# Stable Local-Global Stepsize Control for Runge-Kutta Methods*

G.Yu. Kulikov[†]

**Abstract**

In this paper we develop a new procedure to control stepsize for Runge-Kutta methods applied to both ordinary differential equations and semi-explicit index 1 differential-algebraic equations. In contrast to the standard approach, the error control mechanism presented here is based on monitoring and controlling both the local and global errors of Runge-Kutta formulas. As a result, Runge-Kutta methods with the local-global stepsize control solve differential or differential-algebraic equations with any prescribed accuracy (up to round-off errors).

For implicit Runge-Kutta formulas we give the sufficient number of both full and modified Newton iterations allowing the iterative approximations to be correctly used in the procedure of the local-global stepsize control. In addition, we develop a stable local-global error control mechanism which is applicable for stiff problems. Numerical tests support the theoretical results of the paper.

**Key words.** differential-algebraic equations, Runge-Kutta methods, asymptotic error expansions, error control mechanism

**AMS subject classifications.** 65L06

## 1 Introduction

Many numerical methods are available for solving systems of differential-algebraic equations (DAEs) of the form:

$$x'(t) = g(x(t), y(t), \alpha(t)), \tag{1.1a}$$

$$y(t) = f(x(t), y(t), \alpha(t)), \tag{1.1b}$$

*This work was supported in part by the Russian Academy of Sciences and the Ministry of General and Professional Education of Russia

[†]Department of Mathematics and Mechanics, Ulyanovsk State University, 432700 Ulyanovsk, Russia. E-mail: KulGYu@mci.univ.simbirsk.su

$$x(t_0) = x^0,\ y(t_0) = y^0 \tag{1.1c}$$

where $t \in [t_0, t_0 + T]$, $x(t) \in \mathbf{R}^m$, $y(t) \in \mathbf{R}^n$, $\alpha(t) \in \mathbf{R}^l$, $\alpha(t)$ is a known vector, $g : D \subset \mathbf{R}^{m+n+l} \to \mathbf{R}^m$, $f : D \subset \mathbf{R}^{m+n+l} \to \mathbf{R}^n$, and where initial conditions (1.1c) are consistent: $y^0 = f(x^0, y^0, \alpha(t_0))$.

Several authors have shown that discrete schemes based on implicit one-step (or multistep) fixed-stepsize methods have the same orders as the underlying methods if problem (1.1) is of index 1 [1], [3], [5]–[10], [12], [13], [15]–[17]. Unfortunately, the fixed-stepsize implementation of such methods is difficult to use, as a user has to understand many details about the method and the problem before he or she can gain confidence in the accuracy of the corresponding discrete solution.

The variable-stepsize Runge-Kutta methods were proposed in [13] for solving the index 1 differential-algebraic system (1.1). Such methods based on the classical stepsize control allow for the automatic determination of the "optimal" stepsize in the course of computation. In [2] the local error control procedure improved for multistep methods applied to ODEs with invariants and DAEs in multibody dynamics. On the other hand, the relationship between the accuracy obtained and the specified tolerance is generally extremely sensitive to both the problem and the method. By this reason we develop a new error control mechanism for Runge-Kutta methods in the paper. It is based on monitoring and controlling both the local and global errors of the discrete approximation. The new stepsize control is aimed to be capable of keeping the global error of the numerical solution as small as we want. In this case a user may know nothing about the problem and the method. The user only needs to set the required accuracy which will be reached automatically in the course of computation. Thus, we obtain Runge-Kutta methods with the guaranteed error[1].

The remainder of this paper is organized as follows: In section 2 we present the local-global stepsize control for Runge-Kutta methods. In section 3 we construct iterative Runge-Kutta methods with both trivial and nontrivial predictors. We also study how to use the iterative approximations from these methods in the error control procedure. In section 4 we examine the local-global stepsize control for a stiff problem and derive a stable controller which is appropriate for stiff semi-explicit DAEs. In section 5 we simplify the local-global stepsize control mechanism in order to reduce computing time. In the final section we discuss the results presented in the paper.

# 2 A local-global stepsize control

We will further consider a system of DAEs of the form

$$x'(t) = g(x(t), y(t)), \tag{2.1a}$$

$$y(t) = f(x(t), y(t)), \tag{2.1b}$$

[1]In the paper we deal with problems when the interval of integration is not very large

$$x(0) = x^0,\ y(0) = y^0, \tag{2.1c}$$

where $t \in [0,T]$, $x(t) \in \mathbf{R}^m$, $y(t) \in \mathbf{R}^n$, $g : D \subset \mathbf{R}^{m+n} \to \mathbf{R}^m$, $f : D \subset \mathbf{R}^{m+n} \to \mathbf{R}^n$, and where initial conditions (2.1c) are consistent; i.e., $y^0 = f(x^0, y^0)$. We note that we will consider only autonomous systems because any nonautonomous system may be converted to an autonomous one by introducing a new independent variable.

To solve problem (2.1) numerically we introduce a nonuniform grid

$$\omega_\tau = \{t_{k+1} = t_k + \tau_k,\ k = 0, 1, ..., K-1,\ t_0 = 0,\ t_K = T\}$$

in the interval $[0,T]$ and define $\tau$ as the diameter of the grid $\omega_\tau$

$$\tau = \max_{0 \le k \le K-1} \{\tau_k\}.$$

Applying an $l$-stage Runge-Kutta (RK) method given by the table

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array},$$

where $A$ is a real matrix of dimension $l \times l$, $c$ and $b$ are real vectors of dimension $l$, to problem (2.1), we obtain the following discrete problem:

$$x_{ki} = x_k + \tau_k \sum_{j=1}^{l} a_{ij} g(x_{kj}, y_{kj}), \tag{2.2a}$$

$$y_{ki} = f(x_{ki}, y_{ki}), \quad i = 1, 2, ..., l, \tag{2.2b}$$

$$x_{k+1} = x_k + \tau_k \sum_{i=1}^{l} b_i g(x_{ki}, y_{ki}), \tag{2.2c}$$

$$y_{k+1} = f(x_{k+1}, y_{k+1}), \quad k = 0, 1, ..., K-1, \tag{2.2d}$$

$$x_0 = x^0,\ y_0 = y^0. \tag{2.2e}$$

We assume that the RK formula is of order $s$, and that problem (2.1) satisfies the following conditions on the compact set $D_1 \subset \mathbf{R}^{m+n}$:

I. *The smoothness condition.* The mapping $G : D_1 \subset \mathbf{R}^{m+n} \to \mathbf{R}^{m+n}$ has continuous partial derivatives of orders $1, 2, ..., s+2$ on the set $D_1$, where $G$ is obtained by combining the mappings $g$ and $f$; i.e., $G \stackrel{\text{def}}{=} (g^T, f^T)^T$.
II. *The nonsingularity condition.* The matrix $I_n - \partial_y f(x,y)$ is nonsingular for any

$z \stackrel{\text{def}}{=} (x^T, y^T)^T \in D_1$ where $I_n$ is the identity matrix of dimension $n$.
III. *The inclusion condition* [2]. There exists a convex set $D_0$ such that $z^0 \stackrel{\text{def}}{=} ((x^0)^T, (y^0)^T)^T \in D_0 \subset D_1$.

It is known [5], [6], [7], [9], [12], [13] that if conditions I–III hold then problem (2.1) is of index 1 and has a unique solution $z(t) \stackrel{\text{def}}{=} (x(t)^T, y(t)^T)^T \in D_0$, and then method (2.2) is of order s. Moreover, it was proved in [13] that the local error of method (2.2) has the form

$$x(t_{k+1}) - \tilde{x}_{k+1} = \check{\psi}_{s+1}(t_k)\tau_k^{s+1} + O(\tau_k^{s+2}), \tag{2.3a}$$

$$y(t_{k+1}) - \tilde{y}_{k+1} = \check{\phi}_{s+1}(t_k)\tau_k^{s+1} + O(\tau_k^{s+2}), \tag{2.3b}$$

$k = 0, 1, \ldots, K-1$, where $\tilde{z}_{k+1} \stackrel{\text{def}}{=} \left((\tilde{x}_{k+1})^T, (\tilde{y}_{k+1})^T\right)^T$ is the solution of (2.2) under the condition $z_k = z(t_k)$. Then from theorem 2 in [14], we obtain the asymptotic expansion for the global error of method (2.2):

$$x(t_{k+1}) - x_{k+1} = \psi_s(t_{k+1})(\tau_k^*)^s + O((\tau_k^*)^{s+1}), \tag{2.4a}$$

$$y(t_{k+1}) - y_{k+1} = \phi_s(t_{k+1})(\tau_k^*)^s + O((\tau_k^*)^{s+1}), \tag{2.4b}$$

$k = 0, 1, \ldots, K-1$, where the pair of the functions $\psi_s(t)$ and $\phi_s(t)$ is a solution of the DAEs

$$\psi_s'(t) = \partial_x g(x(t), y(t))\psi_s(t) + \partial_y g(x(t), y(t))\phi_s(t) + \check{\psi}_{s+1}(t), \tag{2.5a}$$

$$\phi_s(t) = \partial_x f(x(t), y(t))\psi_s(t) + \partial_y f(x(t), y(t))\phi_s(t), \tag{2.5b}$$

$$\psi_s(0) = 0, \quad \phi_s(0) = 0. \tag{2.5c}$$

Thus, regarding $\check{\psi}_{s+1}(t)$ as a known function because we can use the Richardson extrapolation or two RK methods of different orders to compute it [13], formula (2.5) gives a way to determine the principal term of the global error of method (2.2). Moreover, it follows from (2.4) that while computing the functions $\psi_s(t)$ and $\phi_s(t)$ at the point $t_{k+1}$ we can limit ourselves by accuracy $O(\tau_k^*)$ if $\tau_k^*$ is sufficiently small. Therefore, taking into account possible stiffness of problem (2.5), we apply the implicit Euler method to this problem and obtain the following recursion relation for the approximate computation of the functions $\psi_s(t)$ and $\phi_s(t)$:

$$\begin{pmatrix} \tilde{\psi}_s(t_{k+1}) \\ \tilde{\phi}_s(t_{k+1}) \end{pmatrix} = P_{k+1}^{-1} \begin{pmatrix} \tilde{\psi}_s(t_k) + \tau_k \hat{\psi}_{s+1}(t_{k+1}) \\ 0 \end{pmatrix}, \tag{2.6a}$$

[2] The inclusion $D_0 \subset D_1$ implies that $D_0$ is contained in $D_1$ together with some neighborhood

$$\tilde{\psi}_s(0) = 0, \quad \tilde{\phi}_s(0) = 0, \quad k = 0, 1, \ldots, K-1, \tag{2.6b}$$

where

$$P_{k+1} \stackrel{\text{def}}{=} \begin{pmatrix} I_m - \tau_k \partial_x g(\tilde{x}_{k+1}, \tilde{y}_{k+1}) & -\tau_k \partial_y g(\tilde{x}_{k+1}, \tilde{y}_{k+1}) \\ -\partial_x f(\tilde{x}_{k+1}, \tilde{y}_{k+1}) & I_n - \partial_y f(\tilde{x}_{k+1}, \tilde{y}_{k+1}) \end{pmatrix}.$$

In formula (2.6a), 0 denotes the null vector of dimension $n$. Here we have used (2.3) and the fact that all the ways to compute $\check{\psi}_{s+1}(t_{k+1})$ give at least accuracy $O(\tau_k)$; i.e., $\hat{\psi}_{s+1}(t_{k+1}) = \check{\psi}_{s+1}(t_{k+1}) + O(\tau_k)$. Hence, formulas (2.6) yield

$$\begin{pmatrix} \tilde{\psi}_s(t_{k+1}) \\ \tilde{\phi}_s(t_{k+1}) \end{pmatrix} = \begin{pmatrix} \psi_s(t_{k+1}) \\ \phi_s(t_{k+1}) \end{pmatrix} + O(\tau_k) \tag{2.7}$$

if the matrix $P_{k+1}$ is nonsingular for $k = 0, 1, \ldots, K-1$.

