IMM
INSTITUTE OF MATHEMATICAL MODELLING

Technical University of Denmark
DK-2800 Lyngby – Denmark

1996-04-16
pgt

CONVERGENCE AND IMPLEMENTATION OF IMPLICIT RUNGE-KUTTA METHODS FOR DAEs

Gennady Yu. Kulikov
Per G. Thomsen

TECHNICAL REPORT

IMM-REP-1996-7

IMM

Convergence and implementation of implicit Runge-Kutta methods for DAEs

G.Yu. Kulikov[*]   P.G. Thomsen[†]

Abstract

We consider three classes of numerical methods for solving the Cauchy problem for systems of differential-algebraic equations of index 1. These methods use implicit variable step size Runge-Kutta formulae and iterative processes such as simple iteration, and full and modified Newton iterations. For these methods we prove convergence theorems and give error estimates. We obtain some limits which allow implementation of variable step size methods more efficiently. We consider different step size control procedures and illustrate their efficiency by applying the methods to a given test problem.

AMS subject classifications: 65L06

Key words: differential-algebraic equations, index, Runge-Kutta methods, iterative methods, convergence results, error estimates, step size control

1 Introduction

This paper considers numerical methods for solving systems of differential-algebraic equations (DAEs) in the form

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

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

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

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

[†] Institute of Mathematical Modelling, Technical University of Denmark, Building 321, DK–2800, Denmark. E-mail: pgt@imm.dtu.dk

(1.1d) $$y(t_0) = y^0,$$

where $t \in [t_0, t_0 + T]$, $x(t) \in \mathbf{R}^m$, $y(t) \in \mathbf{R}^n$, $\alpha(t) \in \mathbf{R}^l$ is some 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 the initial conditions are consistent: $y^0 = f(x^0, y^0, \alpha(0))$. The interest of problem (1.1) lies in the fact that many mathematical models in medicine, chemistry, mechanics, and technology can be described by such systems [1]–[5], [8], [9], [15], [17]. So numerical methods for problem (1.1) are being studied intensively.

Theoretical results from papers [7], [8], [10]–[16] provide the basis for constructing numerical methods which have a high order and allow to solve problems from practical applications rather accurately even using large step sizes. But from the practical point of view those algorithms are not effective enough because they do not take into account specific features of the solution path and thus require considerable expenditures of computing time.

Combined numerical methods using variable step size[1] are investigated in Section 2. For these methods theorems on the convergence are proved and error estimates of approximate solution depending on both gridsize and number of iterations are obtained. Numerical variable step size methods[2] are presented and substantiated in Section 3. In Section 4 the theoretical results of the paper are compared with results of numerical experiments. Some recommendations on practical implementation of the numerical variable step size methods allowing to reduce the total number of arithmetical operations are given in Section 5.

2 Numerical methods using variable step size

Since any nonautonomous system (1.1) may be converted to an autonomous system by introducing a new independent variable we will only consider autonomous problem:

(2.1a) $$x'(t) = g\big(x(t), y(t)\big),$$

(2.1b) $$y(t) = f\big(x(t), y(t)\big),$$

(2.1c) $$x(0) = x^0,$$

(2.1d) $$y(0) = y^0,$$

[1] Here and later this term denotes numerical methods that use a nonuniform grid but says nothing about a procedure for choosing such grid

[2] Here and later this term denotes numerical methods together with a procedure for choosing a nonuniform grid

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 initial conditions (2.1c,d) are consistent, i.e., $y^0 = f(x^0, y^0)$.

We introduce in the interval $[0,T]$ a nonuniform grid

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

and define τ as the diameter of grid ω_τ

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

Applying an implicit l-stage Runge-Kutta (RK) method using variable step size

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

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

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

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

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

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

We assume that initial conditions (2.1c,d) are satisfied. Applying three iterative processes, namely the simple iteration, and the full and modified Newton iterations for solving problem (2.2) we obtain three classes of numerical methods for solving DAEs (2.1) using variable step size.

Now we will define each class of the methods in detail. We will denote by $z(t)$ the vector formed by combining the vectors $x(t)$ and $y(t)$ $\left(z(t) = \left(x(t), y(t)\right)^T \in \mathbf{R}^{m+n}\right)$, and by G the mapping obtained by combining the mappings g and f $\left(G = (g, f)^T : D \subset \mathbf{R}^{m+n} \to \mathbf{R}^{m+n}\right)$. Let $z(t_k)$ be the value of the exact solution of problem (2.1) at the point t_k, and let $\tilde{z}_k$ be the value of the exact solution of problem (2.2) at t_k, and $\bar{z}_k = \bar{z}_k(N)$ be the value of the approximate solution of problem (2.2) at t_k obtained after N iterations of some iterative method.

In addition, we introduce the vector

$$Z_{k+1} = (z_{k1}, ..., z_{kl}, z_{k+1})^T \in \mathbf{R}^{(m+n)(l+1)}$$

and define the mapping

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

by the formula

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

Using these notations we can write three classes of the numerical methods.

The Runge-Kutta-Simple Iteration (RKSI) method using variable step size is

(2.3a) $$Z_{k+1}^i = \bar{G}_k^{\tau} Z_{k+1}^{i-1},$$

(2.3b) $$Z_{k+1}^0 = (\bar{z}_k, ..., \bar{z}_k)^T \in \mathbf{R}^{(m+n)(l+1)},$$

$$\bar{z}_k = z_k^N, \quad k = 0, 1, ..., K-1,\ i = 1, 2, ..., N,$$

(2.3c) $$\bar{Z}_0 = Z^0 = (z^0, ..., z^0)^T \in \mathbf{R}^{(m+n)(l+1)}.$$

The equality $\bar{z}_k = z_k^N$ means we take the last $m+n$ components of the vector Z_k^N obtained after N iterations of algorithm (2.3a) as the approximate solution of problem (2.2) at the point t_k.

