Journal of Mathematical Sciences, Vol. 139, No. 6, 2006

TAYLOR SERIES METHOD FOR DYNAMICAL SYSTEMS WITH CONTROL: CONVERGENCE AND ERROR ESTIMATES

L. K. Babadzhanjanz and D. R. Sarkissian UDC 517.987.5

Abstract. To optimize a complicated function constructed from a solution of a system of ordinary differential equations (ODEs), it is very important to be able to approximate a solution of a system of ODEs very precisely. The precision delivered by the standard Runge–Kutta methods often is insufficient, resulting in a "noisy function" to optimize.

We consider an initial-value problem for a system of ordinary differential equations having polynomial right-hand sides with respect to all dependent variables. First we show how to reduce a wide class of ODEs to such polynomial systems. Using the estimates for the Taylor series method, we construct a new "aggregative" Taylor series method and derive guaranteed a priori step-size and error estimates for Runge–Kutta methods of order r. Then we compare the 8,13-Prince–Dormand's, Taylor series, and aggregative Taylor series methods using seven benchmark systems of equations, including van der Pol's equations, the "brusselator," equations of Jacobi's elliptic functions, and linear and nonlinear stiff systems of equations. The numerical experiments show that the Taylor series method achieves the best precision, while the aggregative Taylor series method achieves the best computational time.

The final section of this paper is devoted to a comparative study of the above numerical integration methods for systems of ODEs describing the optimal flight of a spacecraft from the Earth to the Moon.

CONTENTS

1. Introduction

In Secs. 2 and 4, we consider error estimates for the Runge–Kutta and Taylor series and aggregative Taylor series methods for polynomial systems of ODEs. In Secs. 3 and 6, we compare the performance of the numerical integration methods with respect to benchmark examples and the equations of optimal flight of a spacecraft from the Earth to the Moon. Some issues of the numerical implementation are discussed in Sec. 5.

The idea of reduction to polynomial systems goes back to [10]. The general method for such reduction was proposed in [1] and refined in [2] and [6, Sec. 2.1]. In particular, the main differential equations of celestial mechanics can be reduced to a form with right-hand sides that are just quadratic polynomials (see Sec. 6).

For the sake of completeness, we outline the reduction method below.

Translated from Sovremennaya Matematika i Ee Prilozheniya (Contemporary Mathematics and Its Applications), Vol. 24, Dynamical Systems and Optimization, 2005.

1072–3374/06/1396–7025

Consider the Cauchy problem

$$\frac{dx_i}{dt} = f_i(x_1, \dots, x_m, t), \tag{1}$$

$$x_i(t_0) = x_{i0}, \quad i = 1, \dots, m. \tag{2}$$

From now on, we suppose that (1) belongs to the class of differential systems for which

$$f_i(x_1, \dots, x_m, t) = P_i(x_1, \dots, x_m; t; \phi_1(x_1, \dots, x_m, t), \dots, \phi_k(x_1, \dots, x_m, t)),$$

where for all $1 \le r \le k$ and $1 \le s \le m$,

$$\frac{\partial \phi_r}{\partial t} = \Phi_r(x_1, \dots, x_m, t, \phi_1, \dots, \phi_k), \qquad \frac{\partial \phi_r}{\partial x_s} = \Phi_{r,s}(x_1, \dots, x_m, t, \phi_1, \dots, \phi_k)$$

and all the functions $P_1, \dots, P_m$, $\Phi_1, \dots, \Phi_k$, and $\Phi_{1,1}, \dots, \Phi_{k,m}$ are polynomials with respect to t, $x_1, \dots, x_m$, and $\phi_1, \dots, \phi_k$.

Then, introducing the variables $x_{m+1} = t$, $x_{m+2} = \phi_1$, ..., $x_n = \phi_k$, we reduce the original Cauchy problem for the system of ODEs (1) to the following polynomial Cauchy problem for $x_1, \dots, x_n$:

$$\frac{dx_j}{dt} = X_j(x_1, \dots, x_n), \tag{3}$$

$$x_j(t_0) = x_{j0}, \quad j = 1, \dots, n, \tag{4}$$

where all the functions

$$X_1 = P_1, \dots, X_m = P_m, X_{m+1} = 1,$$
$$X_{m+2} = \sum_{r=1}^{m} \Phi_{1,r} X_r + \Phi_1, \dots, X_n = \sum_{r=1}^{m} \Phi_{k,r} X_r + \Phi_k$$

are polynomials with respect to $x_1, \dots, x_n$.

Remark 1. If $\mathcal{P}$ is the set of all such right-hand sides f_i, then the set $\mathcal{S}$ of scalar functions with scalar arguments that can be expressed by a system of ODEs with right-hand sides from $\mathcal{P}$ includes practically all analytic functions from the known handbooks of mathematical functions. We also note that the set $\mathcal{P}$ is closed with respect to algebraic operations, differentiation, integration, and composition with any function from $\mathcal{S}$. It is very important to realize that the class of polynomial systems of ODEs is a huge one.

Remark 2. Piecewise analytic functions can be considered without difficulty, which is the case for the example in Sec. 6.

For example, given the "modal" Cauchy problem

$$\dot{x}_1 = x_1 \sin \ln(x_2) + {x_1}^2 \ln(x_2), \quad \dot{x}_2 = x_2 \ln(x_1),$$
$$x_1(t_0) = e, \quad x_2(t_0) = 1,$$

we introduce the new variables

$$x_3 = \ln(x_2), \quad x_4 = \ln(x_1), \quad x_5 = \sin \ln(x_2), \quad x_6 = \cos \ln(x_2), \quad x_7 = x_1 x_3$$

and reduce it to the polynomial Cauchy problem

$$\dot{x}_1 = x_1 x_7 + x_1 x_5, \quad \dot{x}_2 = x_2 x_4, \quad \dot{x}_3 = x_4, \quad \dot{x}_4 = x_7 + x_5,$$
$$\dot{x}_5 = x_4 x_6, \quad \dot{x}_6 = -x_4 x_5, \quad \dot{x}_7 = x_7^2 + x_5 x_7 + x_1 x_4,$$
$$x_1(t_0) = e, \quad x_2(t_0) = 1, \quad x_3(t_0) = 0, \quad x_4(t_0) = 1, x_5(t_0) = 0, \quad x_6(t_0) = 1, \quad x_7(t_0) = 0.$$

1.1. Benchmark examples.

We will use the following systems of equations (see [7, 8]) to compare the numerical integration methods in this paper.

The first four benchmark systems already have the desired polynomial form:

SIMPLEST: the Cauchy problem for $x(t) = 1/(1-t)$ is

$$\dot{x}(t) = x(t)^2, \quad x(0) = 1.$$

STIFF–LINEAR: the Cauchy problem for $y(t) = s(t) + e^{-100t}y(0)$, $s(t) = \sin(t)$, $c(t) = \cos(t)$ is

$$\dot{y}(t) = -100y(t) + c(t) + 100s(t), \quad \dot{s}(t) = c(t), \quad \dot{c}(t) = -s(t),$$
$$y(0) = 2, \quad s(0) = 0, \quad c(0) = 1.$$

STIFF–CAPS: the Cauchy problem for $y_1(t) = y_2^2(t)$, $y_2(t) = e^{-t}$ with parameter $\alpha = 1.25 \cdot 10^4$ is

$$\dot{y}_1(t) = -(\alpha+2)y_1(t) + \alpha y_2(t)^2, \quad \dot{y}_2(t) = y_1(t) - y_2(t) - y_2(t)^2,$$
$$y_1(0) = 1, \quad y_2(0) = 1.$$

JACOB: the Cauchy problem for Jacobi's elliptic functions is

$$\frac{d}{dt}\operatorname{sn}(t) = \operatorname{cn}(t)\operatorname{dn}(t), \quad \frac{d}{dt}\operatorname{cn}(t) = -\operatorname{sn}(t)\operatorname{dn}(t), \quad \frac{d}{dt}\operatorname{dn}(t) = -0.5\operatorname{sn}(t)\operatorname{cn}(t),$$
$$\operatorname{sn}(0) = 0, \quad \operatorname{cn}(0) = 1, \quad \operatorname{dn}(0) = 1.$$

The solution has the period $4K$, where $K = 1.8540746773013719184$.

The other two systems are reduced to the quadratic polynomial form as follows.

VDPL: the Cauchy problem for van der Pol's equation is

$$\dot{y}_1(t) = y_2(t), \quad \dot{y}_2(t) = (1 - y_1(t)^2)y_2(t) - y_1(t),$$
$$y_1(0) = 2.0086198608748431365, \quad y_2(0) = 0.$$

Introducing the additional variable $y_3(t) = 1 - y_1^2(t)$, we obtain the system

$$\dot{y}_1(t) = y_2(t), \quad \dot{y}_2(t) = y_2(t)y_3(t) - y_1(t), \quad \dot{y}_3(t) = -2y_1(t)y_2(t),$$
$$y_1(0) = 2.0086198608748431365, \quad y_2(0) = 0, \quad y_3(0) = 1 - y_1^2(0).$$

The solution has period $T = 6.6632868593231301897$.