Now we require

$$0 < \omega \le \frac{\tau_k^*}{\tau_k} \le \Omega < \infty \tag{2.8}$$

where $\omega$ and $\Omega$ are constants. From estimate (2.7) and condition (2.8) it follows that

$$\begin{pmatrix} \tilde{\psi}_s(t_{k+1}) \\ \tilde{\phi}_s(t_{k+1}) \end{pmatrix} = \begin{pmatrix} \psi_s(t_{k+1}) \\ \phi_s(t_{k+1}) \end{pmatrix} + O(\tau_k^*). \tag{2.9}$$

This allows relation (2.6) to be used for the computation of the leading term of the global error of method (2.2).

The next important stage in the numerical integration of the semi-explicit DAEs of index 1 is choosing the maximum stepsize keeping the truncated local and global errors bounded. Let $\epsilon_l$ and $\epsilon_g$ be given limits for the local and global errors of method (2.2), respectively. Considering that the global error of the corrected numerical solution at the point $t_k$ does not exceed $\epsilon_g$, we will now show how to compute the approximate solution of problem (2.1) at the point $t_{k+1}$ with the given accuracy.

We first compute the local error at the point $t_{k+1}$, by using any classical approach, with the accuracy $\epsilon_l$. After that we determine the coefficient of the principal term of the global error by formulas (2.6). Then we find the maximum stepsize $\tau_k^*$ guaranteeing the necessary accuracy

$$p = \left[\tau_k \left( \|(\tilde{\psi}_s(t_{k+1})^T, \tilde{\phi}_s(t_{k+1})^T)^T\| / \epsilon_g \right)^{\frac{1}{s}}\right] + 1, \tag{2.10a}$$

$$\tau_k^* = \tau_k / p. \tag{2.10b}$$

Further, we integrate problem (2.1) in the interval $[t_k, t_{k+1}]$ by method (2.2) with the fixed stepsize $\tau_k^*$ and obtain the numerical solution at the point $t_{k+1}$ with the accuracy $\epsilon_g$. In addition, we add the estimated global error to the approximate solution increasing the order of method (2.2) at least by 1. Now we set the next stepsize

$$\tau_{k+1} = \tau_k^* \tag{2.11}$$

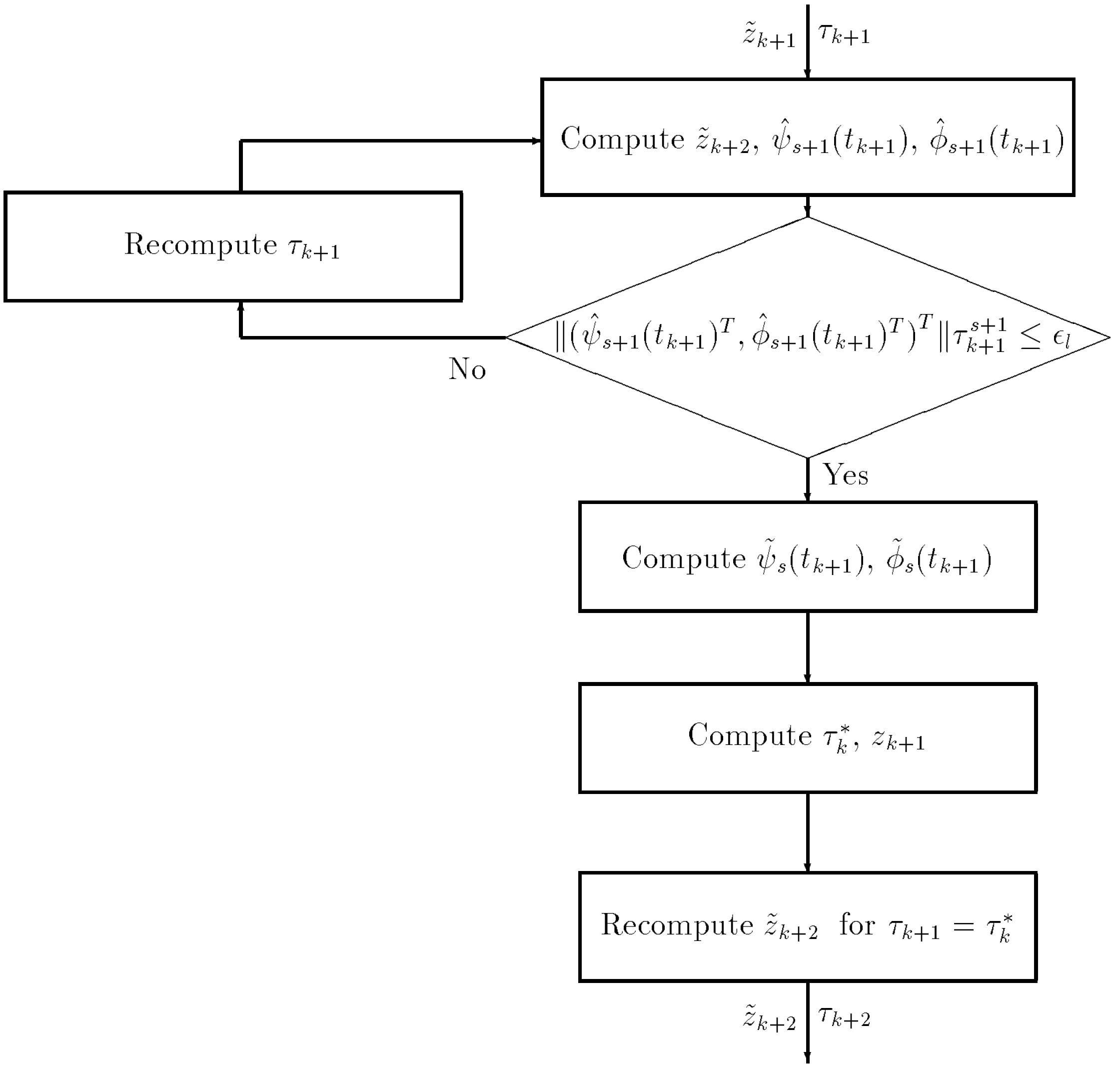


Figure 1: The local-global stepsize control procedure.

and repeat all computations (see fig. 1). Formula (2.11) provides (2.8) and hence (2.9).

Finally, we note that, by applying the trapezoidal rule to system (2.5), we derive the exacter controller without extra computations

$$\begin{pmatrix} \tilde{\psi}_s(t_{k+1}) \\ \tilde{\phi}_s(t_{k+1}) \end{pmatrix} = Q_{k+1}^{-1} R_k \begin{pmatrix} \tilde{\psi}_s(t_k) \\ \tilde{\phi}_s(t_k) \end{pmatrix} + \frac{\tau_k}{2} Q_{k+1}^{-1} \begin{pmatrix} \hat{\psi}_{s+1}(t_k) + \hat{\psi}_{s+1}(t_{k+1}) \\ 0 \end{pmatrix} \tag{2.12a}$$

$$\tilde{\psi}_s(0) = 0, \quad \tilde{\phi}_s(0) = 0, \quad k = 0, 1, \ldots, K-1, \tag{2.12b}$$

where

$$Q_{k+1} \stackrel{\text{def}}{=} \begin{pmatrix} I_m - \frac{\tau_k}{2}\partial_x g(\tilde{x}_{k+1}, \tilde{y}_{k+1}) & -\frac{\tau_k}{2}\partial_y g(\tilde{x}_{k+1}, \tilde{y}_{k+1}) \\ -\partial_x f(\tilde{x}_{k+1}, \tilde{y}_{k+1}) & I_n - \partial_y f(\tilde{x}_{k+1}, \tilde{y}_{k+1}) \end{pmatrix},$$

$$R_k \stackrel{\text{def}}{=} \begin{pmatrix} I_m + \frac{\tau_k}{2}\partial_x g(\tilde{x}_k, \tilde{y}_k) & \frac{\tau_k}{2}\partial_y g(\tilde{x}_k, \tilde{y}_k) \\ 0_m & 0_n \end{pmatrix},$$

and where $0_m$ and $0_n$ are the null matrices of dimensions $n \times m$ and $n \times n$, respectively.

# 3 Implicit RK methods

We still assumed that the approximate solution from method (2.2) had been known exactly. However, it is true only when problem (2.1) is linear. If we take an implicit RK formula and apply it to a nonlinear problem (2.1), we get (2.2) as a system of nonlinear algebraic equations. When using an iterative method to solve the system, we may find only an iterative approximation to the exact solution of (2.2). Therefore it is important to study how to use the approximate solution of (2.2) in the local-global stepsize control.

Let us denote the approximate solution of problem (2.2) obtained after $N$ iterations of some iterative method by $\bar{z}_k \stackrel{\text{def}}{=} \bar{z}_k(N)$, $k = 1, 2, \ldots, K$. In addition, we introduce the vector

$$Z_{k+1} \stackrel{\text{def}}{=} \left((z_{k1})^T, \ldots, (z_{kl})^T, (z_{k+1})^T\right)^T \in \mathbf{R}^{(m+n)(l+1)}$$

and define the mapping

$$\bar{G}_k^\tau : D \subset \mathbf{R}^{(m+n)(l+1)} \to \mathbf{R}^{(m+n)(l+1)}, k = 0, 1, \ldots, K-1,$$

by the formula

$$\bar{G}_k^\tau Z_{k+1} \stackrel{\text{def}}{=} \Bigg( \Big(\bar{x}_k + \tau_k \sum_{j=1}^{l} a_{1j} g(z_{kj})\Big)^T, \ f(z_{k1})^T, \ \ldots, \ \Big(\bar{x}_k + \tau_k \sum_{j=1}^{l} a_{lj} g(z_{kj})\Big)^T, \ f(z_{kl})^T,$$
$$\Big(\bar{x}_k + \tau_k \sum_{i=1}^{l} b_i g(z_{ki})\Big)^T, \ f(z_{k+1})^T \Bigg)^T.$$

Having applied the full and modified Newton iterations to problem (2.2), we derive two classes of iterative RK methods solving semi-explicit index 1 differential-algebraic equations [16]. More precisely, with notations introduced above we have:

*The Runge-Kutta-Newton (RKN) method* with nontrivial predictor

$$(3.1a)\qquad Z^i_{k+1} = Z^{i-1}_{k+1} - \partial \bar{F}^\tau_k (Z^{i-1}_{k+1})^{-1} \bar{F}^\tau_k Z^{i-1}_{k+1}, \quad i = 1, 2, ..., N,$$

$$(3.1b)\qquad \begin{aligned} Z^0_{k+1} &= \left(\bar{H}_q(t_k + c_1\tau_k)^T, ..., \bar{H}_q(t_k + c_l\tau_k)^T, \bar{H}_q(t_{k+1})^T\right)^T \in \mathbf{R}^{(m+n)(l+1)}, \\ & k = q, q+1, ..., K-1, \end{aligned}$$

$$(3.1c)\qquad \bar{Z}_k = Z_k = ((z_k)^T, ..., (z_k)^T)^T \in \mathbf{R}^{(m+n)(l+1)}, \quad k = 0, 1, ..., q,$$

where $\bar{F}^\tau_k \overset{\text{def}}{=} I_{(m+n)(l+1)} - \bar{G}^\tau_k$ and $\partial \bar{F}^\tau_k (Z^{i-1}_{k+1})$ is the Jacobian of the mapping $\bar{F}^\tau_k$ at the point $Z^{i-1}_{k+1}$, and where $\bar{H}_q(t)$ is the Newton interpolation formula of degree $q$ based on the last $q+1$ points $\bar{z}_j$, $j = k, k-1, \dots, k-q$. The starting values $z_j$, $j = 0, 1, \dots, q$, are assumed to be known.