The Runge-Kutta-Newton (RKN) method using variable step size is

(2.4a) $$Z_{k+1}^i = Z_{k+1}^{i-1} - \partial \bar{F}_k^{\tau}(Z_{k+1}^{i-1})^{-1} \bar{F}_k^{\tau} Z_{k+1}^{i-1},$$

(2.4b) $$Z_{k+1}^0 = (\bar{z}_k, ..., \bar{z}_k)^T \in \mathbf{R}^{(m+n)(l+1)},$$

$$\bar{z}_k = z_k^N, \quad k = 0, 1, ...K-1,\ i = 1, ..., N,$$

(2.4c) $$\bar{Z}_0 = Z^0 = (z^0, ..., z^0)^T \in \mathbf{R}^{(m+n)(l+1)},$$

where $\bar{F}_k^{\tau} = I_{(m+n)(l+1)} - \bar{G}_k^{\tau}$ and $\partial \bar{F}_k^{\tau}(Z_{k+1}^{i-1})$ is the Jacobian of the mapping $\bar{F}_k^{\tau}$ at the point Z_{k+1}^{i-1} ($I_{(m+n)(l+1)}$ is the indentity matrix of dimension $(m+n)(l+1)$).

The Runge-Kutta-modified Newton (RKmN) method using variable step size is

$$(2.5a) \qquad Z^i_{k+1} = Z^{i-1}_{k+1} - \partial\bar{F}^T_k(\bar{Z}_k)^{-1}\bar{F}^T_k Z^{i-1}_{k+1},$$

$$(2.5b) \qquad Z^0_{k+1} = (\bar{z}_k, ..., \bar{z}_k)^T \in \mathbf{R}^{(m+n)(l+1)},$$

$$\bar{z}_k = z^N_k, \quad k = 0, 1, ..., K-1, \; i = 1, 2..., N,$$

$$(2.5c) \qquad \bar{Z}_0 = Z^0 = (z^0, ..., z^0)^T \in \mathbf{R}^{(m+n)(l+1)}.$$

Now we will prove the convergence of methods (2.3)–(2.5). We assume that problem (2.1) satisfies the following conditions on the compact set D_1.

I. *The smoothness condition.* The mapping $G : D_1 \subset \mathbf{R}^{m+n} \rightarrow \mathbf{R}^{m+n}$ has continuous partial derivatives of orders $1, 2, ..., s+2$ on the set D_1, where s is the order of the underlying RK formula.
Hence we have the estimate

$$\|\partial G(z') - \partial G(z'')\| \leq \gamma\|z' - z''\| \quad \forall\, z', z'' \in D_1,$$

where γ is a constant.
II. *The nonsingularity condition.* The matrix $I_n - \partial f_y(x, y)$ is nonsingular for any $z \in D_1$.
III. *The inclusion condition* [3]. There exists a convex set D_0 such that $z^0 \in D_0 \subset D_1$.

If the conditions I–III hold then problem (2.1) has a unique solution $z(t) \subset D_0$ [12] and the following theorem on the convergence of the exact solution of algebraic equations (2.2) to the exact solution of DAEs (2.1) is valid.

Theorem 1 *Suppose problem (2.1) satisfies conditions I–III on the set D_1. Then there is a $\tau_0 > 0$ such that for any grid ω_τ with diameter $\tau < \tau_0$ a unique solution of problem (2.2) exists converging to the exact solution of problem (2.1) as $\tau \rightarrow 0$. Further, the error estimate*

$$(2.6) \qquad \|z(t_k) - \tilde{z}_k\| = O(\tau^s), \quad k = 0, 1, ..., K,$$

is valid, where s is the order of the underlying RK formula.

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

Using theorem 1 we obtain the following convergence results for the methods (2.3)-(2.5). Further we introduce condition IV:

IV. *The boundedness condition.* The estimate

$$\|\partial f(z)\| \leq d < 1$$

holds for any $z \in D_1$.

Theorem 2 *Suppose problem (2.1) satisfies conditions I, III, and IV on the set D_1. Then there is a $\tau_0 > 0$ and a function $N : (0, \tau_0) \longrightarrow N$ such that for any grid ω_τ with diameter $\tau < \tau_0$ the approximate solution $\bar{z}_k(N), k = 1, 2, ..., K$, obtained by RKSI method (2.3) exists and converges to the exact solution of problem (2.1) as $\tau \rightarrow 0$. Further, we have the following error estimate for the RKSI method:*

$$\|z(t_k) - \bar{z}(t_k)\| = O(d^N + \tau^s), \quad k = 0, 1, ..., K, \tag{2.7}$$

where s is the order of the underlying RK formula.

Theorem 3 *Suppose problem (2.1) satisfies conditions I-III on the set D_1. Then there is a $\tau_0 > 0$ and a function $N : (0, \tau_0) \longrightarrow N$ such that for any grid ω_τ with diameter $\tau < \tau_0$ the approximate solution $\bar{z}_k(N), k = 1, 2, ..., K$, obtained by RKN method (2.4) exists and converges to the exact solution of problem (2.1) as $\tau \rightarrow 0$. Further, we have the following error estimate for the RKN method:*

$$\|z(t_k) - \bar{z}(t_k)\| = O(\tau^s), \quad k = 0, 1, ..., K, \tag{2.8}$$

where s is the order of the underlying RK formula.