BRUS: the Cauchy problem for the "brusselator" equations is

$$\dot{y}_1(t) = 2 + y_1(t)^2 y_2(t) - 9.533y_1(t), \quad \dot{y}_2(t) = 8.533y_1(t) - y_1(t)^2 y_2(t),$$
$$y_1(0) = 1, \quad y_2(0) = 4.2665.$$

Introducing the additional variables $y_3(t) = y_1(t)y_2(t) - 8.533$, $y_4(t) = y_1(t)^2$, and $y_5(t) = 1$, we obtain the system

$$\begin{aligned}
&\dot{y}_1(t) = 2y_5(t) + y_1(t)y_3(t) - y_1(t), \quad \dot{y}_2(t) = -y_1(t)y_3(t), \\
&\dot{y}_3(t) = -8.533y_5(t) + 2y_2(t) + 7.533y_3(t) + y_3(t)^2 - y_3(t)y_4(t), \\
&\dot{y}_4(t) = 4y_1(t) - 2y_4(t) + 2y_3(t)y_4(t), \quad \dot{y}_5(t) = 0, \\
&y_1(0) = 1, \quad y_2(0) = 4.2665, \quad y_3(0) = -4.2665, \quad y_4(0) = 1, \quad y_5(0) = 1.
\end{aligned}$$

2. Local Error Estimates for Taylor Series and Runge–Kutta Methods

Consider the polynomial Cauchy problem (3),(4) with

$$X_j = \sum_{m=1}^{L+1} \sum_{i \in I(m)} a_j[i] x^i, \tag{5}$$

$$i = (i_1, \dots, i_n) \in \mathbb{Z}^n, \quad x = (x_1, \dots, x_n), \quad x^i = x_1^{i_1} \cdots x_n^{i_n},$$
$$I(m) = \{i \in \mathbb{Z}^n : i_1 \geq 0, \dots, i_n \geq 0; |i| = m\}, \quad |i| = i_1 + \dots + i_n,$$
$$1 \leq j \leq n, \quad 0 \leq L < +\infty, \quad x_{j0} \in \mathbb{C}, \quad a_j[i] \in \mathbb{C}.$$

We denote by $x(t, t_0, x_0)$ the solution of this problem, where $x_0 = (x_{10}, \dots, x_{n0})$, and let $O_a(b) = \{t \in \mathbb{C} : |t - b| < a\}$.

2.1. Taylor series method. We denote the partial sum of Taylor series of the Cauchy problem as follows:

$$T_M x(t, t_0, x_0) = \sum_{m=0}^{M} \left.\frac{d^m x}{dt^m}\right|_{t=t_0} \frac{(t - t_0)^m}{m!}. \tag{6}$$

Let $R(t_0, x_0)$ be the radius of convergence for the Taylor series $T_\infty x(t, t_0, x_0)$.

The Taylor series method for solving the Cauchy problem (1), (2) approximates the true solution $x(t)$ of the Cauchy problem as

$$\begin{aligned} \tilde{x}(t_1) &= T_{M_1} x(t_1, t_0, x_0), \\ &\dots\dots\dots\dots\dots\dots \\ \tilde{x}(t_w) &= T_{M_w} x(t_w, t_{w-1}, \tilde{x}(t_{w-1})), \\ &\dots\dots\dots\dots\dots\dots \end{aligned} \tag{7}$$

where $M_1, M_2, \dots$ are some natural numbers, $t_1 = t_0 + h_1, t_2 = t_1 + h_2, \dots$, and the local steps $h_1, h_2, \dots$ satisfy the inequalities $|h_w| < R(t_{w-1}, \tilde{x}(t_{w-1}))$.

The natural number M_w is the order of the method for the step number w. To obtain $T_M x(t, \tau, \tilde{x}(\tau))$, we should calculate the first M derivatives $(d^l x/dt^l)|_{t=\tau}$, $l = 1, \dots, M$, under the assumption that $(d^0 x/dt^0)|_{t=\tau} = \tilde{x}(\tau)$. It is the prohibitive computational cost of calculating these derivatives that prevents the use of the Taylor series method for the general Cauchy problem. But in the case of polynomial Cauchy problems fast recursive formulas for the derivatives $(d^l x/dt^l)|_{t=\tau}$ are obtained as follows, allowing the Taylor series method to compete with traditional Runge–Kutta methods with respect to both computational time and precision of approximation.

2.1.1. Formulas for the calculation of Taylor coefficients.

The general polynomial case. Consider the Cauchy problem (3), (4) with polynomial right-hand sides (5) and initial conditions $x_k(t_0) = c_k$, $k = 1, \dots, n$, where t_0 and c_k are complex constants, L is a natural number ($+\infty$ in the case of an analytical system), and $a_k[i]$ are some complex holomorphic functions in t in a neighborhood of t_0.

To emphasize that the right-hand sides X_k depend also on $a_k[i]$, we use the notation $X_k(\{a_k[i]\}, x_1, \dots, x_n)$ along with $X_k(x_1, \dots, x_n)$. Let

$$x_k^r = \sum_{j=0}^{r} x_{k,j} (t - t_0)^j, \qquad a_k^r[i] = \sum_{j=0}^{r} a_{k,j}[i] (t - t_0)^j$$

be the Taylor polynomials for $x_k(t)$, $a_k[i]$. Denoting the coefficient of the rth-degree term in the polynomial $X_k(\{a_k^r[i]\}, x_1^r, \dots, x_n^r)$ by $\langle X_k(\{a_k^r[i]\}, x_1^r, \dots, x_n^r)\rangle_r$, we note that to evaluate it, we need to retain in every computation the terms up to rth degree only.

Substituting the Taylor expansions into system (3), (4), (5) and collecting similar terms, we obtain the following formulas:

$$x_{k,r+1} = \frac{1}{r+1}\langle X_k(\{a_k^r[i]\}, x_1^r, \dots, x_n^r)\rangle_r,$$
$$x_{k,0} = c_k, \quad k = 1, \dots, n, \quad r = 0, 1, \dots .$$

The case of a quadratic polynomial. Consider the quadratic Cauchy problem

$$\dot{x}_k(t) = a_k + \sum_{l=1}^{n} a_{kl}\, x_l(t) + \sum_{i,j=1}^{n} a_{kij}\, x_i(t)\, x_j(t),$$
$$x_k(t_0) = c_k,$$

where t_0, c_k, a_k, a_{kl}, and a_{kjl} are complex constants and t is a complex variable. Substituting the Taylor series expansion

$$x_k(t) = \sum_{m=0}^{\infty} x_{k,m}(t - t_0)^m$$

into the equations and collecting similar terms, we obtain the formulas

$$x_{k,m+1} = \frac{1}{m+1}\left(\delta_m a_k + \sum_{l=1}^{n} a_{kl}\, x_{l,m} + \sum_{i,j=1}^{n} a_{kij} \sum_{p=0}^{m} x_{i,p}\, x_{j,m-p}\right),$$
$$x_{k,0} = c_k, \quad k = 1, \dots, n, \quad m = 0, 1, \dots,$$

where $\delta_0 = 1$, $\delta_1 = \delta_2 = \dots = 0$.

2.1.2. The step size and error estimates. To estimate the steps h_i and the local error of the method, we use the following theorem.

Theorem 3. *Let $a_j[i]$ be complex constants and let $L > 0$. If the inequality $|x_{j0}| \le \gamma$ holds for every $1 \le j \le n$, then*

(a) *the solution $x = (x_1, \dots, x_n)$ of problem (3), (4), (5) is regular in a circle $O_\rho(t_0)$, where*

$$\rho = \frac{1}{Ls(\gamma)}, s(\gamma) = \max_{1\le j\le n} \sum_{m=1}^{L+1} \gamma^{m-1} \sum_{i\in I(m)} |a_j[i]|; \tag{8}$$

(b) *if $t \in O_\rho(t_0)$, then the following inequalities hold for all $1 \le j \le n$:*

$$|x_j(t) - (T_M x(t, t_0, x_0))_j| \le \gamma\left(1 - \frac{|t - t_0|}{\rho}\right)^{-1/L} \left|\frac{t - t_0}{\rho}\right|^{M+1}. \tag{9}$$

Proof. See [3, Proposition 2]. □

Remark 4. If $L = 0$ in system (3), (4), (5), then we introduce a “fake” equation $\dot{x}_{n+1} = \varepsilon x_{n+1}^2$, $\varepsilon \approx 0$, with initial condition $x_{n+1}(t_0) = 0$ and apply Theorem 3 with $L = 1$.

Remark 5. To improve the estimates of Theorem 3, we may rewrite the system with respect to new variables $y_j(t) = x_j(t)/\sigma_j(t_0)$, $1 \le j \le n$, with appropriate $\sigma_1 > 0, \dots, \sigma_n > 0$, since the true convergence radius and true relative errors for the new variables y_j remain the same as for the old variables x_j.

Remark 6. Generalizations of Theorem 3 for systems of ordinary differential equations (5) with holomorphic $a_j[i]$ in the cases $L < +\infty$ and $L = +\infty$ are given in [4] and [5].

2.1.3. Taylor series method estimates for benchmark examples. In this section, we apply Theorem 3 to write the convergence-radius estimates for the benchmark examples from Sec. 1.1.

SIMPLEST: $\gamma = |x(t_0)|$, $s(\gamma) = \gamma$, $L = 1$, $\rho = 1/(Ls(\gamma))$.

Note that the convergence-radius estimate ρ gives exactly the value of the true convergence radius and the estimate of local error (9) gives exactly the value of the true local error.

In the remaining examples, we use appropriate σ_i from Remark 5 to improve the estimates of $s(\gamma)$.

STIFF-LINEAR:

$$\sigma_1 = 1, \quad \sigma_2 = \sigma_3 = 0.01, \quad s(\gamma) = 101.01,$$
$$\gamma = \max\left\{|y(t_0)|,\ |100s(t_0)|,\ |100c(t_0)|\right\}.$$

STIFF-CAPS:

$$\sigma_1 = \alpha, \quad \sigma_2 = 1, \quad s(\gamma) = 2 + \alpha + \gamma,$$
$$\gamma = \max\left\{\frac{|y_1(t_0)|}{\alpha},\ |y_2(t_0)|\right\}.$$

JACOB:

$$\sigma_1 = 1, \quad \sigma_2 = 1, \quad \sigma_3 = \sqrt{0.5}, \quad s(\gamma) = \sqrt{0.5}\gamma,$$
$$\gamma = \max\left\{|\operatorname{sn}(t_0)|,\ |\operatorname{cn}(t_0)|,\ \sqrt{2}|\operatorname{dn}(t_0)|\right\}.$$

VDPL:

$$\sigma_1 = 0.5, \quad \sigma_2 = 1, \quad \sigma_3 = \sqrt{2}, \quad s(\gamma) = \max\left\{2,\ \sqrt{2}\gamma + 0.5\right\},$$
$$\gamma = \max\left\{|2y_1(t_0)|,\ |y_2(t_0)|,\ \left|\frac{y_3(t_0)}{\sqrt{2}}\right|\right\}.$$

BRUS:

$$\sigma_1 = \sigma_2 = 1, \quad \sigma_3 = \max\left\{1,\ \frac{|y_3(t_0)|}{\gamma}\right\},$$
$$\sigma_4 = \max\left\{1,\ \frac{|y_4(t_0)|}{\gamma}\right\}, \quad \sigma_5 = \frac{1}{\gamma},$$
$$\gamma = \max\left\{|y_1(t_0)|,\ |y_2(t_0)|\right\},$$
$$s(\gamma) = 2 + \sigma_3\gamma + \max\left\{\left|5.533 + \frac{2}{\sigma_3} + \sigma_4\gamma + \frac{8.533}{\gamma\sigma_3}\right|,\ \left|\frac{4}{\sigma_4} + \sigma_3\gamma\right|\right\}.$$

2.2. Runge–Kutta methods. The main problem with heuristic step estimates for Runge–Kutta methods is that they can overstep the solution singularity, causing catastrophic loss of integration precision. Based on Theorem 3, we derive theoretical a priori step size and error estimates for general explicit Runge–Kutta methods as follows.

