

# On a Family of A-stable Collocation Methods with High Derivatives

Gennady Y. Kulikov<sup>1</sup>, Arkadi I. Merkulov<sup>2</sup>, and Ekaterina Y. Khrustaleva<sup>2</sup>

<sup>1</sup> School of Computational and Applied Mathematics, University of the Witwatersrand, Private Bag 3, Wits 2050, Johannesburg, South Africa  
gkulikov@cam.wits.ac.za

<sup>2</sup> Ulyanovsk State University, L. Tolstoy Str. 42, 432970 Ulyanovsk, Russia  
merkul@vda.ru, shabalkina@mail.ru

**Abstract.** In this paper we develop a family of A-stable one-step methods with high derivatives by means of a collocation technique. We present construction details and a theory to justify such sort of methods. We also concentrate on an effective way of their practical implementation.

## 1 Introduction

In this paper we study one-step numerical methods for ordinary differential equations (ODEs) of the form

$$x'(t) = g(t, x(t)), \quad t \in [t_0, t_0 + T], \quad x(t_0) = x^0 \quad (1)$$

where  $x(t) \in \mathbf{R}^n$  and  $g: D \subset \mathbf{R}^{n+1} \rightarrow \mathbf{R}^n$  is a sufficiently smooth function. Problem (1) is quite usual in theoretical research and in practical engineering (see, for example, [2], [4], [6], [7], [12]). Therefore efficient methods for finding its solution with any set accuracy (up to round-off errors) are important as in theory as in practice.

Here, we present a family of one-step collocation methods with high derivatives. Note that such topic of research proved its efficiency. So, there are many papers in this field (see [5], [7], [8], [11] and so on). Especially, we pay attention to the general linear methods developed by Butcher [4] which also include high derivatives. In general, all that methods are implicit. Therefore the most difficult task is their correct and effective implementation because we have to involve an additional iterative scheme.

The outline of this paper is organized as follows: We present the family of one-step collocation methods with high derivatives in Sect. 2. There, we also study convergence and A-stability of these methods. An effective implementation based on Newton-like iterations is given in Sect. 3. The last section of the paper is devoted to the problem of step size selection. Theoretical results are clearly confirmed by numerical experiments.

## 2 Collocation Methods with High Derivatives

As in [1], we use the collocation technique to construct numerical methods with high derivatives. First, we fix the points  $t_k$ ,  $t_{k+1/2}$  and  $t_{k+1}$  on an arbitrary

interval  $[t_k, t_{k+1}]$  as collocation ones. Second, we assume that we know derivatives of the right-hand part of problem (1) at the nodes  $t_k$  and  $t_{k+1}$  up to order  $p$ . Now we can construct the Hermit interpolation formula (see, for example, [3]) and substitute it into the right-hand part of system (1) instead of  $g(t, x(t))$ . Integrating the newly obtained system on the interval  $[t_k, t_{k+1}]$  we come to the numerical method

$$x_{k+1/2} = x_k + \tau_k \sum_{r=0}^p \tau_k^r \left( a_1^{(r)} g_k^{(r)} + a_3^{(r)} g_{k+1}^{(r)} \right) + \tau_k a_2 g_{k+1/2}^{(0)}, \quad (2a)$$

$$x_{k+1} = x_k + \tau_k \sum_{r=0}^p \tau_k^r \left( b_1^{(r)} g_k^{(r)} + b_3^{(r)} g_{k+1}^{(r)} \right) + \tau_k b_2 g_{k+1/2}^{(0)}, \quad (2b)$$