Theorem 4 *Suppose problem (2.1) satisfies conditions I-III on the set D_1. Then there is a $\tau_0 > 0$ and a function $N : (0, \tau_0) \longrightarrow N$ such that for any grid ω_τ with diameter $\tau < \tau_0$ the approximate solution $\bar{z}_k(N), k = 1, 2, ..., K$, obtained by RKmN method (2.5) exists and converges to the exact solution of problem (2.1) as $\tau \rightarrow 0$. Further, we have the following error estimate for the RKmN method:*

$$\|z(t_k) - \bar{z}(t_k)\| = O(\tau^s), \quad k = 0, 1, ..., K, \tag{2.9}$$

where s is the order of the underlying RK formula.

Theorems 1–4 may be proved as well as convergence results for the Euler method with the simple iteration and the implicit Adams method of order two with both the full and modified Newton iterations were proved in [11]–[15].

Theorems 3 and 4 give the same error estimates for the both RKN and RKmN methods. But there are different limits for the minimal number of iterations which allow to guarantee the convergence of order $O(\tau^s)$ for each method. They are:

$$(2.10) \qquad N_0 \geq \log_2(s+1) \qquad \text{for the RKN method,}$$

$$(2.11) \qquad N_0 \geq s \qquad \text{for the RKmN method.}$$

Thus, we have constructed three classes of the combined numerical methods using variable step size for solving DAEs (2.1) and proved their convergence. However, from the practical point of view it gives us nothing because we cannot build the optimal grid ω_τ if we do not know the behavior of the solution path. But variable step size methods allow to solve this problem. In this case some grid close to the optimal one that reduces expenditures of computer time significantly is built automatically.

3 Numerical variable step size methods

As for numerical solving ordinary differential equations (ODEs) by numerical variable step size methods we will choose the next step τ_k for any method from Section 2 such that the local error does not exceed a given value ϵ, which is called the error tolerance [6]. Thus, we need to solve two problems in order to construct the variable step size methods. First, we must find the local error. Second, we must define step size which allows to guarantee that the local error does not exceed ϵ.

We will now solve the first problem. We will define the local error of algebraic equations (2.2), i.e., we will analyse the difference $z(t_{k+1}) - \tilde{z}_{k+1}$ provided $z(t_k) = \tilde{z}_k = z_k$. From conditions I–III and the Leibniz formula [6] we can obtain the following representation for the x-component of the local error:

$$(3.1a) \qquad x(t_{k+1}) - \tilde{x}_{k+1} = \psi(x_k, y_k)\tau_k^{s+1} + O(\tau_k^{s+2}),$$

where

$$\psi(x_k, y_k) = \frac{1}{(s+1)!}\Big(\sum_{p+q=s} \partial^s g_{x^{(p)}y^{(q)}}(x_k, y_k) \cdot x_k^{(p)} \cdot y_k^{(q)}$$
$$-(s+1) \sum_{p+q=s} \partial^s \tilde{g}_{x^{(p)}y^{(q)}}(x_k, y_k) \cdot x_k^{(p)} \cdot y_k^{(q)} \Big),$$

$$\tilde{g}(x_k, y_k) = x_k + \tau_k \sum_{i=1}^{l} b_i g(x_k, y_k),$$

$$\partial^s g_{x^{(p)}y^{(q)}}(x, y) = \frac{\partial^{p_1}}{\partial x_1^{p_1}} \cdots \frac{\partial^{p_m}}{\partial x_m^{p_m}} \frac{\partial^{q_1}}{\partial y_1^{q_1}} \cdots \frac{\partial^{q_n}}{\partial y_n^{q_n}} g(x, y),$$

p and q are multi-indices such that $x^{(p)} = x_1^{(p_1)} \cdot \ldots \cdot x_m^{(p_m)}$, $y^{(q)} = y_1^{(q_1)} \cdot \ldots \cdot y_n^{(q_n)}$, and $p_1 + \ldots + p_m = p$, $q_1 + \ldots + q_n = q$. Thus, $\psi(x, y)$ is a smooth function with

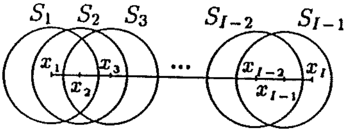

Figure 1: The open covering of the interval $[x(t_{k+1}), \tilde{x}_{k+1}]$.

respect to the partial derivatives of the right part of problem (2.1) as arguments. Then taking into account that we have used the partial derivatives up to the order s only, from the smoothness condition (I) we obtain $\psi(x,y) \in C^2_{D_1}$ and $\psi(x,y)$ is a bounded mapping on the set D_1.

Now we will prove that the similar formula

$$y(t_{k+1}) - \tilde{y}_{k+1} = \phi(x_k, y_k)\tau_k^{s+1} + O(\tau_k^{s+2}), \tag{3.1b}$$

where $\phi(x,y)$ is a bounded mapping, is valid for the y-components of the local error as well as for the x-components. Repeating the proof of formula (2.68) in [15, p.85] we construct a sequence of points x_i, $i = 1,2,...,I$, in the interval $[x(t_{k+1}), \tilde{x}_{k+1}]$ and open balls $S_i = S(x_i, r_i)$ such that, first, $x_{i+1} \in S_i$, $i = 1,2,...,I-1$, and $[x(t_{k+1}), \tilde{x}_{k+1}] \subset \bigcup_{i+1}^{I-1} S_i$, and, second, the implicit function theorem is valid in each ball S_i (see fig. 1). Thus, we obtain the following chain of equalities:

$$\begin{aligned} y(t_{k+1}) - \tilde{y}_{k+1} &= \sum_{i=1}^{I-1}(y_{i+1} - y_i) = \sum_{i=1}^{I-1}(h_i(x_{i+1}) - h_i(x_i)) \\ &= \sum_{i=1}^{I-1}\Big(\partial h_i(x_i)(x_{i+1} - x_i) + O\big((x_{i+1} - x_i)^2\big)\Big), \end{aligned} \tag{3.2}$$

where y_i is the solution of the system $y = f(x_i, y)$, and

$$\partial h_i(x_i) = -\big(I_n - \partial f_y(x_i, y_i)\big)^{-1}\partial f_x(x_i, y_i).$$