*The Runge-Kutta-modified Newton (RKmN) method* with nontrivial predictor

$$(3.2a)\qquad Z^i_{k+1} = Z^{i-1}_{k+1} - \partial \bar{F}^\tau_k (Z^0_{k+1})^{-1} \bar{F}^\tau_k Z^{i-1}_{k+1}, \quad i = 1, 2, ..., N,$$

$$(3.2b)\qquad \begin{aligned} Z^0_{k+1} &= \left(\bar{H}_q(t_k + c_1\tau_k)^T, ..., \bar{H}_q(t_k + c_l\tau_k)^T, \bar{H}_q(t_{k+1})^T\right)^T \in \mathbf{R}^{(m+n)(l+1)}, \\ & k = q, q+1, ..., K-1, \end{aligned}$$

$$(3.2c)\qquad \bar{Z}_k = Z_k = ((z_k)^T, ..., (z_k)^T)^T \in \mathbf{R}^{(m+n)(l+1)}, \quad k = 0, 1, ..., q.$$

If we set $q = 0$ then $\bar{H}_0(t) \equiv \bar{z}_k$ and we obtain methods (3.1) and (3.2) with trivial predictor.

We assume below that the starting values are given with accuracy $O(\tau^r)$; i.e.,

$$(3.3)\qquad \|z(t_k) - z_k\| \le C_r \tau^r, \quad k = 0, 1, ..., q,$$

where $r \ge 1$ and $C_r$ is a constant.

In paper [16] we have proved that the RKN and RKmN methods are convergent if conditions I–III hold and the diameter $\tau$ is sufficiently small. Moreover, their global error estimates are:

$$(3.4)\qquad \|z(t_k) - \bar{z}_k(N)\| \le C_\mu \tau^\mu, \quad k = q+1, q+2, \dots, K,$$

where $\mu = \min\{\xi 2^N - 1, \zeta\}$, $\xi = \min\{q+1, \zeta, \nu\}$, $\zeta = \min\{r, s\}$, and where $s$ and $\nu$ are classical and stage orders of the underlying RK formula, respectively, $C_\mu$ is a constant;

$$(3.5)\qquad \|z(t_k) - \bar{z}_k(N)\| \le C_\mu \tau^\mu, \quad k = q+1, q+2, \dots, K,$$

where $\mu = \min\{\xi(N+1) - 1, \zeta\}$ and the remaining parameters are the same as those in (3.4).

The local-global stepsize control will work correctly for methods (3.1) and (3.2) if the error of the approximate solution of (2.2) does not exceed the errors in the computation of both the local and global errors of method (2.2). More precisely, from (3.3) and (3.4) it follows that we have to require

$$r \geq s+1 \tag{3.6a}$$

and

$$N \geq \log_2\Big(\frac{s+2}{\xi}\Big) \tag{3.6b}$$

for the variable-stepsize RKN method with the local-global stepsize control. Error estimate (3.5) gives the similar conditions for the RKmN method:

$$r \geq s+1, \tag{3.7a}$$

$$N \geq \frac{s+2}{\xi} - 1. \tag{3.7b}$$

We note that conditions (3.6a) and (3.7a) are not essential for the RKN and RKmN methods. If we know the initial values of problem (2.1) exactly, we will compute the necessary starting values by the methods with trivial predictor, keeping the number of iterations $N$ sufficiently large (see formulas (3.21) and (3.22) in [13]), and then (3.6a) and (3.7a) will be fulfilled. Thus, we have only to make enough iterations (as in formulas (3.6b) and (3.7b)) per time point to use the solutions from methods (3.1) and (3.2) in the local-global stepsize control mechanism.

We will now test the variable-stepsize RKN and RKmN methods with the local-global stepsize control on the following problem:

$$x1'(t) = 10t \cdot \exp\Big(5(y2(t)-1)\Big) \cdot x2(t), \tag{3.8a}$$

$$x2'(t) = -2t \cdot \ln(y1(t)), \tag{3.8b}$$

$$y1(t) = x1(t)^{\frac{1}{5}}, \tag{3.8c}$$

$$y2(t) = (x2(t)^2 + y2(t)^2)/2 \tag{3.8d}$$

where $t \in [1.0708712, 1.4123836]$, and where the initial conditions are defined by

$$x1(1.0708712) = \exp(5\sin(1.0708712^2)), \quad x2(1.0708712) = \cos(1.0708712^2),$$

$$y1(1.0708712) = \exp(\sin(1.0708712^2)), \quad y2(1.0708712) = \sin(1.0708712^2) + 1.$$

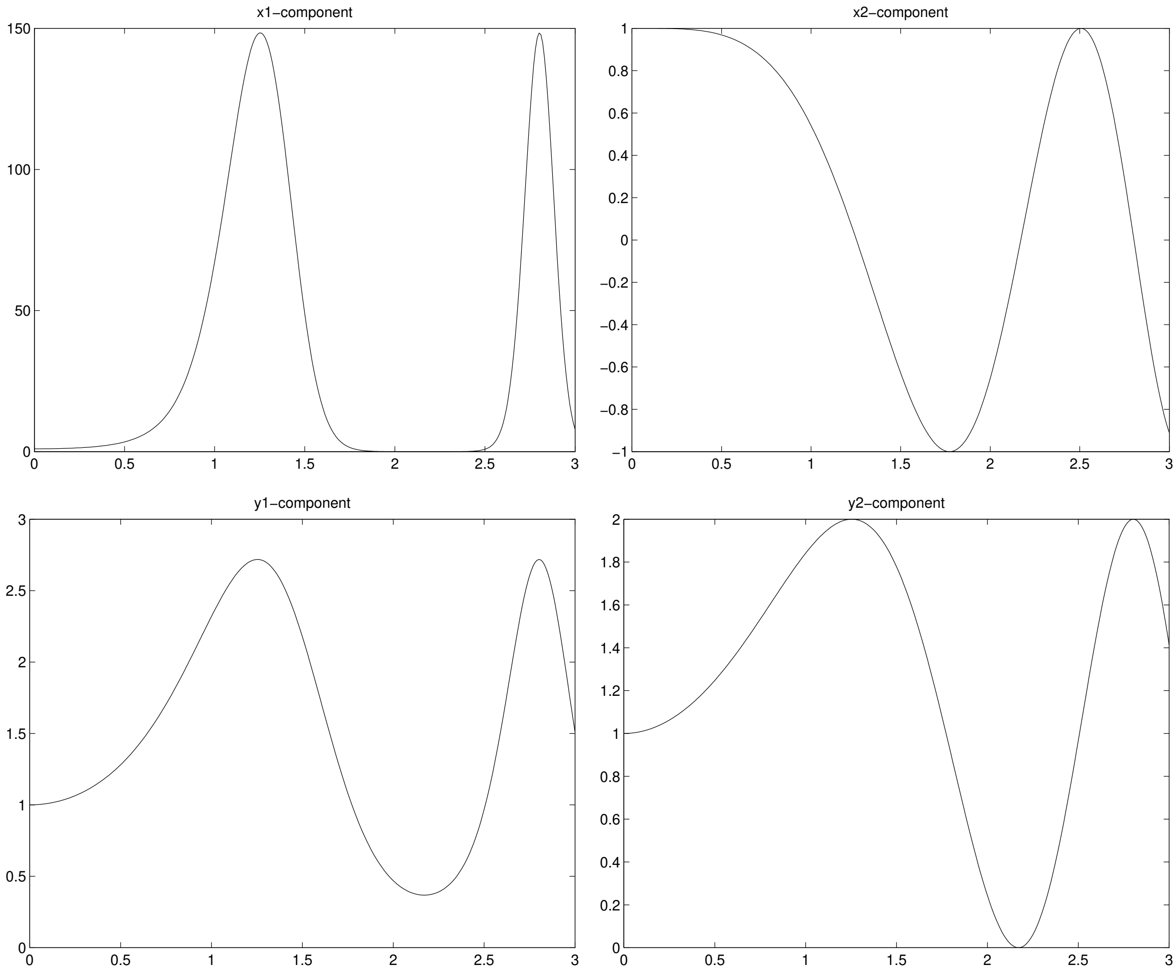


Figure 2: The solution of problem (3.8).

Problem (3.8) has the exact solution (see fig. 2):

$$x1(t) = \exp(5\sin(t^2)), \quad x2(t) = \cos(t^2), \tag{3.9a}$$

$$y1(t) = \exp(\sin(t^2)), \quad y2(t) = \sin(t^2) + 1. \tag{3.9b}$$

We can easily see that problem (3.8) satisfies conditions I–III in a neighbourhood of solution (3.9) when $t \in [1.0708712, 1.4123836]$. Therefore all theoretical results of the paper hold. We use the Hammer and Hollingsworth method of order four

$$\begin{array}{c|cc}
\frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\
\frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \\
\hline
 & \frac{1}{2} & \frac{1}{2}
\end{array}$$

and the Kuntzmann and Butcher one of order six [4]

$$\begin{array}{c|ccc}
\frac{1}{2} - \frac{\sqrt{15}}{10} & \frac{5}{36} & \frac{2}{9} - \frac{\sqrt{15}}{15} & \frac{5}{36} - \frac{\sqrt{15}}{30} \\
\frac{1}{2} & \frac{5}{36} + \frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36} - \frac{\sqrt{15}}{24} \\
\frac{1}{2} + \frac{\sqrt{15}}{10} & \frac{5}{36} + \frac{\sqrt{15}}{30} & \frac{2}{9} + \frac{\sqrt{15}}{15} & \frac{5}{36} \\
\hline
 & \frac{5}{18} & \frac{4}{9} & \frac{5}{18}
\end{array}$$

to get the Hammer-Hollingsworth-Newton (HHN) and Hammer-Hollingsworth-modified Newton (HHmN) methods, the Kuntzmann-Butcher-Newton (KBN) and Kuntzmann-Butcher-modified Newton (KBmN) methods. To solve problem (3.8) we will apply these methods with trivial predictor and with predictor of order two. The enough number of iterations that guarantees the maximum order for each method can be determined from estimates (3.6b), (3.7b). In this case all the methods reach the maximum orders (four and six, respectively).