We denote the r, q-Runge–Kutta method (of degree r with q stages) by

$$\begin{aligned} x_j^{(1)} &= x_{j0}, \\ k_j^{(1)} &= X_j(x_1^{(1)}, \ldots, x_n^{(1)}), \\ &\ldots\ldots\ldots \\ x_j^{(i)} &= x_{j0} + h\sum_{l=1}^{i-1} \beta_{il} k_j^{(l)}, \\ k_j^{(i)} &= X_j(x_1^{(i)}, \ldots, x_n^{(i)}), \\ &\ldots\ldots\ldots \end{aligned} \tag{10}$$

$$x_j(t) \approx R_j x(t, t_0, x_0) = x_{j0} + h\sum_{i=1}^{q} p_i k_j^{(i)} \tag{11}$$

with real constants p_i, β_{il} and $h = t - t_0 > 0$ such that $1 \le j \le n$, $2 \le i \le q$.

In the case of quadratic systems of ODEs with real constants t_0, c_k, a_k, a_{kl}, a_{kjl} and real t, in order to use the estimates (see Theorem 7 for details), the following quantities should be computed:

$$\begin{aligned} s_1 &= \max\left\{s(\gamma), s_2, \gamma s_3\right\}, \\ s_2 &= \max_{1\le j\le n} \sum_{v=1}^{n} \left| a_{jv} + \sum_{m=1}^{n} (a_{jvm} + a_{jmv}) x_j^{(1)} \right|, \\ s_3 &= \max_{1\le j\le n} \sum_{v,m=1}^{n} |a_{jvm} + a_{jmv}|. \end{aligned} \tag{12}$$

2.2.1. Step size and error estimates for Runge–Kutta methods.

Theorem 7. *Let $R\,x(t, t_0, x_0)$ be the r, q-Runge–Kutta approximation defined by* (11). *Then under the conditions of Theorem* 3 *with $L = 1$ for all $h < 1/(\tau s_1)$ the following inequality holds*:

$$|x_j(t_0 + h) - R\,x(t_0 + h, t_0, x_0)_j| \le \gamma(h s_1 \tau)^{r+1} \left(\frac{D}{\tau} + \left(\frac{s(\gamma)}{s_1\tau}\right)^{r+1} \left(1 - \frac{s(\gamma)}{s_1\tau}\right)^{-1} \right), \tag{13}$$

where constants τ and D depend only on β_{il} and p_i (see Sec. 2.2.2 *and* (21)).

Proof. See Sec. 4.1. □

2.2.2. Parameters for the Runge–Kutta methods. In this section, we write down the constants r, q, τ, and D required in Theorem 7. The original parameters of the Runge–Kutta methods were taken from [8] and [11].

4,4-Runge–Kutta method:

$$\beta_{21} = \beta_{32} = \frac{1}{2}, \quad \beta_{43} = 1, \quad \beta_{31} = \beta_{41} = \beta_{42} = 0,$$
$$p_1 = p_4 = \frac{1}{6}, \quad p_2 = p_3 = \frac{1}{3},$$
$$r = 4, \quad q = 4, \quad \tau = 2, \quad D < 0.577.$$

8,13-Prince–Dormand method:

$$r = 8, \quad q = 13, \quad \tau = 33.77, \quad D < 3.58.$$

2.2.3. Runge–Kutta method estimates for benchmark examples.

SIMPLEST: $s_1 = 2\gamma$.
STIFF-LINEAR: $s_1 = 101.01$.
STIFF-CAPS: $s_1 = \alpha + 2 + 2\gamma$.
JACOB: $s_1 = \max\left\{\sqrt{2}\gamma,\ \sqrt{0.5}|\operatorname{cn}(t_0)| + |\operatorname{dn}(t_0)|,\ \sqrt{0.5}|\operatorname{sn}(t_0)| + |\operatorname{dn}(t_0)|\right\}$.
VDPL: $s_1 = \max\left\{s(\gamma),\ 2\sqrt{2}\gamma,\ 0.5 + \sqrt{2}|y_2(t_0)| + |y_3(t_0)|\ \right\}$.
BRUS:

$$\sigma_3 = \max\left\{1,\ \frac{|y_3(t_0)|}{\gamma}\right\}, \quad \sigma_4 = \max\left\{1,\ \frac{|y_4(t_0)|}{\gamma}\right\},$$

$$s_1 = \max\left\{s(\gamma),\ 2\gamma(\sigma_3+\sigma_4),\ |y_3(t_0)-1| + \sigma_3|y_1(t_0)| + \frac{2}{\gamma}, 4\sigma_3\gamma,\ |7.533 + 2y_3(t_0) - y_4(t_0)| \right.$$
$$\left. + \frac{\sigma_4}{\sigma_3}|y_3(t_0)| + \frac{2}{\sigma_3} + \frac{8.533}{\gamma\sigma_3}, \sigma_3|y_1(t_0)| + |y_3(t_0)|,\ \frac{4}{\sigma_4} + 2\frac{\sigma_3}{\sigma_4}|y_4(t_0)| + |2y_3(t_0) - 2|\right\}.$$

2.3. Aggregative Taylor series method. Since we want to approximate the true solution $x(t)$ very precisely, the local error estimate (9) assumes that the local step size $\tau_{i+1} - \tau_i$ must be a tiny fraction of the convergence-radius estimate ρ_i at each instant of time τ_i and it will take many steps to step out of the convergence radius ρ_0 around the point τ_0. That is, using operators $T_M x(\tau_{i+1}, \tau_i, x_0)$ as defined in (6), we have

$$h = \frac{\tau_r - \tau_0}{r}, \quad \tau_i = \tau_0 + ih, \quad |t - t_0| = |\tau_r - \tau_0| < \rho_0$$

for some large number r of steps of the Taylor series method.

Thus, we define one step of the aggregative Taylor series method as the result of r steps of the same size $(t - t_0)/r$ of the Taylor series method as follows:

$$ATx(r, N; t, t_0, x_0) = ATx(r, N; \tau_r, \tau_0, x_0) = \tilde{x}_r, \tag{14}$$

where

$$\begin{aligned} &\tilde{x}_1 = T_N x(\tau_1, \tau_0, x_0), \\ &\cdots\cdots\cdots\cdots \\ &\tilde{x}_r = T_N x(\tau_r, \tau_{r-1}, \tilde{x}_{r-1}). \end{aligned} \tag{15}$$

$ATx(r, N; t, t_0, x_0)$ is precisely the result of calculation of the solution $x(t)$ by r steps of the Taylor series method. Our goal is to collect similar terms and obtain an explicit formula for $\tilde{x}_r$ using the first few derivatives $(d^l x(t_0)/dt^l), l = 1, \ldots, Nr$. We hope that the aggregative Taylor series method will reduce both the compounded round-off errors and the computational time in practical calculations and will be applicable to both stiff and nonstiff initial-value problems for systems of ODEs. The numerical experiments we have carried out in Sec. 3 seem to justify our hopes.

Theorem 8. *The formula for $ATx(r, N; t, t_0, x_0)$ can be obtained in the following form*:

$$ATx(r, N; t, t_0, x_0) = \sum_{m=0}^{Nr} \alpha(m, r, N) \frac{(t - t_0)^m}{m!} \frac{d^m x(t_0)}{dt^m}, \tag{16}$$

where

$$\alpha(m, r, N) = r^{-m} \sum_{\substack{k_1 + \ldots + k_r = m, \\ 0 \le k_1, \ldots, k_r \le N}} \frac{m!}{k_1! \cdots k_r!}.$$

Proof. See Sec. 4.2. □

In actual computations we truncate each step of the aggregative Taylor series method as

$$AT_M x(r,N;t,t_0,x_0) = \sum_{m=0}^{M} \alpha(m,r,N) \frac{(t-t_0)^m}{m!} \frac{d^m x(t_0)}{dt^m} \tag{17}$$

and construct a table of approximative values $\tilde{x}(t_1)$, …, $\tilde{x}(t_w)$ as follows:

$$\begin{aligned} \tilde{x}(t_1) &= AT_{M_1} x(r_1, N_1; t_1, t_0, x_0), \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ \tilde{x}(t_w) &= AT_{M_w} x(r_w, N_w; t_w, t_{w-1}, x_{w-1}), \\ &\ldots\ldots\ldots\ldots\ldots\ldots \end{aligned} \tag{18}$$

where $t_1 = t_0 + r_1 h_1$, $t_2 = t_1 + r_2 h_2, \ldots$ are the time moments and the natural numbers $1 \le r_w, N_w, M_w \le r_w N_w$ are parameters of the aggregative Taylor series method at the wth step (a detailed algorithm for calculating these parameters is given in Sec. 5.4).

Remark. Operating at local tolerances near the computer epsilon, the collection of similar terms, offered by the aggregate Taylor series method improves the accuracy of the computations by compounding fewer round-off errors. Also, for small tolerances one step of the aggregative Taylor series method is a computational shortcut to obtain the result of many steps of the Taylor series method cheaply, since the estimates of Sec. 5.4 will often let us get away with small M_w.