The substitution (3.1a) into (3.2) gives

$$y(t_{k+1}) - \tilde{y}_{k+1} = \sum_{i=1}^{I-1}\partial h_i(x_i)(x_{i+1} - x_i) + O(\tau_k^{2s+2}).$$

Then taking into account the smoothness of $\partial h_i(x)$ we have

$$y(t_{k+1}) - \tilde{y}_{k+1} = \partial h_1(x_k)\big(x(t_{k+1}) - x_{k+1}\big) + O(\tau_k^{s+2}). \tag{3.3}$$

Finally, using (3.1a) and (3.3) we obtain (3.1b), where $\phi(x_k, y_k) = \partial h_1(x_k) \cdot \psi(x_k, y_k)$. Moreover, we have $\phi(x, y) \in C^2_{D_1}$ because of the smoothness of both $\partial h_1(x)$ and $\psi(x, y)$ on the set D_1. Hence, $\phi(x, y)$ is a bounded mapping on the set D_1.

To estimate the local error we will take into account the principal error term only. So for grids ω_τ with sufficiently small diameter τ we can neglect the terms of order $O(\tau_k^{s+2})$ in formulae (3.1) and consider that the following expressions are valid:

$$x(t_{k+1}) - \tilde{x}_{k+1} \cong \psi(x_k, y_k)\tau_k^{s+1}, \tag{3.4a}$$

$$y(t_{k+1}) - \tilde{y}_{k+1} \cong \phi(x_k, y_k)\tau_k^{s+1}. \tag{3.4b}$$

We will use two approaches to estimate the local error. The first is based on the Richardson extrapolation. The second uses two RK formulae of different orders [6].

We start with the first approach. We assume that the value of the exact solution of problem (2.1) at the point t_k is known, i.e., $z(t_k) = \tilde{z}_k = z_k$, and compute the value of the exact solution of problem (2.2) at the point $t_{k+1} = t_k + \tau_k$. We denote this solution by $\tilde{z}_{k+1}$. Taking into account formulae (3.4) we have that $\tilde{z}_{k+1}$ is an approximate solution of problem (2.1) of order $O(\tau_k^{s+1})$ at the point t_{k+1}. Now we return at the point t_k and make two steps of the size $\tau_k/2$ using system (2.2). After the first one we obtain an approximate solution of problem (2.1) $\hat{z}_{k+1/2}$ at the point $t_{k+1/2} = t_k + \tau_k/2$. According to (3.4) the method has the following local error on this step:

$$x(t_{k+1/2}) - \hat{x}_{k+1/2} \cong \psi(x_k, y_k)(\tau_k/2)^{s+1}, \tag{3.5a}$$

$$y(t_{k+1/2}) - \hat{y}_{k+1/2} \cong \phi(x_k, y_k)(\tau_k/2)^{s+1}. \tag{3.5b}$$

After the second step we obtain an approximate solution of problem (2.1) at the point t_{k+1}. In this case the local error is

$$\hat{x}(t_{k+1}) - \hat{x}_{k+1} \cong \psi(\hat{x}_{k+1/2}, \hat{y}_{k+1/2})(\tau_k/2)^{s+1}, \tag{3.6a}$$

$$\hat{y}(t_{k+1}) - \hat{y}_{k+1} \cong \phi(\hat{x}_{k+1/2}, \hat{y}_{k+1/2})(\tau_k/2)^{s+1}, \tag{3.6b}$$

where $\hat{z}(t)$ is the exact solution of problem (2.1) under the initial conditions

$$\hat{x}(t_{k+1/2}) = \hat{x}_{k+1/2}, \tag{3.6c}$$

$$\hat{y}(t_{k+1/2}) = \hat{y}_{k+1/2}. \tag{3.6d}$$

Since the mappings $\psi(x,y), \phi(x,y) \in C^2_{D_1}$ then using the Taylor expansion of both $\psi(x,y)$ and $\phi(x,y)$, and taking into account the smallness of τ (and hence τ_k) we obtain

$$\hat{x}(t_{k+1}) - \hat{x}_{k+1} \cong \psi(x_k, y_k)(\tau_k/2)^{s+1}, \tag{3.7a}$$

$$\hat{y}(t_{k+1}) - \hat{y}_{k+1} \cong \phi(x_k, y_k)(\tau_k/2)^{s+1}. \tag{3.7b}$$

We will now estimate the difference $x(t) - \hat{x}(t)$ at the point t_{k+1}, where the $x(t)$ and $\hat{x}(t)$ are the x-components of the exact solutions of DAEs (2.1) under initial conditions (2.1c,d) and (3.6c,d), respectively.

We rewrite equation (2.1a) in the integral form

$$x(t) = x(t_{k+1/2}) + \int\limits_{t_{k+1/2}}^{t} g(x(\xi), y(\xi))\, d\xi. \tag{3.8}$$

Using (3.8) and the smoothness of problem (2.1) we have

$$\begin{aligned} x(t_{k+1}) - \hat{x}(t_{k+1}) &= x(t_{k+1/2}) - \hat{x}(t_{k+1/2}) \\ &+ \int\limits_{t_{k+1/2}}^{t} \Big(g(x(\xi), y(\xi)) - \hat{g}(x(\xi), y(\xi))\Big)\, d\xi \\ &= x(t_{k+1/2}) - \hat{x}(t_{k+1/2}) + O\Big(\tau_k\big(z(t_{k+1/2}) - \hat{z}(t_{k+1/2})\big)\Big). \end{aligned}$$

