linear problem, the Lorenz system



(a) $\rho = 28$ (b) $\rho = 160$ (c) $\rho = 23.7$

Figure 5: Lorenz system: ORD function

as the example of nonlinear one. All experiments were performed using MATLAB software. The MTSM solver for nonlinear systems of ODEs was successfully implemented. The experiments clearly showed, that MTSM is suitable for solving both linear and nonlinear systems. Moreover, the MTSM could be faster and more accurate than state-of-the art ode solvers in MATLAB.

Future work will be focused to the parallelization and hardware representation of the MTSM.

Acknowledgements

This research was financially supported by the 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-17-4014.

References

[1] R. Barrio. Performance of the Taylor series method for ODEs/DAEs. In *Applied Mathematics and Computation*, volume 163, pages 525–545, 2005. ISSN 00963003.

[2] R. Barrio, F. Blesa, and M. Lara. VSVO Formulation of the Taylor Method for the Numerical Solution of ODEs. In *Computers and Mathematics with Applications*, volume 50, pages 93–111, 2005.

[3] R. Barrio, M. Rodríguez, A. Abad, and F. Blesa. TIDES: A free software based on the Taylor series method. *Monografías de la Real Academia de Ciencias de Zaragoza*, 35:83–95, 2011.

[4] M. Berz. COSY INFINITY version 8 reference manual. *Technical Report MSUCL–1088, National Superconducting Cyclotron Lab., Michigan State University, East Lansing, Mich.*, 1997.

[5] Jan Chaloupka, Gabriela Nečasová, Petr Veigend, Jiří Kunovský, and Václav Šátek. Modern taylor series method in numerical integration: Part 1. In *16th Czech-Polish Conference Modern Mathematical Methods in Engineering (3mi)*, volume 6, pages 263–273, 2017.

[6] Y. F. Chang and G. Corliss. Atomf: solving odesand daes using Taylor series. *Computers Math. Applic.*, 28:209–233, 1994.

[7] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I.* vol. Nonstiff Problems. Springer-Verlag Berlin Heidelberg, 1987. ISBN 3-540-56670-8.

[8] James Hateley. The lorenz system. `http://web.math.ucsb.edu/~jhateley/paper/lorenz.pdf`.

[9] Morris W Hirsch, Stephen Smale, and Robert L Devaney. *Differential equations, dynamical systems, and an introduction to chaos.* Academic press, 2004. ISBN 0-12-349703-5.

[10] A. Jorba and M. Zou. A software package for the numerical integration of ODE by means of high-order Taylor methods. In *Exp. Math.*, volume 14, pages 99–117, 2005.

[11] O. Knill. The lorenz system. `http://www.math.harvard.edu/archive/118r_spring_05/handouts/lorentz.pdf`.

[12] Filip Kocina, Václav Šátek, Petr Veigend, Gabriela Nečasová, Václav Valenta, and Jiří Kunovský. New Trends in Taylor Series Based Applications. In *13rd International Conference of Numerical Analysis and Applied Mathematics*, number 1648, pages 1–4. American Institute of Physics, 2015.

[13] J. Kunovský. *Modern Taylor Series Method.* FEI-VUT Brno, 1994. Habilitation work.

[14] Edward N Lorenz. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2):130–141, 1963.

[15] The MathWorks. *MATLAB and Simulink software*, 2017.

[16] Gabriela Nečasová, Petr Veigend, Václav Šátek, and Jiří Kunovský. Model of the Telegraph Line. In *Proceedings of the 14th International Conference Informatics' 2017*, 2017, accepted.

[17] Nedialko S. Nedialkov and J. Pryce. Solving differential algebraic equations by Taylor series III. the DAETS code. *JNAIAM J. Numer. Anal. Ind. Appl. Math.*, 3:61–80, 2008.

[18] Václav Šátek, Filip Kocina, Jiří Kunovský, and Alexander Schirrer. Taylor Series Based Solution of Linear ODE Systems and MATLAB Solvers Comparison. In *MATHMOD VIENNA 2015 - 8th Vienna Conference on Mathematical Modelling*, ARGESIM REPORT No. 44, pages 693–694. ARGE Simulation News, 2015.

[19] Václav Šátek, Petr Veigend, and Gabriela Nečasová. Taylor series based solution of nonlinear-quadratic ode systems. In *MATHMOD VIENNA 2018 - 9th Vienna Conference on Mathematical Modelling*, 2017, submitted.

[20] TKSL software. *High Performance Computing*, [cit. 21-3-2017]. URL: http://www.fit.vutbr.cz/ kunovsky/TKSL/index.html.en [online].

[21] Petr Veigend, Jiří Kunovský, Filip Kocina, Gabriela Nečasová, Václav Šátek, and Václav Valenta. Electronic Representation of Wave Equation. In *13rd International Conference of Numerical Analysis and Applied Mathematics*, number 1738, pages 1–4. American Institute of Physics, 2015.

Moderní metoda Taylorovy řady v numerické integraci: část II.

Abstrakt: Článek se zabývá přesným, rychlým a stabilním řešením obyčejných diferenciálních rovnic (Cauchyho úlohy). Soustavy těchto rovnic jsou řešeny pomocí Moderní metody Taylorovy řady. Tato metoda je proměnného řádu a využívá proměnný integrační krok. Členy Taylorovy řady se počítají iterativně, díky tomu je možno vypočítat i vyšší derivace.
Článek prezentuje řešení lineárních a nelineárních problémů. Jako lineární problém bylo zvoleno řešení telegrafní rovnice, jako nelineární byl zvolen Lorenzův systém. Experimenty byly provedeny pomocí systému MATLAB s využitím nově implementovaných nástrojů. Moderní metoda Taylorovy řady byla porovnána s běžně používanými řešiči obyčejných diferenciálních rovnic v systému MATLAB.

Klíčová slova: Metoda Taylorovy řady, obyčejné diferenciální rovnice, technické problémy, Cauchyho úloha.