$k = 0, 1, \dots, K-1$ , where

$$a_j^{(r)} = \sum_{i=0}^{p-r} \frac{1}{i!r!} \left[ \frac{(\theta - (j-1)/2)^{p+1}}{\Psi(\theta)} \right]_{\theta=(j-1)/2}^{(i)} \int_0^{1/2} \frac{\Psi(\theta)}{(\theta - (j-1)/2)^{p-i-r+1}} d\theta,$$

$$b_j^{(r)} = \sum_{i=0}^{p-r} \frac{1}{i!r!} \left[ \frac{(\theta - (j-1)/2)^{p+1}}{\Psi(\theta)} \right]_{\theta=(j-1)/2}^{(i)} \int_0^1 \frac{\Psi(\theta)}{(\theta - (j-1)/2)^{p-i-r+1}} d\theta$$

when  $j = 1, 3$  and  $r = 0, 1, \dots, p$ ,

$$a_2 = \frac{(-1)^{p+1}}{4^{p+1}} \int_0^{1/2} \theta^{p+1} (\theta - 1)^{p+1} d\theta, \quad b_2 = \frac{(-1)^{p+1}}{4^{p+1}} \int_0^1 \theta^{p+1} (\theta - 1)^{p+1} d\theta,$$

$x_0 = x^0$ ,  $\theta = (t - t_k)/\tau_k$ ,  $\Psi(\theta) = \theta^{p+1}(\theta - 1/2)(\theta - 1)^{p+1}$ ,  $g_k^{(r)}$  denotes the  $r$ -th derivative<sup>1</sup> of the right-hand part of problem (1) with respect to  $t$  evaluated at the point  $t_k$ , and  $\tau_k$  is a step size which may be fixed or variable.

The results presented above give a way to derive

**Theorem 1** *Let the right-hand part of ODE (1) possess continuous derivatives up to order  $2p+5$  in a neighborhood of the solution  $x(t)$  on the interval  $[t_0, t_0+T]$ , where  $p$  is a nonnegative integer number. Then method (2) is convergent, it has stage order  $2p+3$  and classical order  $2p+4$ , and its coefficients satisfy*

$$a_1^{(r)} = \frac{p+1}{r!2^{p+r+2}} \sum_{i=0}^{p-r} \sum_{l=0}^{i+r} \sum_{j=0}^{p+1} \sum_{q=0}^i \frac{(-1)^l (i+r)!(p+q)!}{l!(i+r-l)!j!(p+1-j)!(l+j+2)q!2^q}, \quad (3a)$$

$$a_2 = \frac{(p+1)!}{2} \sum_{l=0}^{p+1} \frac{(-1)^l}{l!(p+1-l)!(2l+1)}, \quad (3b)$$

---

<sup>1</sup> Here and further the zero-derivative implies the original function.

$$a_3^{(r)} = \frac{(-1)^{r+1}(p+1)}{r!2^{p+r+2}} \sum_{i=0}^{p-r} \sum_{l=0}^{i+r} \sum_{j=0}^{p+1} \frac{(-1)^j(i+r)!}{l!(i+r-l)!j!(p+1-j)!(l+j+2)} \times \sum_{q=0}^i \frac{(p+q)!}{q!2^q}, \quad r = 0, 1, \dots, p, \quad (3c)$$

$$b_1^{(r)} = a_1^{(r)} + (-1)^r a_3^{(r)}, \quad b_2 = 2a_2, \quad b_3^{(r)} = (-1)^r a_1^{(r)} + a_3^{(r)}. \quad (3d)$$

**Theorem 2** *One-step method (2) is A-stable for any integer  $p \geq 0$ .*

Further we call methods of the form (2) as *E-methods with high derivatives*, for short. As an example of E-methods, we present the following numerical method of order 8 obtained by Theorem 1 when  $p = 2$ :

$$x_{k+1/2} = x_k + \frac{689}{2240} \tau_k g_k^{(0)} + \frac{169}{4480} \tau_k^2 g_k^{(1)} + \frac{17}{8960} \tau_k^3 g_k^{(2)} - \frac{81}{2240} \tau_k g_{k+1}^{(0)} + \frac{41}{4480} \tau_k^2 g_{k+1}^{(1)} - \frac{19}{26880} \tau_k^3 g_{k+1}^{(2)} + \frac{8}{35} \tau_k g_{k+1/2}^{(0)}, \quad (4a)$$

$$x_{k+1} = x_k + \frac{57}{210} \tau_k \left( g_k^{(0)} + g_{k+1}^{(0)} \right) + \frac{1}{35} \tau_k^2 \left( g_k^{(1)} - g_{k+1}^{(1)} \right) + \frac{1}{840} \tau_k^3 \left( g_k^{(2)} + g_{k+1}^{(2)} \right) + \frac{16}{35} \tau_k g_{k+1/2}^{(0)}. \quad (4b)$$