The substitution of (3.5) and (3.6c) into the last equality gives

$$x(t_{k+1}) - \hat{x}(t_{k+1}) \cong x(t_{k+1/2}) - \hat{x}(t_{k+1/2}) \cong x(t_{k+1/2}) - \hat{x}_{k+1/2}. \tag{3.9}$$

Then taking into account (3.5a), (3.7a), and (3.9) we obtain

$$\begin{aligned} x(t_{k+1}) - \hat{x}_{k+1} &= x(t_{k+1}) - \hat{x}(t_{k+1}) + \hat{x}(t_{k+1}) - \hat{x}_{k+1} \\ &\cong x(t_{k+1/2}) - \hat{x}_{k+1/2} + \hat{x}(t_{k+1}) - \hat{x}_{k+1} \cong 2\psi(x_k, y_k)(\tau_k/2)^{s+1}. \end{aligned} \tag{3.10a}$$

Now we will prove the same relation for the y-components

$$y(t_{k+1}) - \hat{y}_{k+1} \cong 2\phi(x_k, y_k)(\tau_k/2)^{s+1}. \tag{3.10b}$$

We present the left part of formula (3.10b) as the sum of two differences

$$y(t_{k+1}) - \hat{y}_{k+1} = y(t_{k+1}) - \hat{y}(t_{k+1}) + \hat{y}(t_{k+1}) - \hat{y}_{k+1}. \tag{3.11}$$

Repeating the proof of formula (3.1b) for the first one we have

$$y(t_{k+1}) - \hat{y}(t_{k+1}) \cong \phi(x_k, y_k)(\tau_k/2)^{s+1}. \tag{3.12}$$

Formula (3.7b) is valid for the second difference. Then substituting both (3.7b) and (3.12) into (3.11) we finally obtain (3.10b).

Using (3.4) and (3.11) we can estimate the principal terms of the local errors for the approximate solutions of DAEs (2.1) at the point t_{k+1}

$$(3.13a) \qquad \left\| \left(\psi(x_k, y_k), \phi(x_k, y_k)\right)^T \right\| \tau_k^{s+1} \cong \|\hat{z}_{k+1} - \tilde{z}_{k+1}\| / (1 - 1/2^s),$$

$$(3.13b) \qquad \left\| \left(\psi(x_k, y_k), \phi(x_k, y_k)\right)^T \right\| (\tau_k/2)^{s+1} \cong \|\hat{z}_{k+1} - \tilde{z}_{k+1}\| / (2^s - 1).$$

Since (3.13b) gives a smaller local error than (3.13a) we choose $\hat{z}_{k+1}$ as the approximate solution of problem (2.1) at the point t_{k+1}. Moreover, adding the value of the principal term of the local error to $\hat{z}_{k+1}$ we can compute the approximate solution more exactly. In the case of the local extrapolation we obtain the following local error estimate:

$$(3.14) \qquad \|z(t_{k+1}) - \hat{z}_{k+1}\| = O(\tau_k^{s+2}).$$

The second way to define the local error is based on using two RK formulae of different orders. Here we will apply 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

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

[6]. Using the first method for solving DAEs (2.1) we obtain the following system of algebraic equations:

$$(3.15a) \qquad x_{k1} = x_k + \tau_k g(x_{k1}, y_{k1})/4 + (1/2 - \sqrt{3}/3)\tau_k g(x_{k2}, y_{k2})/2,$$

$$(3.15b) \qquad y_{k1} = f(x_{k1}, y_{k1}),$$

(3.15c) $$x_{k2} = x_k + (1/2 + \sqrt{3}/3)\tau_k g(x_{k1}, y_{k1})/2 + \tau_k g(x_{k2}, y_{k2})/4,$$

(3.15d) $$y_{k2} = f(x_{k2}, y_{k2}),$$

(3.15e) $$x_{k+1} = x_k + \tau_k g(x_{k1}, y_{k1})/2 + \tau_k g(x_{k2}, y_{k2})/2,$$

(3.15f) $$y_{k+1} = f(x_{k+1}, y_{k+1}),$$

(3.15g) $$z_k = z(t_k).$$

The second system of algebraic equations is got by K&B method:

(3.16a) $$\begin{aligned} x_{k1} = x_k &+ 5\tau_k g(x_{k1}, y_{k1})/36 + (2/9 - \sqrt{15}/15)\tau_k g(x_{k2}, y_{k2}) \\ &+ (5/36 - \sqrt{15}/30)\tau_k g(x_{k3}, y_{k3}), \end{aligned}$$

(3.16b) $$y_{k1} = f(x_{k1}, y_{k1}),$$

(3.16c) $$\begin{aligned} x_{k2} = x_k &+ (5/36 + \sqrt{15}/24)\tau_k g(x_{k1}, y_{k1}) + 2\tau_k g(x_{k2}, y_{k2})/9 \\ &+ (5/36 - \sqrt{15}/24)\tau_k g(x_{k3}, y_{k3}), \end{aligned}$$

(3.16d) $$y_{k2} = f(x_{k2}, y_{k2}),$$

(3.16e) $$\begin{aligned} x_{k3} = x_k &+ (5/36 + \sqrt{15}/30)\tau_k g(x_{k1}, y_{k1}) + (2/9 + \sqrt{15}/15) \\ &\cdot \tau_k g(x_{k2}, y_{k2}) + 5\tau_k g(x_{k3}, y_{k3})/36, \end{aligned}$$

(3.16f) $$y_{k3} = f(x_{k3}, y_{k3}),$$

(3.16g) $$\begin{aligned} x_{k+1} = x_k &+ 5\tau_k g(x_{k1}, y_{k1})/18 + 4\tau_k g(x_{k2}, y_{k2})/9 \\ &+ 5\tau_k g(x_{k3}, y_{k3})/18, \end{aligned}$$

(3.16h) $$y_{k+1} = f(x_{k+1}, y_{k+1}),$$

(3.16i) $$z_k = z(t_k).$$

Initial conditions (3.15g) and (3.16i) assume that we know the exact solution of problem (2.1) at the point t_k.