3. Numerical Experiments for Benchmark Examples

Tables detailing the comparative study of various numerical methods using numerical integration of the benchmark problems described in Sec. 1.1 are given at the end of this paper.

For tolerances up to 10^{-15} the numerical methods use double-precision arithmetic with a computer epsilon of $2.22 \cdot 10^{-16}$. For more strict tolerances the numerical methods use software emulating quadruple precision with a computer epsilon of $1.93 \cdot 10^{-34}$. The global error is computed exactly using the exact solutions or periodicity properties of the benchmark examples, except for the brusselator example, where the result of the Taylor series method of degree 30 is assumed to play the role of "exact solution."

SAN SPARC workstation was used. All the methods for these examples are implemented in FORTRAN 77, and the codes are available upon request. For details of the practical realization of these methods, see Sec. 5.

4. Proofs

4.1. Proof of Theorem 7. Introducing the notation

$$\begin{aligned} k_j^{i,w} &= \frac{d^w}{dh^w} k_j^{(i)}(h), \\ f_i(w) &= \sum_{u=1}^{i-1} |\beta_{iu}| g_u(w-1), \quad i \ge 2, \\ g_i(w) &= f_i(w) + \frac{1}{2} \sum_{u=1}^{w-1} f_i(u) f_i(w-u), \\ f_i(0) &= g_i(0) = 1, \quad 1 \le i \le q, \\ f_1(w) &= g_1(w) = 0, \quad w \ge 1 \end{aligned} \tag{19}$$

and representing the Runge–Kutta approximation formula as

$$R_j\, x = \sum_{u=0}^{r} \frac{x^{(u)}(t_0) h^u}{u!} + \sum_{u=r+1}^{2^q-1} \alpha_{ju} h^u,$$

using (9) with $M = r$ and $h < 1/(\tau s_1)$, we obtain

$$|x_j - R_j\, x| \le \gamma \left(1 - \frac{1}{\rho s_1 \tau}\right)^{-1} \left(\frac{h}{\rho}\right)^{r+1} + \sum_{u=r+1}^{2^q-1} |\alpha_{ju}| h^u, \tag{20}$$

where

$$\alpha_{ju} = \frac{1}{(u-1)!} \sum_{i=1}^{q} p_i k_j^{i,u-1}(0).$$

Using the estimates of Lemma 9 below, we deduce

$$\max_j \sum_{u=r+1}^{2^q-1} |\alpha_{ju}| h^u \le \frac{\gamma}{\tau} \sum_{u=r+1}^{2^q-1} (hs_1\tau)^u \sum_{i=1}^{q} |p_i| \frac{g_i(u-1)}{\tau^{u-1}}$$
$$\le \frac{\gamma}{\tau} (hs_1\tau)^{r+1} \sum_{u=r}^{2^q-2} (hs_1\tau)^{u-r} \sum_{i=1}^{q} |p_i| \frac{g_i(u)}{\tau^u} \le (hs_1\tau)^{r+1} \frac{\gamma D}{\tau},$$

where

$$hs_1\tau \le 1 \quad \text{and} \quad D = \sum_{u=r}^{2^q-2} \sum_{i=1}^{q} |p_i| \frac{g_i(u)}{\tau^u}. \tag{21}$$

To complete the proof, it remains to take (20) into account and compute the constant D in terms of β_{ij}, p_i by (19). To reduce the sharply increasing $g_i(u)$, one may choose a suitable value of the parameter $\tau \in (1, +\infty)$.

Lemma 9. *Under the conditions of Theorem 7, for any $w \ge 0$ and $1 \le j \le n$ the inequality*

$$|k_j^{i,w}(0)| \le w! s_1^{w+1} \gamma g_i(w)$$

holds.

Proof. Using the notation

$$x_j^{i,w} = \frac{d^w}{dh^w} x_j^{(i)}(h), \quad \partial_v X_{ji} = \frac{\partial X_j}{\partial x_v}(x_1^{(i)}, \ldots, x_n^{(i)}),$$
$$\partial_v \partial_m X_{ji} = \partial_v (\partial_m X_{ji}) = a_{jmv} + a_{jvm}$$

and differentiating (10), we obtain

$$\begin{aligned} x_j^{i,w} &= w \sum_{u=1}^{i-1} \beta_{iu} k_j^{u,w-1} + h \sum_{u=1}^{i-1} \beta_{iu} k_j^{u,w}, \\ k_j^{i,w} &= \sum_{v=1}^{n} \partial_v X_{ji} x_v^{i,w} + \sum_{v,m=1}^{n} \partial_v \partial_m X_{ji} \sum_{k=1}^{[w/2]} C_{w,k} x_v^{i,k} x_m^{i,w-k} \end{aligned} \tag{22}$$

with natural constants $C_{w,k}$ for $w \ge 1$ (see Lemma 11 below).

Equations (10) also imply that

$$|k_j^{i,0}(0)| \le \gamma s(\gamma), \quad |x_j^{i,0}(0)| \le \gamma, \quad k_j^{1,w} = x_j^{1,w} = 0 \quad \text{for} \quad w \ge 1. \tag{23}$$

Using (22) and (23), we obtain the following estimates by induction on $w = 0, 1, \ldots$ for $1 \le i \le q$, $1 \le j \le n$:

$$|k_j^{i,w}(0)| \le k^{i,w} = w! s_1^{w+1} \gamma g_i(w), \quad |x_j^{i,w}(0)| \le x^{i,w} = w! s_1^w \gamma f_i(w) = x^{i,w}. \tag{24}$$

In fact, for $w = 0$ the inequalities follow from (19) and (23). Note that for $i = 1$ the inequalities are also obvious implications of (19) and (23).

If inequalities (24) hold for $0,\dots,w-1$, then taking Lemma 10 and (22) into consideration for $2\le i\le q$, we obtain

$$|x_j^{i,w}(0)|\le w\sum_{u=1}^{i-1}|\beta_{iu}|(w-1)!s_1^w\gamma g_u(w-1)=x^{i,w},$$

$$k_j^{i,w}(0)|\le\sum_{v=1}^{n}|\partial_v X_{ji}|\,\big|_{h=0}\;x^{i,w}+\frac{1}{2}\sum_{v,m=1}^{n}|\partial_v\partial_m X_{ji}|\sum_{k=1}^{w-1}\binom{w}{k}x^{i,k}x^{i,w-k}$$

$$\le s_1 w!s_1^w\gamma f_i(w)+\frac{s_1}{2}\sum_{k=1}^{w-1}\binom{w}{k}k!s_1^k f_i(k)(w-k)!s_1^{w-k}\gamma f_i(w-k)=w!s_1^{w+1}\gamma g_i(w),$$

because of the inequality

$$s_2\ge\max_j\sum_{v=1}^{n}|\partial X_{ji}|\,\big|_{h=0}=\max_j\sum_{v=1}^{n}\left|a_{iv}+\sum_{m=1}^{n}(a_{jvm}+a_{jmv})\left((x_m^{(1)}+h\sum_{u=1}^{i-1}\beta_{iu}k_j^{(u)}\right)\right|\,\Big|_{h=0}.$$

The proof is complete. □

Lemma 10. *The following equality holds for $w\ge 2$, $1\le v,m\le n$:*

$$\sum_{k=1}^{[w/2]}C_{w,k}x_v^{i,k}x_m^{i,w-k}=\frac{1}{2}\sum_{k=1}^{w-1}\binom{w}{k}x_v^{i,k}x_m^{i,w-k}.$$

Proof. Use Lemma 11. □

Lemma 11. *For $k\le[w/2]$ and $w\ge 2$,*

$$C_{w,k}=\begin{cases}\dfrac{1}{2}\dbinom{w}{\frac{w}{2}}, & w=2k,\\[2ex] \dbinom{w}{k}, & \textit{otherwise.}\end{cases}\tag{25}$$

Proof. Using (22), by induction on w one verifies relation (25).

In fact, for $w=1,2,3$ the formula is obvious.

Let (25) hold for w. We introduce E_{k_1,k_2} and $\{k_1,k_2\}$ as follows:

$$E_{k_1,k_2}=E_{k_2,k_1}=\frac{1}{2}C_{w,k_1}\quad\text{if}\quad 1\le k_1<w,\;\;1\le k_2<w,\;\;k_1\ne k_2,\;\;w=k_1+k_2,$$
$$E_{k,k}=C_{w,k}\qquad\text{if}\;\;2k=w,$$
$$E_{k_1,k_2}=0\qquad\text{otherwise,}$$

$$\{k_1,k_2\}=\{k_2,k_1\}=\sum_{v,m=1}^{n}\partial_v\partial_m X_{ji}x_v^{i,k_1}x_m^{i,k_2}.$$

Then, using (22), we have

$$k_j^{i,w+1}=\sum_{v=1}^{n}\partial_v X_{ji}x_v^{i,w+1}+\{1,w\}+\sum_{k_1+k_2=w}E_{k_1,k_2}(\{k_1+1,k_2\}+\{k_1,k_2+1\})$$

$$=\sum_{v=1}^{n}\partial_v X_{ji}x_v^{i,w+1}+\{1,w\}+\sum_{k_1+k_2=w+1}E_{k_1-1,k_2}\{k_1,k_2\}+\sum_{k_1+k_2=w+1}E_{k_1,k_2-1}\{k_1,k_2\}$$

$$= \sum_{v=1}^{n} \partial_v X_{ji} x_v^{i,w+1} + \{1, w\} + \left(\sum_{\substack{k_1<k_2,\\ k_1+k_2=w+1}} + \sum_{\substack{k_2<k_1,\\ k_1+k_2=w+1}} + \sum_{\substack{k_1=k_2,\\ k_1+k_2=w+1}} \right)(E_{k_1-1,k_2} + E_{k_1,k_2-1})\{k_1, k_2\}$$

$$= \sum_{v=1}^{n} \partial_v X_{ji} x_v^{i,w+1} + \{1, w\} + \sum_{\substack{k_1<k_2,\\ k_1+k_2=w+1}} (E_{k_1-1,k_2} + E_{k_1,k_2-1} + E_{k_2-1,k_1} + E_{k_2,k_1-1})\{k_1, k_2\}$$

$$+ \sum_{\substack{k_1=k_2,\\ k_1+k_2=w+1}} \left(E_{\frac{w+1}{2}-1,\frac{w+1}{2}} + E_{\frac{w+1}{2},\frac{w+1}{2}-1}\right)\left\{\frac{w+1}{2}, \frac{w+1}{2}\right\}.$$