### 3 Implementation of E-methods with High Derivatives

When implementing method (2) in practice, we have to involve some additional iterative scheme to treat the corresponding nonlinear algebraic system. We applied the simple (fixed-point) iteration and the modified (or full) Newton one in [10] to obtain the following combined algorithms:

*E-method with the simple iteration (SI-method):*

$$X_{k+1}^i = \bar{G}_k^\tau X_{k+1}^{i-1}, \quad i = 1, 2, \dots, N, \quad (5a)$$

$$X_{k+1}^0 = \left( \Pi(t_{k+1/2})^T, \Pi(t_{k+1})^T \right)^T \in \mathbf{R}^{2n}, \quad k = 0, 1, \dots, K-1, \quad (5b)$$

where  $X_{k+1} \stackrel{\text{def}}{=} ((x_{k+1/2})^T, (x_{k+1})^T)^T \in \mathbf{R}^{2n}$ , the mapping  $\bar{G}_k^\tau : \mathbf{R}^{2n} \rightarrow \mathbf{R}^{2n}$  presents the right-hand part of system (4),  $\bar{x}_k$  is an approximate solution of problem (2) obtained after  $N$  iterations; i.e.,  $\bar{x}_k \stackrel{\text{def}}{=} x_k^N$ ,  $\bar{g}_k^{(r)} \stackrel{\text{def}}{=} g^{(r)}(t_k, \bar{x}_k)$ , and  $\Pi(t)$  implies a predictor which is an interpolation formula based on the values of numerical solution  $\bar{x}_k$  ( $\Pi(t) \equiv \bar{x}_k$  in the trivial case).

*E-method with the Newton iteration (N-method):*

$$X_{k+1}^i = X_{k+1}^{i-1} - \partial \bar{F}_k^\tau (X_{k+1}^{i-1})^{-1} \bar{F}_k^\tau X_{k+1}^{i-1}, \quad i = 1, 2, \dots, N, \quad (6a)$$

$$X_{k+1}^0 = \left( \Pi(t_{k+1/2})^T, \Pi(t_{k+1})^T \right)^T \in \mathbf{R}^{2n}, \quad k = 0, 1, \dots, K-1, \quad (6b)$$

where  $\bar{F}_k^\tau \stackrel{\text{def}}{=} E_{2n} - \bar{G}_k^\tau$  ( $E_{2n}$  is the identity operator in  $\mathbf{R}^{2n}$ ) and  $\partial \bar{F}_k^\tau(X_{k+1}^{i-1})$  is the Jacobian of the mapping  $\bar{F}_k^\tau$  evaluated at the point  $X_{k+1}^{i-1}$ .

*E-method with the modified Newton iteration (MN-method):*

$$X_{k+1}^i = X_{k+1}^{i-1} - \partial \bar{F}_k^\tau(X_{k+1}^0)^{-1} \bar{F}_k^\tau X_{k+1}^{i-1}, \quad i = 1, 2, \dots, N, \quad (7a)$$

$$X_{k+1}^0 = \left( \Pi(t_{k+1/2})^T, \Pi(t_{k+1})^T \right)^T \in \mathbf{R}^{2n}, \quad k = 0, 1, \dots, K-1. \quad (7b)$$