In the local-global error control mechanism presented in section 2 we have two free parameters $\epsilon_l$ and $\epsilon_g$. Varying these parameters we can change properties of the local-global stepsize control such as stability, time of computation, accuracy and so on. But the problem of searching the optimal $\epsilon_l$ and $\epsilon_g$ with respect to the properties indicated above will be the topic of future research. Now we have limited ourselves by the case $\epsilon_l = \epsilon_g$. In this case we can compare the standard local stepsize control and the local-global one developed in the paper.

We solve problem (3.8) by each method (HHN, HHmN, KBN, KBmN) with trivial predictor (and nontrivial one) for various values of $\epsilon_g$. Table 1 contains the global errors of the methods with trivial predictor. The similar data for the methods with predictor of

Table 1: Global errors of the methods with trivial predictor and with extrapolation (enough iterations have been fulfilled for every method, $\epsilon_l = \epsilon_g$)

| method | $\epsilon_g$ | | | | |
|---|---|---|---|---|---|
| | $10^{-3}$ | $10^{-6}$ | $10^{-9}$ | $10^{-12}$ | $10^{-15}$ |
| HHN | $1.371 \cdot 10^{-6}$ | $3.366 \cdot 10^{-8}$ | $6.731 \cdot 10^{-14}$ | $3.331 \cdot 10^{-16}$ | $1.422 \cdot 10^{-15}$ |
| HHmN | $1.832 \cdot 10^{-6}$ | $3.388 \cdot 10^{-8}$ | $1.046 \cdot 10^{-12}$ | $5.829 \cdot 10^{-16}$ | $2.852 \cdot 10^{-15}$ |
| KBN | $9.976 \cdot 10^{-8}$ | $1.578 \cdot 10^{-10}$ | $1.250 \cdot 10^{-11}$ | $2.776 \cdot 10^{-16}$ | $2.790 \cdot 10^{-16}$ |
| KBmN | $3.111 \cdot 10^{-7}$ | $1.940 \cdot 10^{-9}$ | $1.370 \cdot 10^{-11}$ | $1.249 \cdot 10^{-15}$ | $1.596 \cdot 10^{-16}$ |

Table 2: Global errors of the methods with predictor of order two and with extrapolation (enough iterations have been fulfilled for every method, $\epsilon_l = \epsilon_g$)

| method | $\epsilon_g$ | | | | |
|---|---|---|---|---|---|
| | $10^{-3}$ | $10^{-6}$ | $10^{-9}$ | $10^{-12}$ | $10^{-15}$ |
| HHN | $1.373 \cdot 10^{-6}$ | $3.366 \cdot 10^{-8}$ | $6.739 \cdot 10^{-14}$ | $2.984 \cdot 10^{-16}$ | $2.873 \cdot 10^{-15}$ |
| HHmN | $1.829 \cdot 10^{-6}$ | $3.343 \cdot 10^{-8}$ | $5.354 \cdot 10^{-13}$ | $7.494 \cdot 10^{-16}$ | $2.699 \cdot 10^{-15}$ |
| KBN | $1.292 \cdot 10^{-7}$ | $1.662 \cdot 10^{-10}$ | $1.256 \cdot 10^{-11}$ | $1.110 \cdot 10^{-16}$ | $1.596 \cdot 10^{-16}$ |
| KBmN | $1.384 \cdot 10^{-7}$ | $1.547 \cdot 10^{-10}$ | $1.256 \cdot 10^{-11}$ | $1.110 \cdot 10^{-16}$ | $1.596 \cdot 10^{-16}$ |

order two are given in table 2. The average stepsizes of all the methods are presented in tables 3 and 4. In addition, we give the pictures of both the global error and the global error estimated by (2.6) for the HHN and KBN methods with nontrivial predictor (see fig. 3 and 4).

From the tables and the pictures we see that the local-global controller (2.6) allows problem (3.8) to be solved with given accuracy[3]. We have the same result also for controller (2.12).

[3]When $\epsilon_g$ is too small ($10^{-15}$) round-off errors influence to the numerical solutions from the HHN and HHmN methods

Table 3: Average stepsizes of the methods with trivial predictor and with extrapolation (enough iterations have been fulfilled for every method, $\epsilon_l = \epsilon_g$)

| method | $\epsilon_g$ | | | | |
|---|---|---|---|---|---|
| | $10^{-3}$ | $10^{-6}$ | $10^{-9}$ | $10^{-12}$ | $10^{-15}$ |
| HHN | $1.797 \cdot 10^{-2}$ | $8.987 \cdot 10^{-3}$ | $3.162 \cdot 10^{-3}$ | $8.559 \cdot 10^{-4}$ | $2.181 \cdot 10^{-4}$ |
| HHmN | $1.797 \cdot 10^{-2}$ | $8.987 \cdot 10^{-3}$ | $3.162 \cdot 10^{-3}$ | $8.559 \cdot 10^{-4}$ | $2.184 \cdot 10^{-4}$ |
| KBN | $2.009 \cdot 10^{-2}$ | $1.626 \cdot 10^{-2}$ | $1.035 \cdot 10^{-2}$ | $5.336 \cdot 10^{-3}$ | $2.175 \cdot 10^{-3}$ |
| KBmN | $2.009 \cdot 10^{-2}$ | $1.626 \cdot 10^{-2}$ | $1.035 \cdot 10^{-2}$ | $5.336 \cdot 10^{-3}$ | $2.175 \cdot 10^{-3}$ |

Table 4: Average stepsizes of the methods with predictor of order two and with extrapolation (enough iterations have been fulfilled for every method, $\epsilon_l = \epsilon_g$)

| method | $\epsilon_g$ | | | | |
|---|---|---|---|---|---|
| | $10^{-3}$ | $10^{-6}$ | $10^{-9}$ | $10^{-12}$ | $10^{-15}$ |
| HHN | $1.797 \cdot 10^{-2}$ | $8.987 \cdot 10^{-3}$ | $3.162 \cdot 10^{-3}$ | $8.559 \cdot 10^{-4}$ | $2.184 \cdot 10^{-4}$ |
| HHmN | $1.797 \cdot 10^{-2}$ | $8.987 \cdot 10^{-3}$ | $3.162 \cdot 10^{-3}$ | $8.559 \cdot 10^{-4}$ | $2.184 \cdot 10^{-4}$ |
| KBN | $2.009 \cdot 10^{-2}$ | $1.626 \cdot 10^{-2}$ | $1.035 \cdot 10^{-2}$ | $5.336 \cdot 10^{-3}$ | $2.175 \cdot 10^{-3}$ |
| KBmN | $2.009 \cdot 10^{-2}$ | $1.626 \cdot 10^{-2}$ | $1.035 \cdot 10^{-2}$ | $5.336 \cdot 10^{-3}$ | $2.175 \cdot 10^{-3}$ |

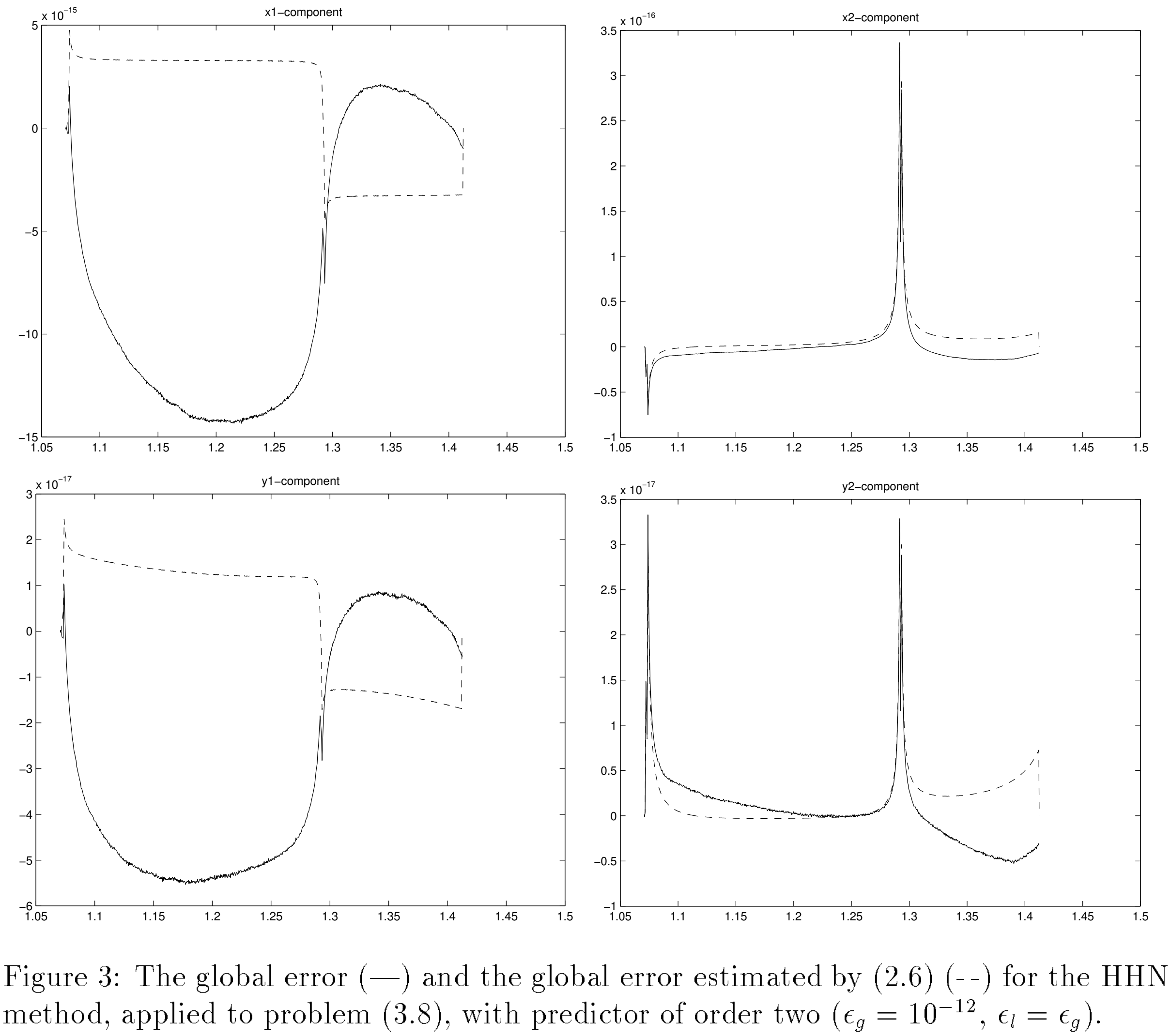


Figure 3: The global error (—) and the global error estimated by (2.6) (- -) for the HHN method, applied to problem (3.8), with predictor of order two ($\epsilon_g = 10^{-12}$, $\epsilon_l = \epsilon_g$).