Collecting similar terms, one can see the following:

$$\begin{aligned}
C_{w+1,1} &= 1 + E_{0,w} + E_{1,w-1} + E_{w-1,1} + E_{w,0} = 1 + C_{w,1},\\
C_{w+1,k} &= E_{k-1,w+1-k} + E_{k,w-k} + E_{w-k,k} + E_{w+1-k,k-1}\\
&= C_{w,k-1} + C_{w,k} \quad \text{if} \quad 2 \le k < \frac{w}{2},\\
C_{w+1,\frac{w}{2}} &= E_{\frac{w}{2}-1,\frac{w}{2}+1} + E_{\frac{w}{2},\frac{w}{2}} + E_{\frac{w}{2},\frac{w}{2}} + E_{\frac{w}{2}+1,\frac{w}{2}-1}\\
&= C_{w,\frac{w}{2}-1} + 2C_{w,\frac{w}{2}} \quad \text{if} \quad w \text{ is even},\\
C_{w+1,\frac{w+1}{2}} &= C_{w,\frac{w-1}{2}} \qquad\qquad \text{if} \quad w \text{ is odd}.
\end{aligned}$$

The proof is complete. □

4.2. Proof of Theorem 8.

Remark 12. We have deduced (16) using the infinite-system method, proposed by one of the authors (see [1, 2]). Our attempts to find a simple proof of this result by induction failed. So we begin with the infinite-system method.

Introduction to the infinite-system method. Omitting details, we outline this method as follows.

Step 1. Problem (3), (4), (5) is reduced to an infinite linear system of first-order ODEs.

Step 2. Taking advantage of the linearity of the infinite system, we obtain the desired results for the linear problem derived.

Step 3. We interpret the results of step 2 in terms of the original problem (3), (4), (5).

Now we prove Theorem 8 in accordance with the above scheme.

For every $i \in \bigcup_{m=1}^{+\infty} I(m)$, we introduce the new variable

$$y[i] = x^i = x_1^{i_1}\, x_2^{i_2} \ldots x_n^{i_n}.$$

To order the set of variables we just introduced, we arrange the sets $\chi_m = \{y[i] | i \in I(m)\}$ from left to right in order $\chi_1, \chi_2, \ldots$ and, for every m, we arrange the elements of χ_m in lexicographic order, i.e., $x[i_1, \ldots, i_n]$ precedes $x[k_1, \ldots, k_n]$ if $i_1 = k_1$, …, $i_j = k_j$, $i_{j+1} > k_{j+1}$. Then we enumerate all the elements of the ordered set $\bigcup_{m=1}^{+\infty} \chi_m$ as $y_1, y_2, \ldots$.

Let us write (3), (4), (5) in the matrix form as

$$\frac{dy}{dt} = \mathcal{A}y, \quad y(t_0) = y_0, \tag{26}$$

where

$$y = (y_1, y_2, \ldots), \quad \frac{dy}{dt} = \left(\frac{dy_1}{dt}, \frac{dy_2}{dt}, \ldots\right),$$

$$y_0 = (x_{10}, \ldots, x_{n0},\, x_{10}^2,\, x_{10} \cdot x_{20}, \ldots, x_{n0}^2, \ldots).$$

Here $\mathcal{A}$ is an infinite band-type matrix, i.e., a matrix for which every row and every column has only a finite number of nonzero elements. If the coefficients $a_k[i]$ of the original problem (3), (4), (5) are constants (real or complex, it does not matter), then $\mathcal{A}$ is also a constant matrix.

Remark 13. A similar idea for algebraic systems goes back to K. Weierstrass. For ODEs, R. Bellman refers to Carleman. The general formulas for the coefficients of the matrix $\mathcal{A}$ have been proposed in [1].

Differentiating Eq. (26), we obtain

$$\frac{d\,y}{dt} = \mathcal{A}y, \qquad \frac{d^2y}{dt^2} = \mathcal{A}\frac{dy}{dt} = \mathcal{A}\mathcal{A}y = \mathcal{A}^2 y, \; \ldots, \; \frac{d^r y}{dt^r} = \mathcal{A}^r y.$$

Using these formulas and denoting the first n rows of the matrix $\mathcal{A}$ by $[\mathcal{A}]$, we rewrite (6) in the form

$$T_N x(t, t_0, x_0) = \sum_{m=0}^{N} \frac{(t-t_0)^m}{m!} \, [\mathcal{A}^m] \, x_0,$$

which implies the following explicit formulas for (15):

$$\tilde{x}_1 = \sum_{k_1=0}^{N} \frac{h^{k_1}}{k_1!} \left[\mathcal{A}^{k_1}\right] x_0,$$

$$\tilde{x}_2 = \sum_{k_2=0}^{N} \frac{h^{k_2}}{k_2!} \left[\mathcal{A}^{k_2}\right] \left(\sum_{k_1=0}^{N} \frac{h^{k_1}}{k_1!} \left[\mathcal{A}^{k_1}\right] x_0 \right) = \sum_{k_2=0}^{N} \sum_{k_1=0}^{N} \frac{h^{k_1+k_2}}{k_1! k_2!} \left[\mathcal{A}^{k_1+k_2}\right] x_0,$$

$$\cdots\cdots\cdots$$

$$\tilde{x}_r = \sum_{k_r=0}^{N} \cdots \sum_{k_1=0}^{N} \frac{h^{k_1+\cdots+k_2}}{k_1! \cdots k_r!} \left[\mathcal{A}^{k_1+\cdots+k_r}\right] x_0.$$

To complete the proof, we take into account Eq. (14) and $t - t_0 = rh$.

5. Practical Implementation of the Methods

In this section, we consider some necessary details of the implementation of the numerical methods.

5.1. Practical error estimate for the Dormand–Prince method.

The code for this method was taken from [8]. The coefficients of the method are correct up to 18 decimal digits.

5.2. Guaranteed relative error estimate for the r, q-Runge–Kutta method.

The positive constants $\tau > 1$ and D (see Sec. 2.2.2 and (21)) should be computed only once for each Runge–Kutta method.

If t_0 and x_0 are known from the previous step or as initial data, then, according to formula (13), the inequality

$$|x_j(t_0+h, t_0, x_0) - R\,x(t_0+h, t_0, x_0)_j| \le \varepsilon\gamma$$

is guaranteed if the integration step is calculated as follows (see also (12)):

$$h = \frac{1}{s_1\tau} \min\left\{ 1, \sqrt[r+1]{\varepsilon \left(\frac{D}{\tau} + \left(1 - \frac{1}{\rho s_1 \tau}\right)^{-1} (\rho s_1 \tau)^{-r-1} \right)^{-1}} \right\}.$$

5.3. Guaranteed local error estimate for the Taylor series method. Let M denote the order of the Taylor series method (see (6)). If t_0 and x_0 are known from the previous step or as initial data, then the integration step is calculated as follows (see (9)):

$$h = \rho \min\left\{\frac{1}{2}, \sqrt[M+1]{\frac{\varepsilon}{2}}\right\}, \tag{27}$$

which, according to formula (9), guarantees the inequality

$$|x_j(t_0+h, t_0, x_0) - T_M x(t_0+h, t_0, x_0)_j| \leq \varepsilon\gamma.$$

5.4. Practical error estimate for the aggregative Taylor series method. The aggregative Taylor series method is based on formulas (16) and (17). In these formulas, we set

$$|t - t_0| \leq \lambda\rho, \quad t = t_0 + rh, \tag{28}$$

where ρ is the estimate of the convergence radius and $0 < \lambda < 1$ is fixed. In numerical experiments we put $\lambda = 0.5$ (see Sec. 3).

In the absence of round-off errors, we obtain $AT_{Nr}x = ATx = \tilde{x}_r$ (see (15)–(17)), i.e., in this case $ATx(r, N; t, t_0, x_0)$ is precisely the result of calculating the solution $x(t)$ by r identical steps of the Nth order Taylor series method. We denote by ε the relative local admissible error of each step of the aggregative Taylor series method.

To obtain a more accurate approximation by one step of the aggregative Taylor series method compared with r consequent steps of the Taylor series method we have to choose the parameters M, r, and λ so that

$$|ATx(r, N; t, t_0, x_0) - AT_M x(r, N; t, t_0, x_0)| = \left| \sum_{m=M+1}^{Nr} \alpha(m, r, N) \frac{r^m h^m}{m!} \frac{d^m x}{dt^m}\bigg|_{t=t_0} \right|$$

is sufficiently small. On the other hand, the greater the M, the more laborious the computation of $AT_M x$. That is why we restrict $M \leq M_{\max}$ and $r \leq R_{\max}$, where $M_{\max}$ and $R_{\max}$ are some fixed parameters.

Using the heuristic (which always holds if ε is sufficiently small) estimate

$$|ATx(r, N; t, t_0, x_0) - AT_M x(r, N; t, t_0, x_0)| \leq \mathrm{Err}(M, h), \tag{29}$$

where

$$\mathrm{Err}(M, h) = (Nr - M) \left| \alpha(M, r, N) \frac{r^M h^M}{M!} \frac{d^M x}{dt^M}\bigg|_{t=t_0} \right|,$$

we propose the following *rule for selecting the parameters* M and r.