Let us test our combined algorithms with the underlying method (4) on the restricted three body problem

$$x_1''(t) = x_1(t) + 2x_2'(t) - \mu_1 \frac{x_1(t) + \mu_2}{y_1(t)} - \mu_2 \frac{x_1(t) - \mu_1}{y_2(t)}, \quad (8a)$$

$$x_2''(t) = x_2(t) - 2x_1'(t) - \mu_1 \frac{x_2(t)}{y_1(t)} - \mu_2 \frac{x_2(t)}{y_2(t)}, \quad (8b)$$

$$y_1(t) = \left( (x_1(t) + \mu_2)^2 + x_2(t)^2 \right)^{3/2}, \quad y_2(t) = \left( (x_1(t) - \mu_1)^2 + x_2(t)^2 \right)^{3/2}, \quad (8c)$$

where  $t \in [0, T]$ ,  $T = 17.065216560157962558891$ ,  $\mu_1 = 1 - \mu_2$  and  $\mu_2 = 0.012277471$ . The initial values of problem (8) are:  $x_1(0) = 0.994$ ,  $x_1'(0) = 0$ ,  $x_2(0) = 0$ ,  $x_2'(0) = -2.00158510637908252240$ . The solution path of this problem is a periodic one with the period  $T$  (see, for example, [6]).

**Table 1.** Global errors of the EN-algorithm with trivial predictor for E-method (4)

$N$	$K$				
	$1.000 \cdot 10^{+04}$	$2.000 \cdot 10^{+04}$	$4.000 \cdot 10^{+04}$	$8.000 \cdot 10^{+04}$	$1.600 \cdot 10^{+05}$
1	$2.068 \cdot 10^{+00}$	$1.571 \cdot 10^{+00}$	$2.066 \cdot 10^{+00}$	$2.000 \cdot 10^{+00}$	$2.612 \cdot 10^{+00}$
2	$2.193 \cdot 10^{-02}$	$6.138 \cdot 10^{-05}$	$1.587 \cdot 10^{-07}$	$4.408 \cdot 10^{-10}$	$1.043 \cdot 10^{-12}$
3	$2.582 \cdot 10^{-04}$	$2.059 \cdot 10^{-07}$	$6.728 \cdot 10^{-10}$	$4.039 \cdot 10^{-13}$	$7.865 \cdot 10^{-14}$
4	$2.582 \cdot 10^{-04}$	$2.059 \cdot 10^{-07}$	$6.727 \cdot 10^{-10}$	$3.974 \cdot 10^{-13}$	$7.865 \cdot 10^{-14}$

**Table 2.** Global errors of the EMN-algorithm with trivial predictor for E-method (4)

$N$	$K$				
	$1.000 \cdot 10^{+04}$	$2.000 \cdot 10^{+04}$	$4.000 \cdot 10^{+04}$	$8.000 \cdot 10^{+04}$	$1.600 \cdot 10^{+05}$
1	$2.068 \cdot 10^{+00}$	$1.571 \cdot 10^{+00}$	$2.066 \cdot 10^{+00}$	$2.000 \cdot 10^{+00}$	$2.612 \cdot 10^{+00}$
2	$1.841 \cdot 10^{+00}$	$2.938 \cdot 10^{-01}$	$8.422 \cdot 10^{-03}$	$2.615 \cdot 10^{-04}$	$8.146 \cdot 10^{-06}$
3	$3.716 \cdot 10^{-03}$	$2.582 \cdot 10^{-05}$	$1.192 \cdot 10^{-07}$	$4.959 \cdot 10^{-10}$	$2.401 \cdot 10^{-12}$
4	$4.660 \cdot 10^{-04}$	$3.322 \cdot 10^{-07}$	$1.264 \cdot 10^{-10}$	$3.025 \cdot 10^{-13}$	$5.085 \cdot 10^{-14}$
5	$2.589 \cdot 10^{-04}$	$2.060 \cdot 10^{-07}$	$6.726 \cdot 10^{-11}$	$4.651 \cdot 10^{-13}$	$4.150 \cdot 10^{-14}$