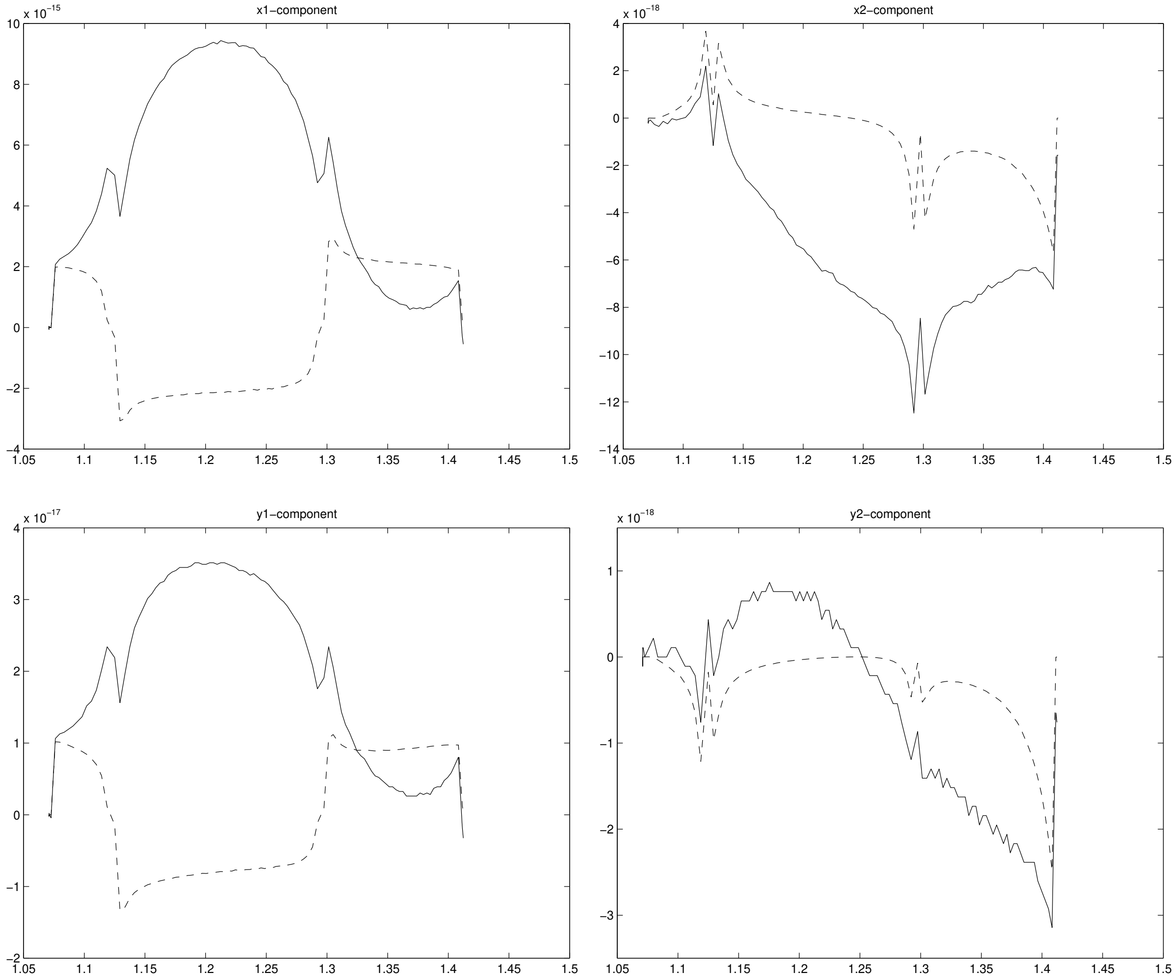


Figure 4: The global error (—) and the global error estimated by (2.6) (- -) for the KBN method, applied to problem (3.8), with predictor of order two ($\epsilon_g = 10^{-12}$, $\epsilon_l = \epsilon_g$).

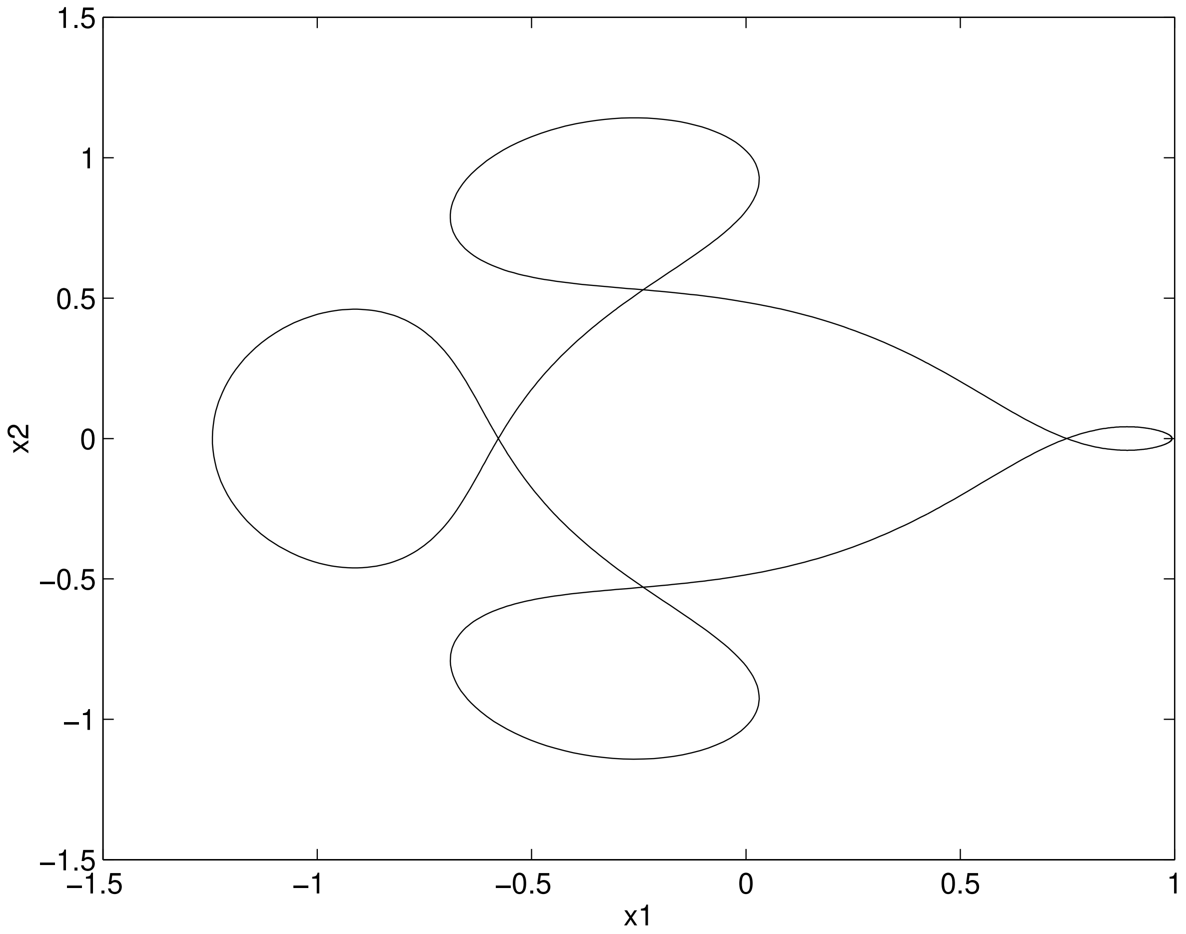


Figure 5: The solution of problem (4.1) obtained by the variable-stepsize KBN method with the stable implicit Euler local-global stepsize control.

# 4 Stiff DAEs

In this section we study how the local-global stepsize control works for stiff semi-explicit differential-algebraic equations. Further we will call the local-global stepsize control based on formulas (2.6) as *the implicit Euler local-global stepsize control*, and the notation *the trapezoidal rule local-global stepsize control* denotes the local-global controller based on formulas (2.12).

As a good test problem, we take the restricted three body problem in differential-algebraic formulation

$$x1'(t) = x3(t), \tag{4.1a}$$

$$x2'(t) = x4(t), \tag{4.1b}$$

$$x3'(t) = x1(t) + 2 \cdot x4(t) - \mu_1 \frac{x1(t) + \mu_2}{y1(t)} - \mu_2 \frac{x1(t) - \mu_1}{y2(t)}, \tag{4.1c}$$

Table 5: Global errors of the variable-stepsize KBN method with the local stepsize control and with extrapolation

| N | $\epsilon_l$ | | | | |
|---|---|---|---|---|---|
| | $10^{-3}$ | $10^{-5}$ | $10^{-7}$ | $10^{-9}$ | $10^{-11}$ |
| 2 | $2.925 \cdot 10^{-3}$ | $9.381 \cdot 10^{-6}$ | $2.807 \cdot 10^{-7}$ | $3.034 \cdot 10^{-10}$ | $1.165 \cdot 10^{-11}$ |
| 3 | $2.952 \cdot 10^{-3}$ | $9.545 \cdot 10^{-6}$ | $2.241 \cdot 10^{-7}$ | $2.813 \cdot 10^{-10}$ | $1.171 \cdot 10^{-11}$ |
| 4 | $2.948 \cdot 10^{-3}$ | $9.545 \cdot 10^{-6}$ | $2.241 \cdot 10^{-7}$ | $2.812 \cdot 10^{-10}$ | $1.159 \cdot 10^{-11}$ |
| 5 | $2.940 \cdot 10^{-3}$ | $9.545 \cdot 10^{-6}$ | $2.241 \cdot 10^{-7}$ | $2.812 \cdot 10^{-10}$ | $1.158 \cdot 10^{-11}$ |

Table 6: Global errors of the variable-stepsize KBN method with the trapezoidal rule local-global stepsize control and with extrapolation ($\epsilon_l = \epsilon_g$)

| N | $\epsilon_g$ | | | | |
|---|---|---|---|---|---|
| | $10^{-3}$ | $10^{-5}$ | $10^{-7}$ | $10^{-9}$ | $10^{-11}$ |
| 2 | $5.834 \cdot 10^{-4}$ | $3.218 \cdot 10^{-5}$ | $4.009 \cdot 10^{-8}$ | $6.395 \cdot 10^{-8}$ | $9.124 \cdot 10^{-12}$ |
| 3 | $2.004 \cdot 10^{-4}$ | $3.228 \cdot 10^{-5}$ | $5.435 \cdot 10^{-8}$ | $6.392 \cdot 10^{-8}$ | $9.450 \cdot 10^{-12}$ |
| 4 | $1.489 \cdot 10^{-4}$ | $3.228 \cdot 10^{-5}$ | $5.435 \cdot 10^{-8}$ | $6.392 \cdot 10^{-8}$ | $9.339 \cdot 10^{-12}$ |
| 5 | $3.635 \cdot 10^{-4}$ | $3.228 \cdot 10^{-5}$ | $5.435 \cdot 10^{-8}$ | $6.392 \cdot 10^{-8}$ | $9.337 \cdot 10^{-12}$ |

$$x4'(t) = x2(t) - 2 \cdot x3(t) - \mu_1 \frac{x2(t)}{y1(t)} - \mu_2 \frac{x2(t)}{y2(t)}, \tag{4.1d}$$

$$y1(t) = \left(2 \cdot x1(t) + y2(t)^{2/3} + 2\mu_2 - 1\right)^{3/2}, \tag{4.1e}$$

$$y2(t) = \left((x1(t) - \mu_1)^2 + x2(t)^2\right)^{3/2}, \tag{4.1f}$$

where $t \in [0, T]$, $T = 17.0652165601579625588\,91$, and where $\mu_1 = 1 - \mu_2$, $\mu_2 = 0.012277471$. The initial values of problem (4.1) are: $x1(0) = 0.994$, $x2(0) = 0$, $x3(0) = 0$, $x4(0) = -2.00158510637908252240$, $y1(0) = ((x1(0) + \mu_2)^2 + x2(0)^2)^{3/2}$, $y2(0) = ((x1(0) - \mu_1)^2 + x2(0)^2)^{3/2}$. The solution of problem (4.1) is given in fig. 5, and it is smooth except two singular points: $x1 = \mu_1$ and $x1 = -\mu_2$, $x2 = 0$.