1. Parameters ε, N, λ, $M_{\max}$, and $R_{\max}$ are taken by the user initially.
2. The parameter r is calculated as (see (9), (28))

$$h = \rho \cdot \min\left\{\lambda, \sqrt[N+1]{\varepsilon(1-\lambda)}\right\}, \qquad r = \min\left\{\left[\frac{\lambda\rho}{h}\right], R_{\max}\right\},$$

   where $[a]$ denotes the largest integer such that $[a] \leq a$.
3. The parameter M is taken as the least natural number such that $M \leq M_{\max}$ and $\mathrm{Err}(M, h) \leq \varepsilon\gamma$. If such M exists, then we compute

$$x(t_0 + rh) \approx AT_M x(r, N; t_0 + rh, t_0, x(t_0)).$$

4. If such M does not exist, we set

$$x(t_0 + rh_1) \approx AT_M x(r, N; t_0 + rh_1, t_0, x(t_0)),$$

   where $M = M_{\max}$ and $h_1 = h \sqrt[M]{(\varepsilon\gamma)/\mathrm{Err}(M, h)}$ (since $\mathrm{Err}(M, h_1) = \varepsilon\gamma$).

6. The Optimal Flight from the Earth to the Moon

In this section, we apply the estimates obtained in this paper to the real-life system of ODEs describing the optimal flight of a spacecraft from the Earth to the Moon (see [9]).

We denote by x, y, u, v, and m the coordinates, velocities, and mass of the spacecraft, respectively, and by p_x, p_y, p_u, p_v, and p_m we denote the corresponding canonically conjugate variables.

These equations involve the following physical constants:

$$x_\Lambda = 384.4, \quad x_B = 4.665507, \quad \mu_\Lambda = 2.81556 \cdot 10^{-6}, \quad \mu_3 = 6.256818 \cdot 10^{-5},$$

$$P_{\max} = 9.81 \cdot 10^{-7}, \quad c = 3.4335 \cdot 10^{-3}, \quad \omega = 2.6676 \cdot 10^{-6}.$$

Following [9], we use the initial conditions

$$\begin{aligned}
u(0) &= -4.406463701184517 \cdot 10^{-3}, & x(0) &= -5.418331088513003,\\
v(0) &= -6.401245846380632 \cdot 10^{-3}, & y(0) &= 3.729848819356288,\\
p_u(0) &= -206534.5274675425, & p_x(0) &= -675.626010487872,\\
p_v(0) &= -978439.312867057, & p_y(0) &= 1016.800527213885,\\
p_m(0) &= 3071.41103, & m(0) &= 1.
\end{aligned}$$

6.1. The system of ODEs. First, we introduce the following notation:

$$g_{11}^1 = x, \quad g_{21}^1 = x - x_\Lambda, \quad g_{i2}^1 = y, \quad g_{i1}^2 = u, \quad g_{i2}^2 = v, \quad i = 1, 2,$$

$$p_1^1 = p_x, \quad p_2^1 = p_y, \quad p_1^2 = u, \quad p_2^2 = p_v,$$

$$d_i = \sqrt{(g_{i1}^1)^2 + (g_{i2}^1)^2}, \quad \rho = \sqrt{(p_1^2)^2 + (p_2^2)^2}, \quad d_{i0} = d_i(0), \quad \rho_0 = \rho(0).$$

This notation enables us to rewrite the system of ODEs for the spacecraft flight motion in symmetric and compact form (30).

To obtain the polynomial Cauchy problem in the form (3), (4), (5) with quadratic right-hand sides (i.e., $L = 1$), we introduce the following variables:

$$\begin{aligned}
&u_i = \frac{d_{i0}}{d_i}, \quad v_i = (u_i)^2, \quad B_i = (u_i)^3, \quad K = \frac{\rho_0}{\rho},\\
&x_{ij} = \frac{g_{ij}^1 u_i}{d_i}, \quad w_{ij} = x_{ij} u_i, \quad C_{ij} = w_{ij} u_i, \quad D_{ijk} = x_{ij} w_{ik},\\
&y_j = \frac{g_{1j}^2}{\alpha}, \quad z_{ij} = y_j u_i, \quad \theta_i = \frac{1}{\beta} \sum_{j=1}^{2} x_{ij} y_j, \quad m_1 = \frac{\xi}{m},\\
&E_j^1 = \frac{p_j^1}{\eta\rho}, \quad E_j^2 = \frac{p_j^2}{\rho}, \quad M_j^i = m_1 E_j^i, \quad F_{kl}^{ij} = E_k^i E_l^j,\\
&E_3^i = E_1^i, \quad M_3^i = E_1^i, \quad F_{k3}^{ij} = F_{k1}^{ij}, \quad A_1 = \frac{1}{\alpha}\left(\frac{\omega}{2}(g_{11}^1 - x_B) + g_{12}^2\right),\\
&A_2 = \frac{1}{\alpha}\left(\frac{\omega}{2} g_{12}^1 - g_{11}^2\right), \quad E = \lambda \frac{p_m}{\rho}.
\end{aligned}$$

Remark 14. It is easily seen that the denotations for y, u, and v are not unique. For example, u is denoted both by g_{11}^2 and by g_{21}^2. Similarly, $E_1^i = E_3^i$, $M_1^i = M_3^i$, $F_{k1}^{ij} = F_{k3}^{ij}$, and $q_1 = q_3$.

In Sec. 6.2, we choose real positive parameters α, η, ξ, and λ to improve the relative local error and step-size estimates and to guarantee that $\gamma < 1$.

Differentiating the variables u_i, ..., E and using the original equations from [9], we obtain the following system of ODEs $(i, j, l = 1, 2)$:

$$\dot{u}_i = -\beta\sigma_i u_i\theta_i, \quad \dot{v}_i = -2\beta\sigma_i v_i\theta_i, \quad \dot{B}_i = -3\beta\sigma_i B_i\theta_i, \quad \dot{K} = \phi K,$$
$$\dot{x}_{ij} = \sigma_i u_i z_{ij} - 2\beta\sigma_i x_{ij}\theta_i, \quad \dot{w}_{ij} = \sigma_i b_i z_{ij} - 3\beta\sigma_i w_{ij}\theta_i,$$
$$\dot{C}_{ij} = \sigma_i B_i z_{ij} - 4\beta\sigma_i C_{ij}\theta_i, \quad \dot{D}_{ijl} = \sigma_i C_{il} z_{ij} + \sigma_i C_{ij} z_{il} - 5\beta\sigma_i D_{ijl}\theta_i,$$
$$\dot{y}_j = q_j, \quad \dot{z}_{ij} = u_i q_j - \beta\sigma_i z_{ij}\theta_i,$$
$$\dot{\theta}_i = -2\beta\sigma_i\theta_i^2 + \frac{1}{\beta}\sum_{j=1}^{2}(\sigma_i(z_{ij})^2 + x_{ij}q_j), \quad \dot{m}_1 = \frac{P}{\xi c}(m_1)^2,$$
$$\dot{E}_j^1 = \phi E_j^1 + \frac{1}{\eta}\left(E_j^2\sum_{k=1}^{2}\mu_k(B_k - 3D_{kjj}) - E_j^2\omega^2 - 3E_{1+j}^2\sum_{k=1}^{2}\mu_k D_{k12}\right),$$
$$\dot{E}_j^2 = -\eta E_j^1 + 2\omega\chi_j E_{1+j}^2 + \phi E_j^2,$$

$$\begin{aligned}
\dot{M}_j^1 &= \frac{1}{\eta}\left(M_j^2\sum_{k=1}^{2}\mu_k(B_k - 3D_{kjj}) - M_j^2\omega^2 - 3M_{1+j}^2\sum_{k=1}^{2}\mu_k D_{k12}\right) + +\phi M_j^1 + \frac{P}{\xi c}m_1 M_j^1,\\
\dot{M}_j^2 &= -\eta M_j^1 + 2\omega\chi_j M_{1+j}^2 + \phi M_j^2 + \frac{P}{\xi c}m_1 M_j^2,\\
\dot{F}_{jl}^{11} &= \frac{1}{\eta}\left(F_{lj}^{12}\sum_{k=1}^{2}\mu_k(B_k - 3D_{kjj}) - F_{lj}^{12}\omega^2 - 3F_{l\,1+j}^{12}\sum_{k=1}^{2}\mu_k D_{k12}\right.\\
&\left. + F_{jl}^{12}\sum_{k=1}^{2}\mu_k(B_k - 3D_{kll}) - F_{jl}^{12}\omega^2 - 3F_{j\,1+l}^{12}\sum_{k=1}^{2}\mu_k D_{k12}\right) + 2\phi F_{jl}^{11},\\
\dot{F}_{jl}^{12} &= \frac{1}{\eta}\left(F_{jl}^{22}\sum_{k=1}^{2}\mu_k(B_k - 3D_{kjj}) - F_{jl}^{22}\omega^2 - 3F_{l\,1+j}^{22}\sum_{k=1}^{2}\mu_k D_{k12}\right)\\
&+2\phi F_{jl}^{12} - \eta F_{jl}^{11} + 2\omega\chi_l F_{j\,1+l}^{12},\\
\dot{F}_{jl}^{22} &= -\eta F_{jl}^{12} + 2\omega\chi_j F_{l\,1+j}^{2,2} - \eta F_{lj}^{12} + 2\omega\chi_l F_{j\,1+l}^{22} + 2\phi F_{jl}^{22},\\
\dot{A}_j &= \frac{\omega}{2}y_{1j} + \chi_j q_{1+j}, \quad \dot{E} = \lambda\frac{P}{\xi^2}(m_1)^2 + \phi E,
\end{aligned} \tag{30}$$

where

$$\chi_1 = +1, \quad \chi_2 = -1, \quad \beta = \sqrt{\frac{3}{2}}, \quad \sigma_i = \frac{\alpha}{d_{i0}}, \quad \phi = \eta F_{11}^{21} + \eta F_{22}^{21},$$
$$q_j = \frac{1}{\alpha}\left(\frac{P}{\xi}M_j^2 - \sum_{k=1}^{2}\frac{\mu_k}{(d_{k0})^2}w_{kj}\right) + 2\omega A_j, \quad q_3 = q_1,$$

and the optimal control condition is

$$P = \begin{cases} P_{\max}, & \lambda c m_1 > \xi E\\ 0, & \lambda c m_1 < \xi E. \end{cases}$$

To complete the initial-value problem, we also compute initial values for our new variables u_{i0}, v_{i0}, B_{i0}, K_0, x_{ij0}, y_{j0}, z_{ij0}, w_{ij0}, C_{ij0}, D_{ijk0}, θ_{i0}, A_{i0}, E^i_{j0}, M^i_{j0}, F^{ij}_{kl0}, m_{10}, and E_0 via the old variables $x(0)$, $y(0)$, $\ldots$, $p_m(0)$.