We denote the solution and the local error of problem (3.15) at the point t_{k+1} by $\tilde{z}_{k+1}$ and $\tilde{e}_{k+1}$, respectively. Similarly, we use notations $\hat{z}_{k+1}$ and $\hat{e}_{k+1}$ for problem (3.16). The local errors of methods (3.15) and (3.16) have the forms

$$\tilde{e}_{k+1} = z(t_{k+1}) - \tilde{z}_{k+1} = O(\tau_k^{4+1}),$$

$$\hat{e}_{k+1} = z(t_{k+1}) - \hat{z}_{k+1} = O(\tau_k^{6+1}).$$

From these equalities we obtain the estimate of the local error for method (3.15)

$$\tilde{e}_{k+1} = \hat{z}_{k+1} - \tilde{z}_{k+1} + O(\tau_k^{6+1}). \tag{3.17}$$

Leaving in (3.17) the principal term only we have

$$\tilde{e}_{k+1} \cong \hat{z}_{k+1} - \tilde{z}_{k+1}. \tag{3.18}$$

Thus, we have obtained two different ways to estimate the principal term of the local error of method (2.2). If we have known the solution of system (2.2) then we can estimate the local error of this method by (3.13) or (3.18). However, we cannot obtain the exact solution of problem (2.2) in practice since in the general case (2.2) is a system of nonlinear algebraic equations. So we can define some approximation to the exact solution only.

Let $\bar{z}_{k+1}(N)$ be some iterative approximation to $\tilde{z}_{k+1}$ found by any method of Section 2 and $\check{z}_{k+1}(N)$ be an approximation to $\hat{z}_{k+1}$. Then the following equalities take place:

$$z(t_{k+1}) - \bar{z}_{k+1}(N) = z(t_{k+1}) - \tilde{z}_{k+1} + \tilde{z}_{k+1} - \bar{z}_{k+1}(N), \tag{3.19a}$$

$$z(t_{k+1}) - \check{z}_{k+1}(N) = z(t_{k+1}) - \hat{z}_{k+1} + \hat{z}_{k+1} - \check{z}_{k+1}(N). \tag{3.19b}$$

Using (3.1) we can estimate the first differences in formulae (3.19). Then for the analysis made above to be valid for the approximations to the exact solution of system (2.2) we can require

$$\tilde{z}_{k+1} - \bar{z}_{k+1}(N) = O(\tau_k^{s+2}), \tag{3.20a}$$

$$\hat{z}_{k+1} - \check{z}_{k+1}(N) = O(\tau_k^{s+2}). \tag{3.20b}$$

Theorem 3 and formula (2.10) give the following condition to guarantee (3.20) for the RKN method:

$$N_0 \geq \log_2(s+2), \tag{3.21}$$

where s is the order of the underlying RK formula. Theorem 4 and formula (2.11) give the same condition for the RKmN method:

$$N_0 \geq s+1. \tag{3.22}$$

Unfortunately, estimate (3.20) has the quite complicated form for the RKSI method (theorem 2):

$$N_0 \geq (s+2)\ln(\tau_k)/\ln(d), \tag{3.23}$$

and it is useless practically because we cannot find the constant d exactly. Moreover, from (3.23) we have that $N \to \infty$ as $d \to 1$.

Hence, we can use the approximate solution found by the RKSI, RKN, or RKmN method to estimate the local error if the number of iterations is sufficiently large such that it satisfies (3.23), (3.21) or (3.22), respectively.

Thus, both relations (3.13) and (3.18) give us the practical way to estimate the local error of the methods constructed above. The next important stage in the numerical integration of DAEs (2.1) with step size control is choosing the following step size depending on this local error. Some of such strategies for the step size control are presented in [6]. We will use one of them that allows to choose the maximum step size for a given error tolerance.

Let ϵ be a given limit for the local error. From (3.13) for the local error e_{k+1} we have

$$e_{k+1} \cong \big(\psi(x_k, y_k), \phi(x_k, y_k)\big)^T \tau_k^{s+1}.$$

If $\|e_{k+1}\| \geq \epsilon$ then the numerical method has not reached the given precision and the step has to be recomputed. Then we compute the new step size τ_k^* by the relation

$$\tau_k^* = \vartheta \tau_k, \tag{3.24}$$

where ϑ satisfies the equation

$$\left\| \big(\psi(x_k, y_k), \phi(x_k, y_k)\big)^T \right\| (\vartheta \tau_k)^{s+1} = \epsilon. \tag{3.25}$$

From (3.24) and (3.25) we obtain

$$\vartheta^{s+1} = \epsilon / \|e_{k+1}\|$$

or

$$\vartheta = \big(\epsilon / \|e_{k+1}\|\big)^{\frac{1}{s+1}}. \tag{3.26}$$

And we compute the new approximation $\hat{z}_{k+1}$ at the point $t_{k+1} = t_k + \tau_k^*$.

Besides, if the norm of the local error does not exceed ϵ for the original step size τ_k then we consider that the approximate solution at the point $t_{k+1} = t_k + \tau_k$ satisfies the given tolerance and we take t_{k+1} as the next point in the numerical integration. After that we make the next step τ_k^* from the point t_{k+1} (see fig. 2).