Let us now solve problem (4.1) by the variable-stepsize KBN method with predictor of order two. In table 5 we give global errors of the method with the standard local stepsize control. The global errors have been evaluated at the point $T$. Table 6 contains similar data for the method with the trapezoidal rule local-global stepsize control. We have taken the exacter local-global controller to show that both the local control and the local-global one are not appropriate for solving stiff DAEs with guaranteed error. Indeed, in table 5 the global error exceeds the limit when $\epsilon_l$ is equal to $10^{-3}$, $10^{-7}$ and $10^{-11}$. The same situation occurs in table 6 when $\epsilon_g$ equals $10^{-5}$ and $10^{-9}$. What happened? Of course, the local stepsize control cannot keep the global error bounded relative to the given limit. But why the local-global error control mechanism does not work?

To answer this question we must remember that the Jacobian of the right-hand part of stiff DAEs has large absolute eigenvalues at least in some intervals of the segment of

Table 7: Global errors of the variable-stepsize KBN method with the stable implicit Euler local-global stepsize control and with extrapolation ($\epsilon_l = \epsilon_g$ and $\epsilon_{gl} = \epsilon_g/100$)

| N | $\epsilon_g$ | | | | |
|---|---|---|---|---|---|
| | $10^{-3}$ | $10^{-5}$ | $10^{-7}$ | $10^{-9}$ | $10^{-11}$ |
| 2 | $2.001 \cdot 10^{-4}$ | $9.462 \cdot 10^{-5}$ | $7.926 \cdot 10^{-7}$ | $9.633 \cdot 10^{-8}$ | $4.327 \cdot 10^{-10}$ |
| 3 | $1.998 \cdot 10^{-4}$ | $4.890 \cdot 10^{-6}$ | $1.453 \cdot 10^{-8}$ | $2.814 \cdot 10^{-10}$ | $2.846 \cdot 10^{-13}$ |
| 4 | $1.998 \cdot 10^{-4}$ | $4.890 \cdot 10^{-6}$ | $1.266 \cdot 10^{-8}$ | $2.623 \cdot 10^{-10}$ | $7.889 \cdot 10^{-13}$ |
| 5 | $1.998 \cdot 10^{-4}$ | $4.890 \cdot 10^{-6}$ | $1.266 \cdot 10^{-8}$ | $2.622 \cdot 10^{-10}$ | $6.856 \cdot 10^{-13}$ |

integration. Therefore in these intervals problem (2.5) is very stiff because of the Jacobian and the local error $\tilde{\psi}_{s+1}(t)$, which is a rapidly changing function. In such situation, when solving problem (2.5) by method (2.6) (or (2.12)), we may make an error exceeding the global error itself. The reason is the stepsize $\tau_k$ that may be too large to solve (2.5) accurately enough. So, it is important to keep also the local error of (2.6) (or (2.12)) bounded in order to increase the accuracy of the local-global stepsize control and to make it more stable for stiff problems.

We can easily do this using formulas (2.6) and (2.12). These formulas give the solutions of (2.5) with different order errors. Then, by computing the difference between these solutions, as in [13], we can find the components of the local error of controller (2.6) ($\triangle\tilde{\psi}_s(t_k)$ and $\triangle\tilde{\phi}_s(t_k)$). Thus, we choose the bound $\epsilon_{gl}$ for this error and obtain *the stable implicit Euler local-global stepsize control* (see fig. 6).

Let us test the new controller on problem (4.1). As above, in table 7 we give global errors of the KBN method with the stable implicit Euler local-global stepsize control for the same parameters $N$ ang $\epsilon_g$ as in tables 5 and 6. We have also taken $\epsilon_{gl} = \epsilon_g/100$ there. Having compared the results from tables 5–7, we see that, in contrast to the previous ones, the new controller is more accurate and stable, and it is really appropriate for stiff semi-explicit differential-algebraic equations. In fig. 7 we can observe the behaviour of the controller for $\epsilon_g = 10^{-11}$.

Finally, to point out once again the nice properties of the stable controller even for lower order methods we apply trapezoidal rule to the semi-explicit differential-algebraic equations (2.1) and construct the TRN method of order two [9]. From the results of table 8 it follows that the variable-stepsize TRN method with the stable implicit Euler local-global error control mechanism also gives any accuracy preassigned for the numerical solution of problem (4.1) if the number of iterations $N$ is sufficiently large.

# 5 A semi-explicit local-global stepsize control

In the preceding sections we presented and examined some global error controllers for nonstiff and stiff semi-explicit differential-algebraic equations. All of them were based on an application of implicit one-step methods to problem (2.5). As a result, we obtained the numerical solution from the variable-stepsize RK methods with any desired accuracy

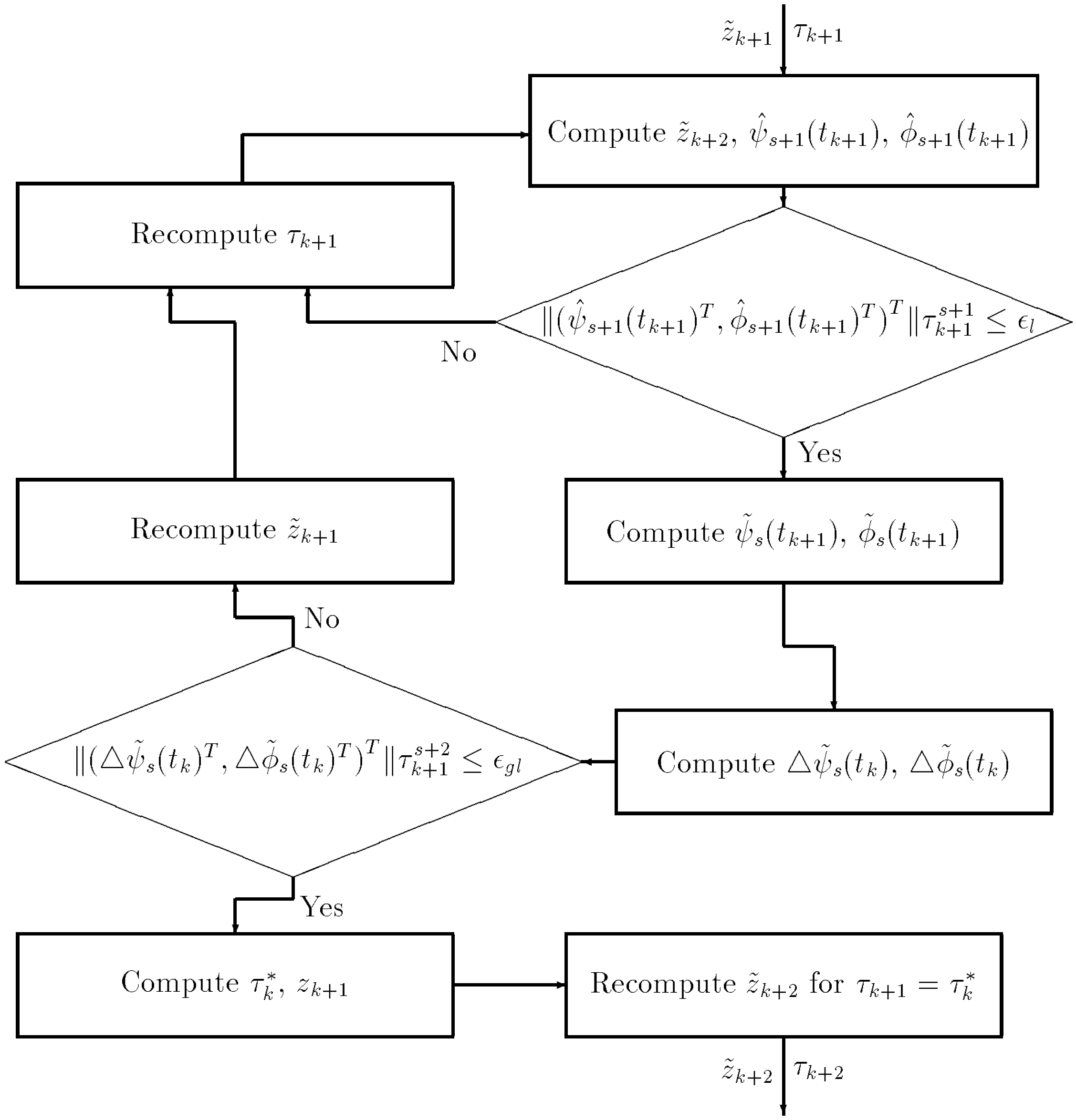


Figure 6: The stable implicit Euler local-global stepsize control procedure.