**Table 3.** Global errors of the ESI-algorithm with trivial predictor for E-method (4)

$N$	$K$				
	$1.000 \cdot 10^{+04}$	$2.000 \cdot 10^{+04}$	$4.000 \cdot 10^{+04}$	$8.000 \cdot 10^{+04}$	$1.600 \cdot 10^{+05}$
3	$5.689 \cdot 10^{+00}$	$1.254 \cdot 10^{+00}$	$2.098 \cdot 10^{-01}$	$2.661 \cdot 10^{-02}$	$3.416 \cdot 10^{-03}$
4	$9.384 \cdot 10^{-01}$	$1.360 \cdot 10^{-01}$	$8.004 \cdot 10^{-03}$	$4.604 \cdot 10^{-04}$	$2.738 \cdot 10^{-05}$
5	$1.971 \cdot 10^{-01}$	$6.712 \cdot 10^{-03}$	$2.045 \cdot 10^{-04}$	$6.314 \cdot 10^{-06}$	$1.964 \cdot 10^{-07}$
6	$8.074 \cdot 10^{-03}$	$1.655 \cdot 10^{-04}$	$2.605 \cdot 10^{-06}$	$3.924 \cdot 10^{-08}$	$5.952 \cdot 10^{-10}$
7	$1.900 \cdot 10^{-03}$	$1.944 \cdot 10^{-05}$	$1.742 \cdot 10^{-07}$	$1.427 \cdot 10^{-09}$	$1.155 \cdot 10^{-11}$
8	$1.225 \cdot 10^{-03}$	$4.420 \cdot 10^{-06}$	$1.622 \cdot 10^{-08}$	$6.113 \cdot 10^{-11}$	$5.988 \cdot 10^{-13}$
9	$3.920 \cdot 10^{-04}$	$5.595 \cdot 10^{-07}$	$8.233 \cdot 10^{-10}$	$1.136 \cdot 10^{-12}$	$1.060 \cdot 10^{-13}$
10	$2.817 \cdot 10^{-04}$	$2.309 \cdot 10^{-07}$	$8.464 \cdot 10^{-11}$	$3.899 \cdot 10^{-13}$	$1.206 \cdot 10^{-13}$

Tables 1–3 exhibit that all the methods under testing are convergent for problem (8) and their orders depend on the number of iterations per grid point. We also see that the full (or modified) Newton iteration demonstrates the maximum order convergence with fewer iteration steps than the simple one. Unfortunately, it may be too expensive to compute the exact Jacobian of method (2) because of the high derivatives. Now we show how to simplify it.

First of all we replace the exact Jacobian  $\partial \bar{F}_k^\tau$  in method (6) with the following matrix:

$$A_k \stackrel{\text{def}}{=} \begin{pmatrix} 1 - \tau_k a_2^{(0)} \partial g_{k+1/2} & -\tau_k a_3^{(0)} \partial g_{k+1} \\ -\tau_k b_2^{(0)} \partial g_{k+1/2} & 1 - \tau_k b_3^{(0)} \partial g_{k+1} \end{pmatrix},$$

and obtain *the E-method with the simplified Newton iteration (ESN-method)*. If we test the new method on our problem (8) we come to Table 4. It is obvious that the ESN-method is cheaper than the previous Newton-type methods, but we see that it acts like the EMN-method. The corresponding theoretical result is given by

**Table 4.** Global errors of the ESN-algorithm with trivial predictor for E-method (4)

$N$	$K$				
	$1.000 \cdot 10^{+04}$	$2.000 \cdot 10^{+04}$	$4.000 \cdot 10^{+04}$	$8.000 \cdot 10^{+04}$	$1.600 \cdot 10^{+05}$
1	$2.102 \cdot 10^{+00}$	$1.862 \cdot 10^{+00}$	$2.056 \cdot 10^{+00}$	$1.989 \cdot 10^{+00}$	$1.839 \cdot 10^{+00}$
2	$2.976 \cdot 10^{+00}$	$2.467 \cdot 10^{-01}$	$1.024 \cdot 10^{-02}$	$3.774 \cdot 10^{-04}$	$1.304 \cdot 10^{-05}$
3	$2.347 \cdot 10^{-02}$	$3.725 \cdot 10^{-04}$	$5.377 \cdot 10^{-06}$	$8.578 \cdot 10^{-08}$	$1.393 \cdot 10^{-09}$
4	$2.066 \cdot 10^{-04}$	$1.274 \cdot 10^{-07}$	$9.618 \cdot 10^{-11}$	$1.803 \cdot 10^{-12}$	$1.140 \cdot 10^{-13}$
5	$2.571 \cdot 10^{-04}$	$2.059 \cdot 10^{-07}$	$6.739 \cdot 10^{-11}$	$4.583 \cdot 10^{-13}$	$4.428 \cdot 10^{-14}$