We estimate the global error by verifying precisely how numerical integration met-hods are able to conserve the constant of motion

$$H = \frac{\rho_0}{K}\left(\sum_{j=1}^{2}\left(\alpha\eta y_j E^1_j + E^2_j\left(2\omega\alpha A_j - \sum_{i=1}^{2}\frac{\mu_i}{(d_{i0})^2}w_{ij}\right)\right) - \frac{P}{\lambda c}E + \frac{P}{\xi}m_1\right). \tag{31}$$

6.2. Scaling. The process of *scaling* the variables (see Remark 5) is designed to improve the relative local error estimates given by formula (9).

To scale the estimates for the Cauchy problem (30), we may assign arbitrary positive values to the parameters. On the other hand, to simplify the computations it is natural to impose suitable restrictions on ξ, λ, α, and η. In particular, in the case $\gamma \le 1$ estimate (8) is much simpler to use. To guarantee this inequality, we set $\xi \le m(t_0)$, $\lambda \le \rho(t_0)/p_m(t_0)$, $\alpha \ge \alpha_0$, and $\eta \ge \eta_0$, where

$$\alpha_0 = \max_{i,j}\Bigg\{ \left|g^2_{ij}(t_0)\right|,\ \frac{1}{\beta}\left|x_{i1}(t_0)g^2_{i1}(t_0) + x_{i2}(t_0)g^2_{i2}(t_0)\right|,$$
$$\left|\frac{\omega}{2}g^1_{12}(t_0) - g^2_{11}(t_0)\right|,\ \left|\frac{\omega}{2}(g^1_{11}(t_0) - x_B) + g^2_{12}(t_0)\right| \Bigg\},$$
$$\eta_0 = \max_i\left\{\frac{\left|p^1_i(t_0)\right|}{\rho(t_0)}\right\}.$$

Then, taking the limit as $\lambda \to 0$, setting $\xi = m(t_0)$, and using formula (8), we obtain

$$s = \max\left\{\alpha c_1,\ \frac{c_2}{\alpha} + \alpha c_3 + c_4,\ \frac{c_5}{\eta} + 2\eta + c_6,\ \frac{2c_5}{\eta} + 4\eta,\ 3\eta + c_6 + 2\omega,\ 6\eta + 4\omega\right\},$$
$$c_1 = (2+5\beta)\max\left\{\frac{1}{d_1(t_0)},\ \frac{1}{d_2(t_0)}\right\},$$
$$c_2 = \frac{2}{\beta}\left(\frac{P}{\xi} + \frac{\mu_1}{(d_1(t_0))^2} + \frac{\mu_2}{(d_2(t_0))^2}\right),$$
$$c_3 = \left(2\beta + \frac{2}{\beta}\right)\max\left\{\frac{1}{d_1(t_0)},\ \frac{1}{d_2(t_0)}\right\},$$
$$c_4 = \frac{4}{\beta}\omega, \quad c_5 = 7\left(\frac{\mu_1}{(d_1(t_0))^3} + \frac{\mu_2}{(d_2(t_0))^3}\right) + \omega^2, \quad c_6 = \frac{P}{\xi c}.$$

Afterwards, we consider s as a function of α and η. We note that $\min_{\alpha,\eta} s = \min_{\Omega} s(\alpha,\eta)$, where

$$\Omega = \{(\alpha_0, \alpha_1, \alpha_2, \eta_0, \ldots, \eta_5) :\ \alpha_1 \ge \alpha_0, \alpha_2 \ge \alpha_0,\ \eta_1 \ge \eta_0, \ldots, \eta_5 \ge \eta_0\},$$
$$\alpha_1 = \sqrt{\frac{c_2}{c_3}}, \quad \alpha_2 = \frac{1}{2(c_1 - c_3)}\left(c_4 + \sqrt{(c_4)^2 + 4(c_1 - c_3)c_2}\right),$$
$$\eta_{1,2} = \frac{1}{2}\left(c_6 + 2\omega \pm \sqrt{(c_6 + 2\omega)^2 - 8c_5)}\right), \quad \eta_3 = \sqrt{\omega^2 + c_5} - \omega,$$
$$\eta_4 = \frac{1}{8}\left(c_6 - 4\omega + \sqrt{(c_6 - 4\omega)^2 + 16c_5}\right), \quad \eta_5 = \sqrt{\frac{c_5}{2}}.$$

6.3. Numerical results. At the end of this paper, a comparative study of different numerical methods for numerical integration over the interval $[\,0,\ 1000\ \text{sec}\,]$ of the initial-value problem for the optimal flight between the Earth and the Moon is summarized in Table 2.

7. Conclusions

Very tight global precision (in some benchmark examples even comparable with computer epsilon) of numerical integration methods is achieved by the Taylor series method and the new aggregative Taylor series method proposed in this paper. Both of these methods are considerably faster and more precise than the explicit Runge–Kutta methods of comparable orders. The aggregative Taylor series method turned out to be the fastest method for our benchmark examples in the case where the local tolerances are tight, while the Taylor series method has the advantage of guaranteed error and step-size estimates. Another advantage both the Taylor and aggregative Taylor series methods share over the Runge–Kutta methods is the straightforward manner in which arbitrary order methods are derived for them.

This makes both the Taylor and aggregative Taylor series methods an excellent choice when the numerical integration should yield consistent results solving the family of Cauchy problems with slowly varying parameters, including the case where a complicated function constructed from a solution of the system of ordinary differential equations (ODEs) should be optimized. The applicability of these methods is demonstrated with the real-life example of integrating the system of ODEs describing the optimal flight from the Earth to the Moon.

Table 1:
T_N: Taylor series method of order N, step-size and error estimation using (9);
AT: aggregative Taylor series method, parameters $R_{\max} = 100$, $N = 8$, $M_{\max} = 26$, $\lambda = 0.5$ (see Sec. 5.4);
DP(8)7: Dormand–Prince method, practical error estimation applied to the original initial-value problems;
DPA: Dormand–Prince method, step-size and error estimation using (13);
LocErr: local tolerable relative error;
GlobalErr: global relative error;
CPU time: integration process computational time (in units), one unit is the duration of the T_8 method.

Problem	**Method**	**GlobalErr**	**Steps**	**CPU time**
SIMPLEST	T_{12}	$2.60 \cdot 10^{-5}$	66	0.5
$t \in [0, 0.99999]$	T_8	$6.47 \cdot 10^{-5}$	155	1
LocErr$< 10^{-10}$	AT	$3.43 \cdot 10^{-4}$	26	1
	DP(8)7	$4.08 \cdot 10^{-3}$	7683	20.5
	DPA	$4.02 \cdot 10^{-3}$	7818	58
SIMPLEST	T_{12}	$7.01 \cdot 10^{-10}$	168	0.43
$t \in [0, 0.99999]$	T_8	$2.46 \cdot 10^{-9}$	572	1
LocErr$< 10^{-15}$	AT	$2.13 \cdot 10^{-11}$	48	0.43
	DP(8)7	$1.30 \cdot 10^{-5}$	2446229	3413
	DPA	$1.12 \cdot 10^{-3}$	28155	59.4
SIMPLEST	T_{12}	$1.53 \cdot 10^{-14}$	414	0.3
$t \in [0, 0.99999]$	T_8	$8.00 \cdot 10^{-14}$	2069	1
LocErr$< 10^{-20}$	AT	$4.08 \cdot 10^{-14}$	82	0.26

Problem	**Method**	**GlobalErr**	**Steps**	**CPU time**
SIMPLEST $t \in [0, 0.9999]$ LocErr$< 10^{-15}$	T_{12}	$7.02 \cdot 10^{-11}$	134	0.6
	T_8	$2.46 \cdot 10^{-10}$	458	1
	AT	$2.14 \cdot 10^{-12}$	38	0.4
	DP(8)7	$1.30 \cdot 10^{-6}$	1957182	2069
	DPA	$1.13 \cdot 10^{-4}$	22497	66.4
STIFF– LINEAR $t \in [0, 2\pi]$ LocErr$< 10^{-10}$	T_{12}	$2.46 \cdot 10^{-16}$	3935	0.7
	T_8	$2.48 \cdot 10^{-16}$	8854	1
	AT	$1.68 \cdot 10^{-11}$	1476	0.12
	DP(8)7	8.28	12	0.007
	DPA	$6.85 \cdot 10^{-14}$	215721	29.6
STIFF– LINEAR $t \in [0, 2\pi]$ LocErr$< 10^{-15}$	T_{12}	$2.45 \cdot 10^{-16}$	9540	0.5
	T_8	$2.47 \cdot 10^{-16}$	31817	1
	AT	$2.16 \cdot 10^{-16}$	1273	0.04
	DP(8)7	8.29	13	0.002
	DPA	$4.91 \cdot 10^{-14}$	775261	27.6
STIFF–LINEAR $t \in [0, 2\pi]$ LocErr$< 10^{-20}$	T_{12}	$2.44 \cdot 10^{-16}$	23130	0.33
	T_8	$2.46 \cdot 10^{-16}$	114345	1
	AT	$3.20 \cdot 10^{-16}$	1271	0.016
STIFF– CAPS $t \in [0, 0.5]$ LocErr$< 10^{-10}$	T_{12}	$5.15 \cdot 10^{-19}$	38759	0.75
	T_8	$1.17 \cdot 10^{-18}$	87204	1
	AT	$6.27 \cdot 10^{-16}$	14534	0.08
	DP(8)7	$2.09 \cdot 10^{3}$	3589	0.18
	DPA	$1.20 \cdot 10^{-12}$	2124967	25.8
STIFF– CAPS $t \in [0, 0.5]$ LocErr$< 10^{-15}$	T_{12}	$1.20 \cdot 10^{-18}$	93968	0.5
	T_8	$5.97 \cdot 10^{-18}$	313395	1
	AT	$2.34 \cdot 10^{-18}$	12536	0.025
	DP(8)7	$2.09 \cdot 10^{3}$	1127284	13.4
STIFF–CAPS $t \in [0, 0.5]$ LocErr$< 10^{-20}$	T_{12}	$6.62 \cdot 10^{-18}$	227822	0.34
	T_8	$2.63 \cdot 10^{-18}$	1126282	1
	AT	$1.96 \cdot 10^{-18}$	12515	0.01
STIFF–CAPS $t \in [0, 0.001]$ LocErr$< 10^{-20}$	T_{12}	$4.34 \cdot 10^{-19}$	456	0.33
	T_8	$1.63 \cdot 10^{-18}$	2253	1
	AT	$1.63 \cdot 10^{-19}$	26	0.01
	DPA	$5.27 \cdot 10^{-16}$	54892	26.9
JACOB $t \in [0, 100K]$ LocErr$< 10^{-10}$	T_{12}	$1.39 \cdot 10^{-11}$	974	0.72
	T_8	$3.19 \cdot 10^{-10}$	2192	1
	AT	$1.23 \cdot 10^{-8}$	365	0.65
	DP(8)7	$1.51 \cdot 10^{-5}$	77938	12.5
	DPA	$8.91 \cdot 10^{-6}$	106782	65
JACOB $t \in [0, 100K]$ LocErr$< 10^{-15}$	T_{12}	$4.33 \cdot 10^{-15}$	2362	0.48
	T_8	$1.37 \cdot 10^{-15}$	7875	1
	AT	$4.53 \cdot 10^{-14}$	406	0.3
	DPA	$2.48 \cdot 10^{-6}$	383755	64.4

