Efficient Solution of Nonlinear DAE Problems using Modern Taylor Series Method

Petr Veigend, Václav Šátek, Gabriela Nečasová
Faculty of Information Technology, Brno University of Technology,
Email: {vegend, satek, inecasova}@fit.vut.cz

BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

Modern Taylor Series Method

- **Exact**, **stable** and **fast** high-order numerical method for solving ordinary differential equations (ODEs): **initial value problems** (**IVPs**)
- **Automatic integration order setting**, i.e. using as many Taylor series terms as the defined accuracy requires
- **Recurrent calculation** of the Taylor series terms rather than derivation of functions for each time step
- **Automatic transformation** of the original technical problem to autonomous system of ODEs

Nonlinear Systems of ODEs

- $\mathbf{y}' = \mathbf{A}\mathbf{y} + \mathbf{B_1}\mathbf{y}_{jk} + \mathbf{B_2}\mathbf{y}_{jkl} + \ldots + \mathbf{b}, \quad \mathbf{y(0)} = \mathbf{y0},$
  - $\mathbf{A} \in \mathrm{R}^{ne \times ne}$ is the constant matrix for the linear part
  - $\mathbf{B_1} \in \mathrm{R}^{ne \times nm_{jk}}, \mathbf{B_2} \in \mathrm{R}^{ne \times nm_{jkl}}$ are constant matrices for the nonlinear part with operation multiplication of 2, 3... terms
  - $\mathbf{b} \in \mathrm{R}^{ne}$ is the right-hand side for the forces incoming to the system
  - $\mathbf{y0}$ is the vector of the initial conditions
  - ne is the number of equations of the system of ODEs
  - nm_{jk}, nm_{jkl} represent number of two and three-function multiplications
  - Unknown function $\mathbf{y}_{jk} \in \mathrm{R}^{nm_{jk}}$ represents the vector of two-term multiplication $\mathbf{y}_j \odot \mathbf{y}_k$, and similarly for three- and more term multiplication
  - Higher derivatives can be calculated recurrently:

$$
\begin{aligned}
&\mathbf{p}_A(1) = \mathbf{A}\mathbf{y} + \mathbf{b}\,, \quad \mathbf{p}_{B1}(1) = \mathbf{B}_1\mathbf{y}_{jk}\,,\\
&\mathbf{p}_{B2}(1) = \mathbf{B_2}\mathbf{y}_{jkl}\,, \qquad \mathbf{p}_A(r) = \mathbf{A}\mathbf{p}(r-1)\,,\\
&\mathbf{p}_{B1}(r) = \mathbf{B}_1 \sum_{a=1}^{r} \mathbf{p}_j(a-1) \odot \mathbf{p}_k(r-a)\,,\\
&\mathbf{p}_{B2}(r) = \mathbf{B}_2 \sum_{a=0}^{r-1} \mathbf{p}_{jj}(a) \odot \left(\sum_{b=1}^{r-a} \mathbf{p}_{kk}(b-1) \odot \mathbf{p}_{ll}(r-a-b) \right)
\end{aligned}
$$

  - Where $r = 2, \ldots, n$. Taylor series terms are caluculated as a sum of linear and nonlinear terms:

$$\mathbf{p}(s) = \frac{h}{s}(\mathbf{p}_A(s) + \mathbf{p}_{B1}(s) + \mathbf{p}_{B2}(s)), \quad s = 1, \ldots, n$$

  - Where r, s are current indeces of Taylor series terms, a, b are auxiliary indeces for the summation of two and three term multiplication in nonlinear part, $p_A(s)$ is the linear term of the system.
  - Next value of the function at time $t = t_i$ can be calculated as:

$$\mathbf{y}_{i+1} = \mathbf{p}(0)_i + \mathbf{p}(1)_i + \mathbf{p}(2)_i + \ldots + \mathbf{p}(n)_i$$

Nonlinear Implementation (DAE Approach)

- To handle the *division* operation efficiently, the problem is represented as an implicit system of DAEs. This allows the MTSM solver to maintain a high integration order without a massive increase in computational complexity.