**Theorem 3** *Let the right-hand part of ODE (1) possess continuous derivatives up to order  $2p+5$  in a neighborhood of the solution  $x(t)$  on the interval  $[t_0, t_0+T]$ ,*

where  $p$  is a nonnegative integer number. Then the ESN-algorithm based on method (2) is convergent, and its error satisfies

$$\|x(t_k) - \bar{x}_k(N)\| \leq C_\mu \tau^\mu, \quad k = 1, 2, \dots, K, \quad (9)$$

where  $\mu = \min\{(\xi + 1)(N + 1) - 2, 2p + 4\}$ ,  $\xi = \min\{\zeta + 1, 2p + 3\}$ ,  $\zeta$  is the order of predictor  $\Pi(t)$  which is used to compute an initial approximation in the simplified Newton iteration,  $\tau \stackrel{\text{def}}{=} \max\{\tau_k\}$  is a diameter of the grid, and  $C_\mu$  is a constant.

## 4 Step Size Control for E-methods with High Derivatives

Now we consider the problem of step size selection for E-methods with high derivatives. Let us assume that the exact solution of problem (1) is known at the point  $t_k$ ; i. e.,  $x_k = x(t_k)$ , and the local error tolerance  $\epsilon_l$  is given. The notation  $\hat{x}_{k+1}$  denotes the numerical solution calculated by two steps of method (2) with the size  $\tau_k/2$ . Then the algorithm to control the local error of E-methods is presented as follows:

### Algorithm 1: Local step size control

*Step 1.* Compute  $\tilde{x}_{k+1}$ ,  $\hat{x}_{k+1}$ .

*Step 2.* Calculate  $\|\Delta\tilde{x}_{k+1}\| = \|\hat{x}_{k+1} - \tilde{x}_{k+1}\|/(1 - 1/2^{2p+4})$ .

*Step 3.*  $\tilde{\tau}_k := \theta \left( \epsilon_l / \|\Delta\tilde{x}_{k+1}\| \right)^{1/(2p+5)} \tau_k$ .

*Step 4.* If  $\|\Delta\tilde{x}_{k+1}\| > \epsilon_l$  then  $\tau_k := \tilde{\tau}_k$  and go to *Step 1*,

else  $\hat{x}_{k+1} := \tilde{x}_{k+1} + \Delta\tilde{x}_{k+1}$  and  $\tau_{k+1} := \tilde{\tau}_k$ .

*Step 5.*  $t_k := t_k + \tau_k$ ,  $k := k + 1$  and go to *Step 1*.

Here,  $\theta \in (0, 1)$  is a safety factor.

If we want to apply the local error control mechanism to iterative methods (5)–(7) we have to provide sufficient iterations at each grid point as the following estimates indicate:

$$N \geq \log_2 \left( \frac{2p + 7}{\xi + 1} \right) \quad \text{for the EN-method,} \quad (10)$$

$$N \geq \frac{2p + 7}{\xi + 1} - 1 \quad \text{for the EMN- and ESN-methods,} \quad (11)$$

$$N \geq 2p + 6 - \xi \quad \text{for the ESI-method.} \quad (12)$$

Let us test our methods with the local step size control on problem (8). If we use the combined algorithms mentioned above to solve this problem we come to the data presented in Table 5. The number of iterations per grid point for each type of the iterative E-methods was calculated by formulas (10)–(12), respectively. These numbers are given in parentheses at the first column of Table 5.