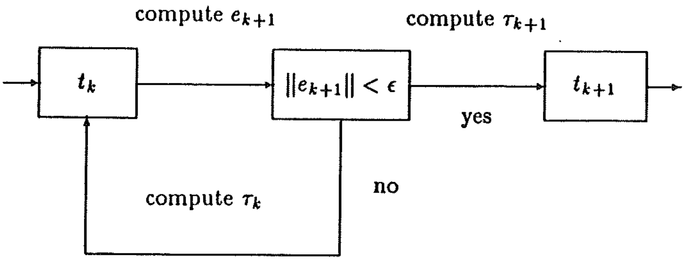

Figure 2: The procedure of choosing the next step size.

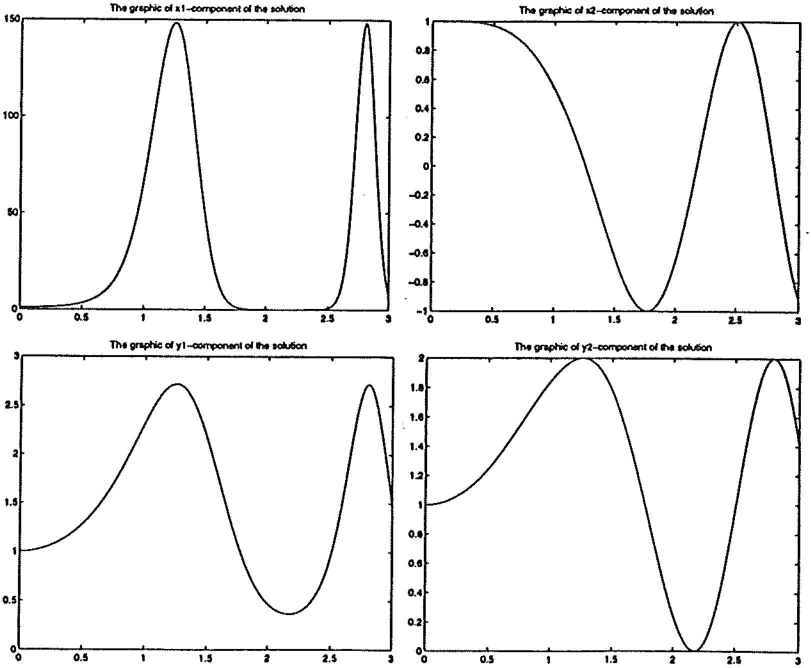

Figure 3: The pictures of the exact solution of problem (4.1).

4 Numerical experiments

We will test the methods constructed in Sections 2 and 3 on the following problem:

$$x_1'(t) = 10t \cdot \exp\Big(5\big(y_2(t) - 1\big)\Big) \cdot x_2(t), \tag{4.1a}$$

$$x_2'(t) = -2t \cdot \ln\big(y_1(t)\big), \tag{4.1b}$$

$$y_1(t) = x_1(t)^{\frac{1}{5}}, \tag{4.1c}$$

$$y_2(t) = \big(x_2(t)^2 + y_2(t)^2\big)/2, \tag{4.1d}$$

where $t \in [t_0, t_0 + T]$ and where the initial conditions defined by the equalities:

$$x_1(t_0) = \exp\big(5\sin(t_0^2)\big), \quad x_2(t_0) = \cos(t_0^2), \tag{4.2a}$$

$$y_1(t_0) = \exp\big(\sin(t_0^2)\big), \quad y_2(t_0) = \sin(t_0^2) + 1. \tag{4.2b}$$

Problem (4.1) has the exact solution (see fig. 3):

$$x_1(t) = \exp\big(5\sin(t^2)\big), \quad x_2(t) = \cos(t^2), \tag{4.3a}$$

$$y_1(t) = \exp\big(sin(t^2)\big), \quad y_2(t) = \sin(t^2) + 1. \tag{4.3b}$$

From estimate (3.23) it follows that the simple iteration is not a good iterative process for constructing the combined numerical variable step size methods because we cannot really estimate the number of iterations needed for convergence of the maximal order. So we will apply the variable step size RKN and RKmN methods only. They are based on the equations (3.15) and (3.16).

Now we solve the problem (4.1) in the interval [1.0708712, 1.4123836] by these methods for various combinations of number of iterations (N) and error tolerance (ϵ). For the given interval all of the assumptions of Section 2 are satisfied, and hence the theoretical results of the paper are valid.

Tables 1 and 2 contain global errors of the RKN methods of orders four and six, respectively. Tables 3 and 4 contain the similar data for the RKmN methods of the same orders. We have used the extrapolation for estimating the local error in these four cases, i.e., formulae (3.13).

Tables 1–4 confirm that the global errors of the numerical variable step size methods are well consistent with the given tolerances. Thus, these methods allow computation of the approximate solution of DAEs (1.1) to high precision.

Table 1: Global errors of RKN method of order four using the extrapolation for estimating the local error

N	ϵ				
	10^{-5}	10^{-8}	10^{-11}	10^{-14}	10^{-17}
2	$3.646 \cdot 10^{-5}$	$1.149 \cdot 10^{-7}$	$6.286 \cdot 10^{-10}$	$1.196 \cdot 10^{-12}$	$3.567 \cdot 10^{-15}$
3	$1.971 \cdot 10^{-7}$	$5.932 \cdot 10^{-11}$	$1.861 \cdot 10^{-14}$	$5.412 \cdot 10^{-16}$	$2.692 \cdot 10^{-15}$
5	$1.972 \cdot 10^{-7}$	$5.929 \cdot 10^{-11}$	$1.878 \cdot 10^{-14}$	$3.053 \cdot 10^{-16}$	$4.441 \cdot 10^{-15}$
7	$1.972 \cdot 10^{-7}$	$5.929 \cdot 10^{-11}$	$1.878 \cdot 10^{-14}$	$3.053 \cdot 10^{-16}$	$4.441 \cdot 10^{-15}$

Table 2: Global errors of RKN method of order six using the extrapolation for estimating the local error