Nonlinear ODE → DAE → MTSM solver

Kepler Problem

- A classic celestial mechanics problem describing the motion of two bodies under mutual gravity. It is a benchmark for numerical stability.
- Governing equations (nonlinear ODEs)

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

- Where e is an eccentricity of a rotating body and $\mathrm{r} = \sqrt{y_1^2 + y_2^2}$

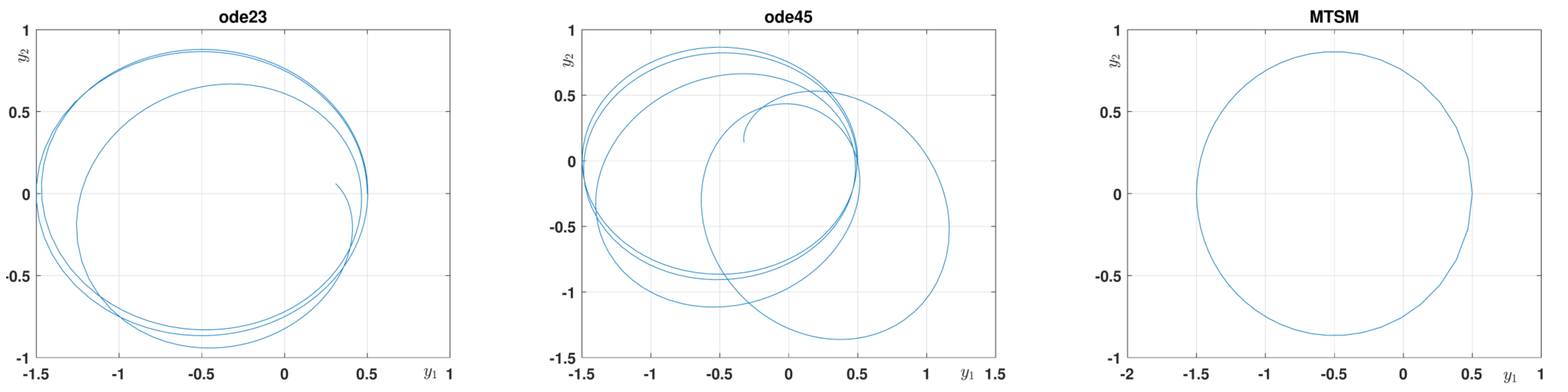

Figure 1: Numerical solution for $e = 0.5$ (stability of solution).

Numerical Results

Parameter	Value
MTSM accuracy	$\varepsilon = 1 \times 10^{-10}$
Relative, absolute tolerances for MATLAB solvers	$rel_{TOL} = abs_{TOL} = 1 \times 10^{-13}$
Number of rotations	$n_r = 2$
Maximum simulation time	$T = 2\pi n_r$ [s]
Eccentricity	$e_1 = 0.25,\ e_2 = 0.5,\ e_3 = 0.75$
Step size for MTSM solvers	$h_1 = \frac{2\pi}{20}$ [s], $h_2 = \frac{2\pi}{50}$ [s], $h_3 = \frac{2\pi}{100}$ [s]
Maximum order for MTSM solvers	$P_{max} = 64$

Table 1: Parameters for numerical experiments.

Solver	Method
ode23	Bogacki-Shampine
ode45	Dormand-Prince
$MTSM_M$	Nonlinear MTSM solver
$MTSM_D$	Nonlinear MTSM solver for DAE systems

Table 2: MATLAB numerical methods.

e	**ode23**	**ode45**	$MTSM_M$	$MTSM_D$
0.25	155054	9601	40	40
0.5	216172	13901	100	100
0.75	313721	21217	200	200

Table 3: Number of steps.

	ode23	**ode45**	$MTSM_M$	$MTSM_D$
e	s_a	s_a	s_a	[s]
0.25	**161.16**	**4.09**	**1.16**	0.0061
0.5	**97.83**	**2.18**	**1.11**	0.013
0.75	**85.57**	**1.81**	**1.08**	0.023

s_a- speedupof DAE approach against the ode solvers

Table 4: Comparisons between the proposed DAEs formulation ($MTSM_D$), previously used formulation ($MTSM_M$) and MATLAB solvers.