**Table 5.** Global errors of variable step size iterative algorithms based on E-method (4) with Algorithm 1 to control step size and with extrapolation

Type of iteration	required accuracy				
	$\epsilon_l = 10^{-03}$	$\epsilon_l = 10^{-04}$	$\epsilon_l = 10^{-05}$	$\epsilon_l = 10^{-06}$	$\epsilon_l = 10^{-07}$
SI(9)	$3.67 \cdot 10^{-03}$	$5.25 \cdot 10^{-04}$	$3.85 \cdot 10^{-06}$	$1.90 \cdot 10^{-06}$	$1.44 \cdot 10^{-07}$
N(3)	$1.46 \cdot 10^{-03}$	$1.94 \cdot 10^{-04}$	$2.53 \cdot 10^{-05}$	$9.16 \cdot 10^{-06}$	$3.73 \cdot 10^{-07}$
MN(5)	$1.51 \cdot 10^{-03}$	$1.74 \cdot 10^{-04}$	$2.74 \cdot 10^{-05}$	$2.08 \cdot 10^{-06}$	$3.00 \cdot 10^{-07}$
SN(5)	$2.92 \cdot 10^{-03}$	$6.67 \cdot 10^{-04}$	$2.36 \cdot 10^{-05}$	$2.55 \cdot 10^{-06}$	$1.34 \cdot 10^{-07}$

**Table 6.** Global errors of variable step size iterative algorithms based on E-method (4) with Algorithm 2 to control step size and with extrapolation ( $\epsilon_l = \epsilon_g$ )

Type of iteration	required accuracy				
	$\epsilon_g = 10^{-03}$	$\epsilon_g = 10^{-04}$	$\epsilon_g = 10^{-05}$	$\epsilon_g = 10^{-06}$	$\epsilon_g = 10^{-07}$
SI(9)	$1.54 \cdot 10^{-04}$	$1.95 \cdot 10^{-05}$	$1.71 \cdot 10^{-06}$	$2.00 \cdot 10^{-07}$	$1.96 \cdot 10^{-08}$
N(3)	$2.89 \cdot 10^{-04}$	$3.60 \cdot 10^{-05}$	$3.15 \cdot 10^{-06}$	$4.44 \cdot 10^{-07}$	$4.27 \cdot 10^{-08}$
MN(5)	$2.16 \cdot 10^{-04}$	$2.85 \cdot 10^{-05}$	$3.28 \cdot 10^{-06}$	$4.79 \cdot 10^{-07}$	$4.02 \cdot 10^{-08}$
SN(5)	$1.63 \cdot 10^{-04}$	$3.38 \cdot 10^{-05}$	$2.80 \cdot 10^{-06}$	$5.04 \cdot 10^{-07}$	$4.09 \cdot 10^{-08}$

We see that the local step size selection is quite efficient for all the types of iterations, but, unfortunately, it does not allow to find automatically the numerical solution with the accuracy set by the user. To improve it, we have to involve a global error control in the step size selection procedure.

By this reason, we apply the local-global step size selection developed with the aim of controlling the global error of Runge-Kutta formulas in [9] to our E-methods with high derivatives. First, we suppose that the numerical solution  $\tilde{x}_k$  has been computed at the point  $t_k$  with an accuracy of  $O(\tau_k^{2p+5})$ . Second, the local and global errors tolerances  $\epsilon_l$  and  $\epsilon_g$  have been given. Third, we assume that some step size  $\tau_k$  has been fixed. Then the algorithm of the local-global step size selection for method (2) is presented as follows:

### Algorithm 2: Local-global step size control

- Step 1.* By Algorithm 1, find estimates  $\Delta\tilde{x}_{k+1}$   $\Delta\tilde{x}_{k+2}$  of the local error of method (2) at the points  $t_{k+1}$ ,  $t_{k+2}$  and step sizes  $\tilde{\tau}_k$ ,  $\tilde{\tau}_{k+1}$  for the tolerance  $\epsilon_l$  (change the step size  $\tau_k$ , if necessary).
- Step 2.* Determine the coefficient of the principal term of the local error  $\hat{\psi}_{2p+5}(t_{k+1})$ .
- Step 3.* Compute the coefficient of the principal term of the global error  $\psi_{2p+4}(t_{k+1})$  by the formula

$$\psi_{2p+4}(t_{k+1}) := \left( E_n - \tau_k \partial_x g(t_{k+1}, \tilde{x}_{k+1}) \right)^{-1} \left( \tau_k \hat{\psi}_{2p+5}(t_{k+1}) \right). \quad (14)$$

- Step 4.* Find an estimate  $\Delta\psi_{2p+4}(t_{k+1})$  of the local error of method (14) by the Richardson extrapolation or by two one-step methods of different orders (see [9], for details).