Table 8: Global errors of the variable-stepsize TRN method with the stable implicit Euler local-global stepsize control and with extrapolation ($\epsilon_l = \epsilon_g$ and $\epsilon_{gl} = \epsilon_g/100$)

| N | $\epsilon_g$ | | | | |
|---|---|---|---|---|---|
| | $10^{-3}$ | $10^{-5}$ | $10^{-7}$ | $10^{-9}$ | $10^{-11}$ |
| 2 | $1.529 \cdot 10^{-2}$ | $2.290 \cdot 10^{-4}$ | $6.305 \cdot 10^{-7}$ | $2.061 \cdot 10^{-10}$ | $3.398 \cdot 10^{-12}$ |
| 3 | $6.518 \cdot 10^{-4}$ | $8.061 \cdot 10^{-6}$ | $4.734 \cdot 10^{-8}$ | $2.148 \cdot 10^{-10}$ | $8.400 \cdot 10^{-13}$ |
| 4 | $6.518 \cdot 10^{-4}$ | $8.061 \cdot 10^{-6}$ | $4.734 \cdot 10^{-8}$ | $2.143 \cdot 10^{-10}$ | $1.003 \cdot 10^{-12}$ |
| 5 | $6.518 \cdot 10^{-4}$ | $8.061 \cdot 10^{-6}$ | $4.734 \cdot 10^{-8}$ | $2.140 \cdot 10^{-10}$ | $2.387 \cdot 10^{-12}$ |

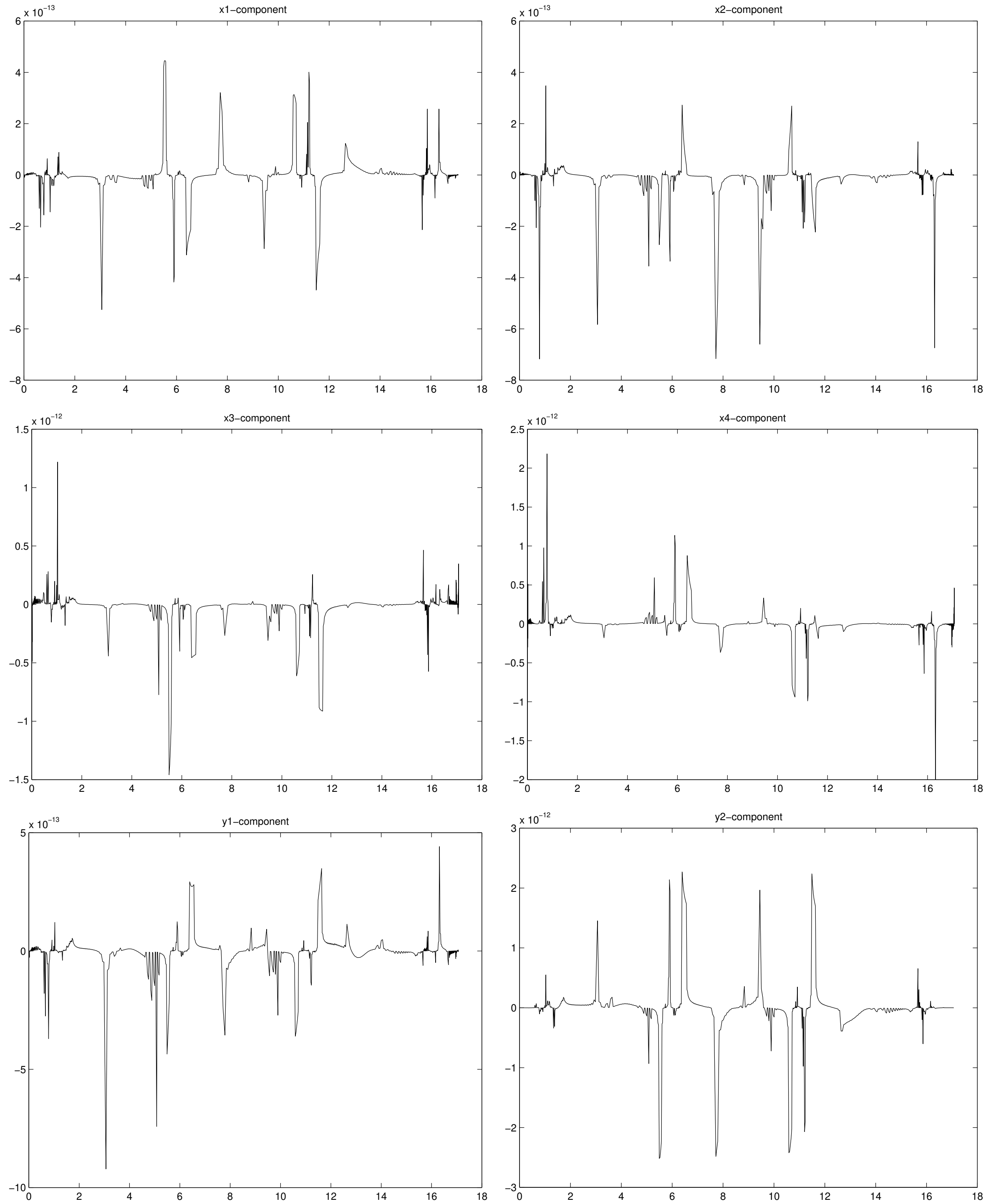


Figure 7: The estimated global error of the variable-stepsize KBN method, applied to problem (4.1), with the stable implicit Euler local-global stepsize control ($N = 3$, $\epsilon_g = 10^{-11}$, $\epsilon_l = \epsilon_g$, $\epsilon_{gl} = \epsilon_g/100$).

up to round-off. But the only drawback is the high cost of the computation of the local error of method (2.2) at the point $t_{k+1}$. Moreover, we have to recompute $\tilde{z}_{k+2}$ during the numerical integration and the local error every time the stable local-global stepsize control requires it. Thus, we need to simplify the error control mechanism for more effective practical implementation.

Since most of the extra computations connects with the need to know the local error $\check{\psi}_{s+1}(t)$ at the point $t_{k+1}$ we replace this error with the local error evaluated at the point $t_k$ in (2.6) that reduces the computing time significantly:

$$\text{(5.1}a) \qquad \begin{pmatrix} \tilde{\psi}_s(t_{k+1}) \\ \tilde{\phi}_s(t_{k+1}) \end{pmatrix} = P_{k+1}^{-1} \begin{pmatrix} \tilde{\psi}_s(t_k) + \tau_k \check{\psi}_{s+1}(t_k) \\ 0 \end{pmatrix},$$

$$\text{(5.1}b) \qquad \tilde{\psi}_s(0) = 0, \quad \tilde{\phi}_s(0) = 0, \quad k = 0, 1, \dots, K-1,$$

where the matrix $P_{k+1}^{-1}$ is the same as that in formulas (2.6). We can do such replacement because of $\check{\psi}_{s+1}(t_{k+1}) = \check{\psi}_{s+1}(t_k) + O(\tau_k)$ (this follows from condition I). Hence, we have changed formula (2.6a) by a value $O(\tau_k^2)$ that does not affect to $\tilde{\psi}_s(t_{k+1})$ and $\tilde{\phi}_s(t_{k+1})$ if the stepsize $\tau_k$ is sufficiently small. We call formulas (5.1) as *the semi-explicit Euler local-global stepsize control*.

Earlier we showed that we have to control the local error of the numerical solution of (2.5) to improve the local-global stepsize control for stiff semi-explicit DAEs. In section 4 we used one more controller with a different order error to compute the local error of (2.6). The same idea is not appropriate for (5.1) because a higher order controller requires additional computations of $\check{\psi}_{s+1}(t)$ in the interval $(t_k, t_{k+1}]$, and the best combination is the pair (2.6) and (2.12). Therefore we use the Richardson local extrapolation to evaluate the error of (5.1). It means that the extrapolation will be also preferable for the computation of the local error of method (2.2) because we have to compute the solution of (2.2) at the middle point of the interval $[t_k, t_{k+1}]$, anyway. Thus, we derive *the stable semi-explicit Euler local-global stepsize control* (see fig. 8).

Let us now test the stable semi-explicit controller on the restricted three body problem. In table 9 we present global errors of the variable-stepsize KBN method for the same parameters as in table 7. We see from these tables that the new controller works as well as the implicit one. In addition, pictures 7 and 9 show similar dynamical behaviour of these controllers. But the computing time was twice less for the stable semi-explicit Euler local-global stepsize controller. Moreover, this time was only a little bit more than the computing time for the standard local stepsize control.

# 6 Conclusions

This paper is probably a first attempt to use the truncated global error of Runge-Kutta methods in the error control mechanism. The advantage of this approach is ability to keep the global error of a Runge-Kutta formula as small as we want. Therefore a user

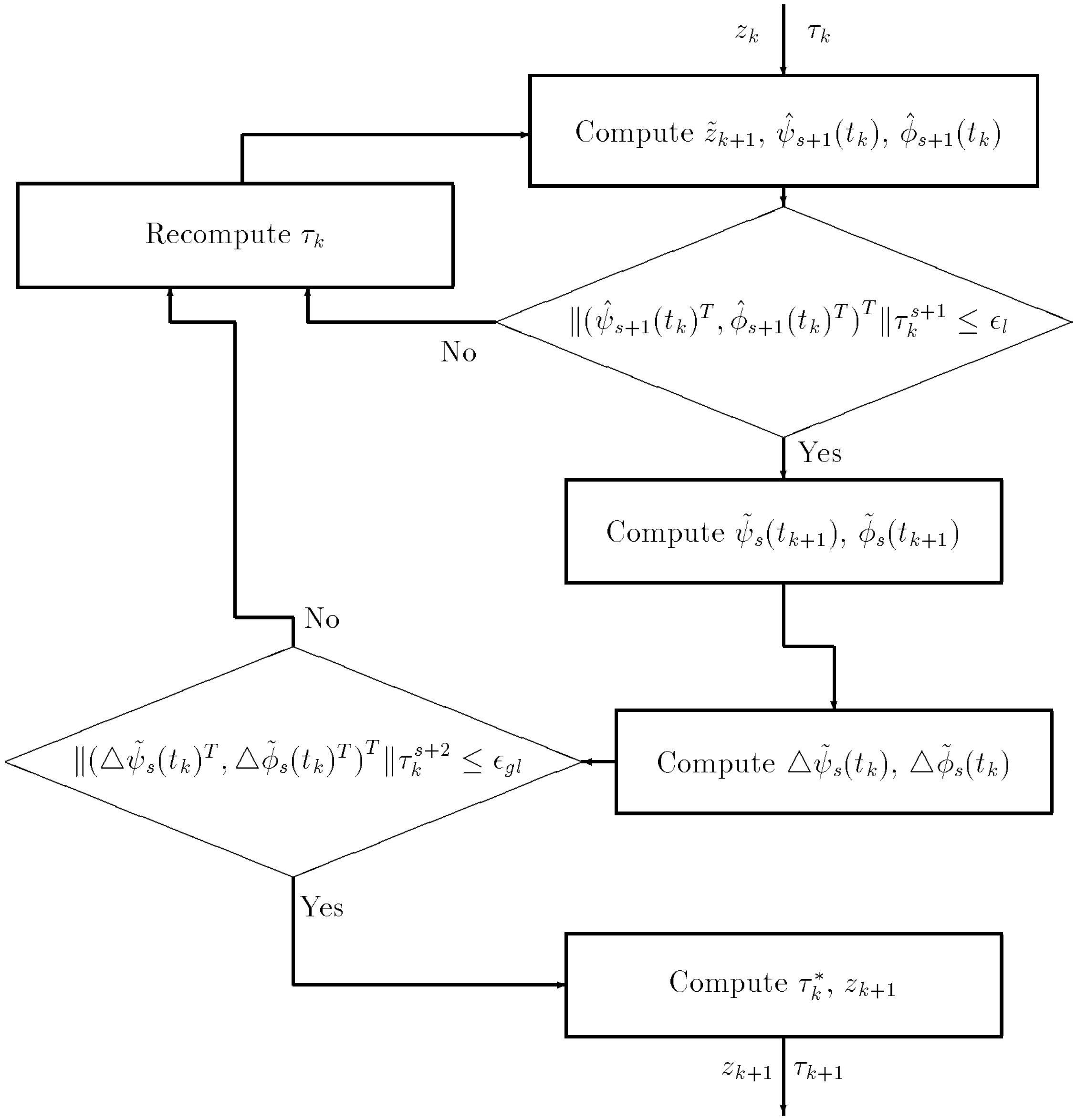


Figure 8: The stable semi-explicit Euler local-global stepsize control procedure.