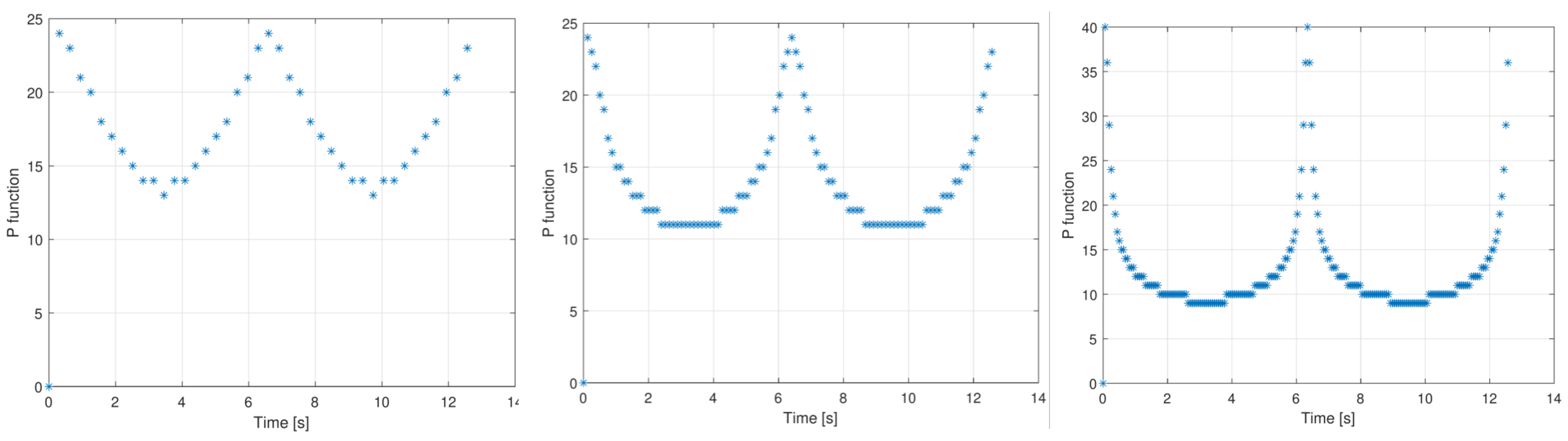

Figure 2: Plot of P function for $e = 0.25, 0.5, 0.75$.

ODEs

$$
\begin{aligned}
y_1' &= y_3 & y_1(0) &= 1 - e\\
y_2' &= y_4 & y_2(0) &= 0\\
y_3' &= -y_{13} & y_3(0) &= 0\\
y_4' &= -y_{14} & y_4(0) &= \sqrt{\frac{1+e}{1-e}}\\
y_5' &= 3y_{10} & y_5(0) &= r(0)^3\\
y_6' &= y_{11} & y_6(0) &= r(0)\\
y_7' &= -3y_{24} & y_7(0) &= \frac{1}{y_5(0)}\\
y_8' &= -y_{25} & y_8(0) &= \frac{1}{y_6(0)}\\
y_9' &= y_{17} + y_{18} - y_{19} & y_9(0) &= y_1(0)y_3(0) + y_2(0)y_4(0)\\
y_{10}' &= y_{20} + y_{26} + y_{27} - y_{28} & y_{10}(0) &= y_6(0)y_9(0)\\
y_{11}' &= -y_{29} + y_{30} + y_{31} - y_{32} & y_{11}(0) &= y_8(0)y_9(0)\\
y_{12}' &= 2y_{22} + 2y_{23} & y_{12}(0) &= y_1(0)^2 + y_2(0)^2
\end{aligned}
$$

DAEs

$$
\begin{aligned}
y_{13} &= y_1y_7 & y_{14} &= y_2y_7\\
y_{15} &= y_7y_7 & y_{16} &= y_8y_8\\
y_{17} &= y_3y_3 & y_{18} &= y_4y_4\\
y_{19} &= y_7y_{12} & y_{20} &= y_9y_{11}\\
y_{21} &= y_{11}y_{11} & y_{22} &= y_1y_3\\
y_{23} &= y_2y_4 & y_{24} &= y_{10}y_{15}\\
y_{25} &= y_{11}y_{16} & y_{26} &= y_6y_{17}\\
y_{27} &= y_6y_{18} & y_{28} &= y_6y_{19}\\
y_{29} &= y_8y_{21} & y_{30} &= y_8y_{17}\\
y_{31} &= y_8y_{18} & y_{32} &= y_8y_{19}\,,
\end{aligned}
$$

VSB TUO IT4I HPCSE 2026 High Performance Computing in Science and Engineering