- Step 5.* If  $\|\Delta\psi_{2p+4}(t_{k+1})\| > \epsilon_g/100$  then calculate a new step size  $\tau_k$  as shown in Algorithm 1 and go to *Step 1*,  
 else  $q := \left\lceil \tau_k \left( \|\psi_{2p+4}(t_{k+1})\| / \epsilon_g \right)^{1/(2p+4)} \right\rceil + 1$ .  
*Step 6.* If  $q > 1$  then  $\tau_k := \tau_k/q$  and go to *Step 1*,  
 else  $\tilde{x}_{k+1} := \tilde{x}_{k+1} + \Delta\tilde{x}_{k+1}$ ,  $\Delta\tilde{x}_{k+1} := \Delta\tilde{x}_{k+2}$  and  $\tau_{k+2} := \tilde{\tau}_{k+1}$ .  
*Step 7.*  $t_k := t_k + \tau_k$ ,  $k := k + 1$  and go to *Step 1*.

Here, the square brackets mean an integer part of the number. A fuller description of Algorithm 2 for the iterative E-methods with high derivatives will appear in [10].

If we now test our iterative E-methods with Algorithm 2 to select a step size on problem (8) we come to Table 6. These data show the great advantage of the local-global step size control (Algorithm 2) over the local one (Algorithm 1), if we compare the global errors obtained in the last numerical experiment with Table 5. Thus, E-methods presented in the paper together with the local-global step size control can be a good computational technique to solve many practical problems including stiff ones.

## References

1. Aul'chenko, S. M., Latypov, A. F., Nikulichev, Yu. V.: A method for the numerical integration of systems of ordinary differential equations using Hermite interpolation polynomials. (*in Russian*) Zh. Vychisl. Mat. Mat. Fiz. **38** (1998) No. 10, 1665–1670; *translation in* Comput. Math. Math. Phys. **38** (1998) No. 10, 1595–1601
2. Bakhvalov, N.S.: Numerical methods. (*in Russian*) Nauka, Moscow, 1975
3. Berezin, I.S., Zhidkov, N.P.: Methods of computations. V. 1. (*in Russian*) Gos. izd-vo fiz.-mat. lit-ry, Moscow, 1962
4. Butcher, J.C.: Numerical methods for ordinary differential equations. John Wiley & Son, Chichester, 2003
5. Fehlberg, E.: New high-order Runge-Kutta formulas with step size control for systems of first and second order differential equations. ZAMM. **44** (1964) T17–T19
6. Hairer, E., Nørsett, S.P., Wanner, G.: Solving ordinary differential equations I: Nonstiff problems. Springer-Verlag, Berlin, 1987
7. Hairer, E., Wanner, G.: Solving ordinary differential equations II: Stiff and differential-algebraic problems. Springer-Verlag, Berlin, 1996
8. Kastlunger, K.H., Wanner, G.: Runge-Kutta processes with multiple nodes. Computing. **9** (1972) 9–24
9. Kulikov, G.Yu.: A local-global version of a stepsize control for Runge-Kutta methods. Korean J. Comput. Appl. Math. **7** (2000) No. 2, 289–318
10. Kulikov, G.Yu., Merkulov, A.I.: On one-step collocation methods with high derivatives for solving ordinary differential equations. (*in Russian*) Zh. Vychisl. Mat. Mat. Fiz. (to appear); *translation in* Comput. Math. Math. Phys. (to appear)
11. Nørsett, S.P.: One-step methods of Hermite type for numerical integration of stiff systems. BIT. **14** (1974) 63–77
12. Ortega, J.M., Poole, W.G.: An introduction to numerical methods for differential equations. Pitman Publishing Inc., 1981