Table 9: Global errors of the variable-stepsize KBN method with the stable semi-explicit Euler local-global stepsize control and with extrapolation ($\epsilon_l = \epsilon_g$ and $\epsilon_{gl} = \epsilon_g/100$)

| N | $\epsilon_g$ | | | | |
|---|---|---|---|---|---|
| | $10^{-3}$ | $10^{-5}$ | $10^{-7}$ | $10^{-9}$ | $10^{-11}$ |
| 2 | $2.032 \cdot 10^{-4}$ | $1.248 \cdot 10^{-6}$ | $2.543 \cdot 10^{-8}$ | $3.472 \cdot 10^{-11}$ | $1.493 \cdot 10^{-12}$ |
| 3 | $2.008 \cdot 10^{-4}$ | $1.372 \cdot 10^{-6}$ | $3.253 \cdot 10^{-8}$ | $3.809 \cdot 10^{-11}$ | $1.193 \cdot 10^{-12}$ |
| 4 | $2.008 \cdot 10^{-4}$ | $1.372 \cdot 10^{-6}$ | $3.253 \cdot 10^{-8}$ | $3.814 \cdot 10^{-11}$ | $1.308 \cdot 10^{-12}$ |
| 5 | $2.007 \cdot 10^{-4}$ | $1.372 \cdot 10^{-6}$ | $3.253 \cdot 10^{-8}$ | $3.812 \cdot 10^{-11}$ | $1.304 \cdot 10^{-12}$ |

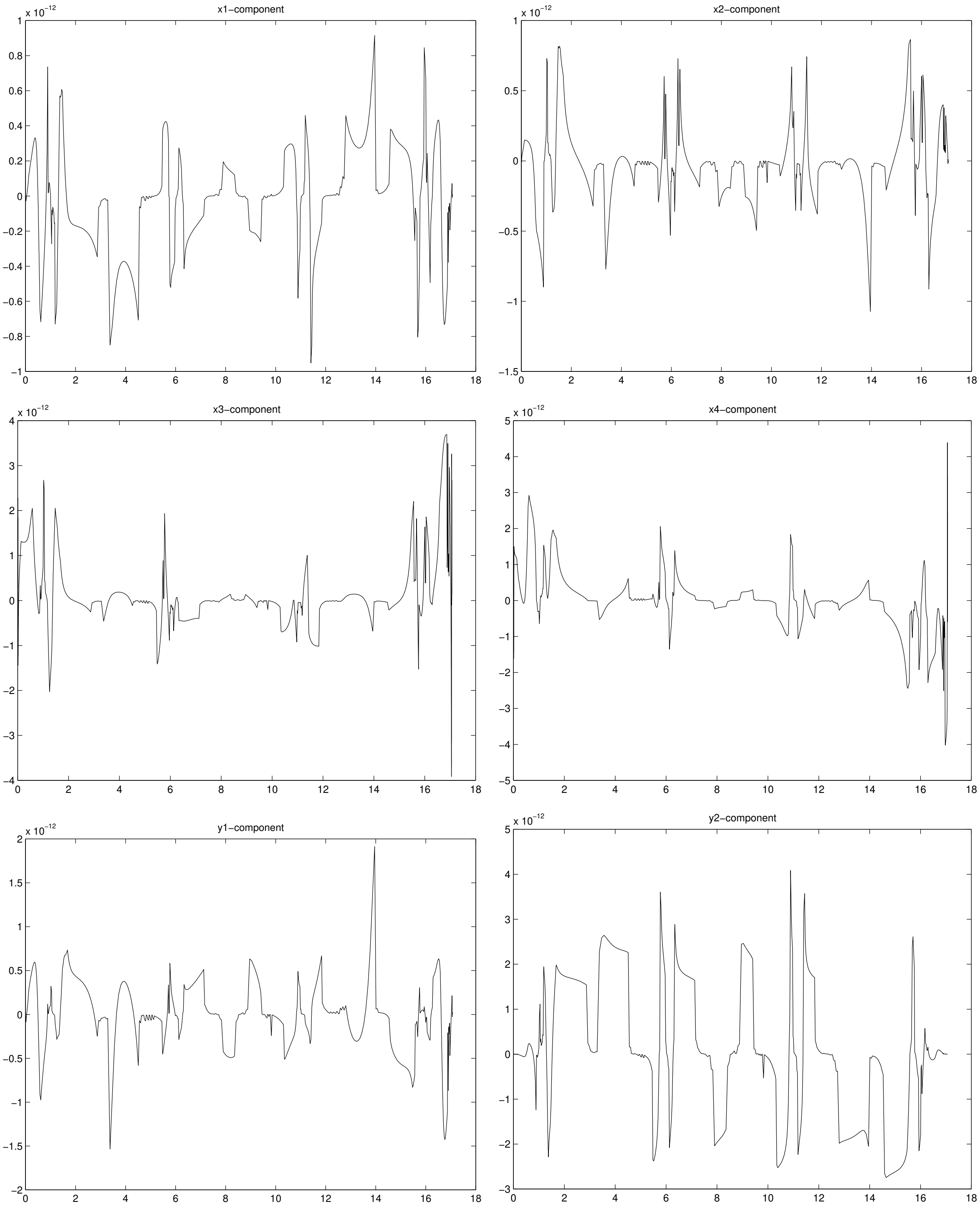


Figure 9: The estimated global error of the variable-stepsize KBN method, applied to problem (4.1), with the stable semi-explicit Euler local-global stepsize control ($N = 3$, $\epsilon_g = 10^{-11}$, $\epsilon_l = \epsilon_g$, $\epsilon_{gl} = \epsilon_g/100$).

may know nothing about the problem and the method in order to solve the problem with necessary accuracy (up to round-off errors).

In section 3 we have shown that the local-global stepsize control presented in the paper can be also used in numerical integration of nonlinear DAEs by implicit Runge-Kutta methods if the number of the full (or modified) Newton iterations is sufficiently large. Moreover, taking into account that we can use the same Jacobian both in the local-global stepsize control procedure and in iterative methods (3.1) and (3.2), we can reduce computing time dramatically. Thus, for implicit methods the cost of the local-global error control mechanism is not much higher than the cost of the standard local stepsize control. In addition, by using the semi-explicit controller from section 5, we can implement the local-global stepsize control in practice very efficiently.

In sections 4 and 5 we presented the stable implicit and semi-explicit controllers, respectively. It was shown that such controllers keep the global error bounded also for stiff semi-explicit DAEs. The advantage of the stable local-global stepsize control over the local one is that we compute and control three values of different orders with respect to the stepsize $\tau_k$. We make the next step only if all the values are sufficiently small. Thus, the stable local-global stepsize control chooses stepsize more carefully than the local one. This increases reliability of the error control mechanism.

Finally, we note that all theoretical results of the paper are also valid in the class of ordinary differential equations. To construct the local-global stepsize control for differential equations we must only eliminate the variable $y$ and algebraic equation (2.1b) from all formulas in the paper. The only difference is less number of iterations $N$ that is enough to guarantee the maximum order for iterative methods (3.1) and (3.2) applied to ordinary differential equations. So, we can easily see that the inequality

$$N \geq \log_2\Bigl(\frac{s+3}{\xi+1}\Bigr)$$

gives the sufficient number of iterations for the RKN method. The similar estimate of the number of iterations for the RKmN method is

$$N \geq \frac{s+3}{\xi+1} - 1.$$

# 7 Acknowledgments

The author is very grateful to Innokenti V. Semoushin and George F. Corliss for their comments concerning this paper. Especially, the author would like to thank Nikolai S. Bakhvalov, Syvert P. Nørsett, Anne Kværnø and many others for discussions of the results of the paper.

# References

[1] C. Arévalo, G. Söderlind, *Convergence of multistep discretizations of DAEs*, BIT, 35 (1995), pp. 143–168.

[2] E. Eich, C. Führer, J. Yen, *On the error control for multistep methods applied to ODEs with invariants and DAEs in multibody dynamics*, Mech. Struct. and Mach., 23(2) (1995), pp. 159–179.

[3] C.W. Gear, L.R. Petzold, *ODE methods for the solution of differential/algebraic systems*, SIAM J. Numer. Anal., 21 (1984), pp. 716–728.

[4] E. Hairer, S.P. Nørsett, G. Wanner, *Solving ordinary differential equations I: Nonstiff problems*, Springer-Verlag, Berlin, 1987, 1993.

[5] E. Hairer, Ch. Lubich, M. Roche, *The numerical solution of differential-algebraic systems by Runge-Kutta methods*, Lecture Note in Math. 1409, Springer-Verlag, Berlin, 1989.

[6] E. Hairer, G. Wanner, *Solving ordinary differential equations II: Stiff and differential-algebraic problems*, Springer-Verlag, Berlin, 1991.

[7] K.R. Jackson, A. Kværnø, S.P. Nørsett, *The use of Butcher series in the analysis of Newton-like iterations in Runge-Kutta formulas*, Applied Numerical Mathematics, 15 (1994), pp. 341–356.

[8] G.Yu. Kulikov, *A method for the numerical solution of the autonomous Cauchy problem with an algebraic relation between the phase variables*, Vestn. MGU Ser. Mat. Mekh., (1992), No. 1, pp. 14–19.

[9] G.Yu. Kulikov, *The numerical solution of the autonomous Cauchy problem with an algebraic relation between the phase variables (non-degenerate case)*, Vestn. MGU Ser. Mat. Mekh., (1993), No. 3, pp. 6–10.

[10] G.Yu. Kulikov, *The numerical solution of the autonomous Cauchy problem with an algebraic relation between the phase variables*, Zh. Vychisl. Mat. Mat. Fiz., 33 (1993), No. 4, pp. 522–540.

[11] G.Yu. Kulikov, *Modern theory of symmetric one-step methods for ODEs*, Technical report 13/1996, IMM, Technical University of Denmark, Lyngby, 1996.

[12] G.Yu. Kulikov, *Convergence theorems for iterative Runge-Kutta methods with fixed step size*, Zh. Vychisl. Mat. Mat. Fiz., 36 (1996), No. 8, pp. 73–89.

[13] G.Yu. Kulikov, P.G. Thomsen, *Convergence and implementation of implicit Runge-Kutta methods for DAEs*, Technical report 7/1996, IMM, Technical University of Denmark, Lyngby, 1996.

[14] G.Yu. Kulikov, *A theory of symmetric one-step methods for differential-algebraic equations*, Russian J. Numer. Anal. Math. Modelling, *to appear in 1997*.

[15] G.Yu. Kulikov, *Numerical methods solving the semi-explicit differential-algebraic equations by implicit multistep fixed stepsize methods*, Korean J. Comput. & Appl. Math., 4 (1997), No. 2, pp. 281–318.

[16] G.Yu. Kulikov, *Numerical solving the Cauchy problem for a system of differential-algebraic equations by implicit Runge-Kutta methods with nontrivial predictor*, Zh. Vychisl. Mat. Mat. Fiz., *to appear in 1998.*

[17] A. Kværnø, *The order of Runge-Kutta methods applied to semi-explicit DAEs of index 1, using Newton-type iterations to compute the internal stage values*, Technical report 2/1992, Mathematical Sciences Div., Norwegian Institute of Technology, Trondheim, 1992.