Problem	Method	GlobalErr	Steps	CPU time
JACOB	T_{12}	$4.25 \cdot 10^{-15}$	5725	0.33
$t \in [0, 100K]$	T_8	$4.03 \cdot 10^{-15}$	28301	1
LocErr$< 10^{-20}$	AT	$4.21 \cdot 10^{-15}$	667	0.13
JACOB	T_{12}	$1.69 \cdot 10^{-16}$	95	0.5
$t \in [0, 4K]$	T_8	$1.81 \cdot 10^{-16}$	315	1
LocErr$< 10^{-15}$	AT	$5.52 \cdot 10^{-17}$	17	0.25
	DP(8)7	$5.52 \cdot 10^{-10}$	987266	1346
	DPA	$3.54 \cdot 10^{-8}$	15351	62.75
VDPL	T_{12}	$1.03 \cdot 10^{-12}$	19152	0.72
$t \in [0, 100T]$	T_8	$1.09 \cdot 10^{-10}$	43092	1
LocErr$< 10^{-10}$	AT	$7.39 \cdot 10^{-10}$	7180	0.33
	DP(8)7	$1.71 \cdot 10^{-5}$	764560	59.3
	DPA	$5.93 \cdot 10^{-4}$	1873519	51
VDPL	T_{12}	$4.00 \cdot 10^{-15}$	46435	0.48
$t \in [0, 100T]$	T_8	$9.26 \cdot 10^{-15}$	154866	1
LocErr$< 10^{-15}$	AT	$3.23 \cdot 10^{-14}$	6194	0.13
VDPL	T_{12}	$4.07 \cdot 10^{-15}$	112579	0.32
$t \in [0, 100T]$	T_8	$1.51 \cdot 10^{-15}$	556558	1
LocErr$< 10^{-20}$	AT	$6.15 \cdot 10^{-15}$	6633	0.05
VDPL	T_{12}	$2.24 \cdot 10^{-16}$	1126	0.33
$t \in [0, T]$	T_8	$2.25 \cdot 10^{-16}$	5566	1
LocErr$< 10^{-20}$	AT	$2.22 \cdot 10^{-16}$	67	0.06
	DPA	$5.46 \cdot 10^{-9}$	241974	54
BRUS $t \in [0, 126.5]$ LocErr$< 10^{-34}$	T_{30}	0	62406	
BRUS	T_{12}	$9.48 \cdot 10^{-14}$	30276	0.75
$t \in [0, 126.5]$	T_8	$2.87 \cdot 10^{-12}$	68124	1
LocErr$< 10^{-10}$	AT	$9.15 \cdot 10^{-11}$	11254	0.16
	$DP(8)7$	$2.87 \cdot 10^{-8}$	99394	0.19
	DPA	$3.89 \cdot 10^{-7}$	2419707	37
BRUS	T_{12}	$3.05 \cdot 10^{-19}$	73408	0.5
$t \in [0, 126.5]$	T_8	$3.14 \cdot 10^{-16}$	244825	1
LocErr$< 10^{-15}$	AT	$2.68 \cdot 10^{-16}$	9802	0.06
BRUS	T_{12}	$8.90 \cdot 10^{-18}$	177975	0.34
$t \in [0, 126.5]$	T_8	$2.55 \cdot 10^{-18}$	879854	1
LocErr$< 10^{-20}$	AT	$4.60 \cdot 10^{-17}$	9909	0.03
BRUS $t \in [0, 20]$ LocErr$< 10^{-34}$	T_{30}	0	9277	1.87
BRUS	T_{12}	$2.89 \cdot 10^{-16}$	10912	0.5
$t \in [0, 20]$	T_8	$5.69 \cdot 10^{-15}$	36294	1
LocErr$< 10^{-15}$	AT	$2.56 \cdot 10^{-15}$	1457	0.06
	$DP(8)7$	$6.56 \cdot 10^{-9}$	4724964	19
	DPA	$2.09 \cdot 10^{-6}$	1265040	36

Table 2.
T_N: Taylor series method of order N, step size and error estimation using (9);
AT: aggregative Taylor series method, parameters $R_{\max} = 100$, $N = 8$, $M_{\max} = 26$, $\lambda = 0.5$ (see Sec. 5.4);
DP(8)7: Dormand–Prince method, practical error estimation applied to the original initial value problems;
DPA: Dormand–Prince method, step size and error estimation using (13);
LocErr: local tolerable relative error;
CPU time: integration process computational time (in units), one unit is the duration of the T_8 method;
IntegralErr $= |(H(1000) - H(0))/H(0)|$ (see (31)).

LocErr	**Method**	**IntegralErr**	**Steps**	**CPU time**
10^{-10}	T_{12}	$5.49 \cdot 10^{-18}$	47	0.78
	T_8	$4.86 \cdot 10^{-17}$	106	1
	AT	$3.25 \cdot 10^{-12}$	18	0.25
	$DP(8)7$	$1.74 \cdot 10^{-8}$	594	0.29
	DPA	$5.41 \cdot 10^{-8}$	5131	9.06
10^{-13}	T_{12}	$7.21 \cdot 10^{-18}$	80	0.62
	T_8	$5.04 \cdot 10^{-18}$	227	1
	AT	$2.71 \cdot 10^{-15}$	16	0.15
	DP8(7)	$5.53 \cdot 10^{-10}$	18457	4.17
	DPA	$5.16 \cdot 10^{-8}$	11053	9.14
10^{-15}	T_{12}	$6.24 \cdot 10^{-18}$	114	0.53
	T_8	$6.91 \cdot 10^{-18}$	379	1
	AT	$1.06 \cdot 10^{-18}$	16	0.11
	$DP(8)7$	$5.53 \cdot 10^{-11}$	184502	24.98
	DPA	$5.06 \cdot 10^{-8}$	18437	9.13

REFERENCES

1. L. K. Babadzanjanz, "Existence of the continuations and representation of the solutions in celestial mechanics," Tr. ITA Akad. Nauk SSSR, **XVII**, 3–45 (1978).
2. L. K. Babadzanjanz, "Existence of the continuations in the N-body problem," *Celestial Mechanics*, **20**, 43–57 (1979).
3. L. K. Babadzanjanz, "Error estimation for numerical solution of the N-body problem," *Letters to AG*, **7**, No. 12, 752–755 (1981).
4. L. K. Babadzanjanz and P. B. Mgojan, "Estimates of holomorphic solutions of ordinary differential equations," *Izv. Akad Nauk Arm. SSR. Ser. Mat.*, **17**, No. 2, 83–91 (1982).
5. L. K. Babadzanjanz and P. B. Mgojan, "Estimates of holomorphic solutions of ordinary differential equations," Vestn. LGU, **7**, 9–16 (1984).
6. L. K. Babadzanjanz, "On the global solution of the N-body problem," *Celestial Mechanics*, **56**, 427–449 (1993).
7. K. Dekker and J. G. Verwer, *Stability of Runge–Kutta Methods for Stiff Nonlinear Differential Equations*, Elsevier Science Publishers B. V., North-Holland, Amsterdam–New York–Oxford (1984).
8. E. Hairer, S. P. Norsett, and G. Wanner, *Solving Ordinary Differential Equations* I. *Nonstiff Problems*, Springer-Verlag, Berlin–Heildeberg (1987).
9. D. E. Okhotsimskii and Yu. G. Sikharulidze, *Foundations of Space Flight Mechanics* [in Russian], Nauka, Moscow (1994).

10. H. Poincaré, “Sur les courbes definies par les equations differentielles (IV partie),” *J. Math. Pures Appl.* (4), **2**, 151–217 (1886); Oeuvres, Gauthier-Villars, Paris, **1**, 167–178, 181 (1928).
11. P. J. Prince and J. R. Dormand, “High order embedded Runge–Kutta formulae,” *J. Comput. Appl. Math.*, **7**, 67–75 (1981).

L. K. Babadzhanjanz
Department of Applied Mathematics
and Processes of Management,
St. Petersburg State University, Russia
E-mail: `voidcaller@mail.domonet.ru`
D. R. Sarkissian
Department of Mathematics and Statistics,
Mississippi State University, MS 39765
E-mail: `voidcaller@mail.domonet.ru`