N	ϵ				
	10^{-5}	10^{-8}	10^{-11}	10^{-14}	10^{-17}
2	$1.033 \cdot 10^{-4}$	$5.096 \cdot 10^{-7}$	$2.178 \cdot 10^{-9}$	$7.955 \cdot 10^{-12}$	$1.216 \cdot 10^{-14}$
3	$2.476 \cdot 10^{-7}$	$2.909 \cdot 10^{-10}$	$1.230 \cdot 10^{-13}$	$2.220 \cdot 10^{-16}$	$3.192 \cdot 10^{-16}$
5	$1.560 \cdot 10^{-7}$	$1.745 \cdot 10^{-10}$	$9.169 \cdot 10^{-14}$	$2.359 \cdot 10^{-16}$	$3.053 \cdot 10^{-16}$
7	$1.560 \cdot 10^{-7}$	$1.745 \cdot 10^{-10}$	$9.165 \cdot 10^{-14}$	$2.082 \cdot 10^{-16}$	$3.192 \cdot 10^{-16}$

But tables 1–4 give no information about the efficiency of the variable step size methods concerning expenditures of computing time. We will obtain such information building tables of average step sizes which are ratios of the length of the interval of the numerical integration to numbers of steps needed to construct the approximate solution of problem (4.1) with the given tolerance. The information for the both RKN and RKmN methods of orders four and six is presented in tables 5–8, respectively.

The results of the numerical experiments show that we can get the maximum average step size for RKN method if the number of iterations $N \geq 3$ (see tables 5 and 6). Otherwise, the RKN methods are not effective. For example, if $N = 2$ then the average step sizes may be significantly smaller thereby increasing the expenditures of computing time. Moreover, for $N = 2$ the average step sizes are close values for both methods. It means that the use of RK formulae of

Table 3: Global errors of RKmN method of order four using the extrapolation for estimating the local error

N	ϵ				
	10^{-5}	10^{-8}	10^{-11}	10^{-14}	10^{-17}
2	$6.279 \cdot 10^{-4}$	$2.486 \cdot 10^{-6}$	$1.368 \cdot 10^{-8}$	$5.290 \cdot 10^{-11}$	$1.584 \cdot 10^{-13}$
3	$4.292 \cdot 10^{-5}$	$2.119 \cdot 10^{-7}$	$3.975 \cdot 10^{-10}$	$1.593 \cdot 10^{-12}$	$1.060 \cdot 10^{-14}$
5	$1.904 \cdot 10^{-7}$	$4.803 \cdot 10^{-11}$	$2.427 \cdot 10^{-14}$	$3.331 \cdot 10^{-16}$	$4.441 \cdot 10^{-15}$
7	$1.972 \cdot 10^{-7}$	$5.929 \cdot 10^{-11}$	$1.862 \cdot 10^{-14}$	$3.053 \cdot 10^{-16}$	$3.900 \cdot 10^{-15}$

Table 4: Global errors of RKmN method of order six using the extrapolation for estimating the local error

N	ϵ				
	10^{-5}	10^{-8}	10^{-11}	10^{-14}	10^{-17}
2	$1.575 \cdot 10^{-3}$	$8.588 \cdot 10^{-6}$	$4.805 \cdot 10^{-8}$	$2.043 \cdot 10^{-10}$	$5.180 \cdot 10^{-13}$
3	$1.220 \cdot 10^{-4}$	$4.970 \cdot 10^{-7}$	$2.038 \cdot 10^{-9}$	$8.180 \cdot 10^{-12}$	$3.311 \cdot 10^{-14}$
5	$1.018 \cdot 10^{-5}$	$1.569 \cdot 10^{-8}$	$3.933 \cdot 10^{-11}$	$1.235 \cdot 10^{-13}$	$6.523 \cdot 10^{-16}$
7	$2.285 \cdot 10^{-7}$	$1.657 \cdot 10^{-10}$	$9.051 \cdot 10^{-14}$	$9.714 \cdot 10^{-17}$	$2.776 \cdot 10^{-16}$

Table 5: Average step sizes of RKN method of order four using the extrapolation for estimating the local error

N	ϵ				
	10^{-5}	10^{-8}	10^{-11}	10^{-14}	10^{-17}
2	$1.797 \cdot 10^{-2}$	$5.336 \cdot 10^{-3}$	$1.472 \cdot 10^{-3}$	$3.561 \cdot 10^{-4}$	$6.486 \cdot 10^{-5}$
3	$2.439 \cdot 10^{-2}$	$8.330 \cdot 10^{-3}$	$2.457 \cdot 10^{-3}$	$6.593 \cdot 10^{-4}$	$1.684 \cdot 10^{-4}$
5	$2.439 \cdot 10^{-2}$	$8.330 \cdot 10^{-3}$	$2.457 \cdot 10^{-3}$	$6.593 \cdot 10^{-4}$	$1.685 \cdot 10^{-4}$
7	$2.439 \cdot 10^{-2}$	$8.330 \cdot 10^{-3}$	$2.457 \cdot 10^{-3}$	$6.593 \cdot 10^{-4}$	$1.685 \cdot 10^{-4}$

Table 6: Average step sizes of RKN method of order six using the extrapolation for estimating the local error

N	ϵ				
	10^{-5}	10^{-8}	10^{-11}	10^{-14}	10^{-17}
2	$2.134 \cdot 10^{-2}$	$6.568 \cdot 10^{-3}$	$1.856 \cdot 10^{-3}$	$4.659 \cdot 10^{-4}$	$8.786 \cdot 10^{-5}$
3	$3.415 \cdot 10^{-2}$	$2.009 \cdot 10^{-2}$	$1.035 \cdot 10^{-2}$	$4.949 \cdot 10^{-3}$	$2.009 \cdot 10^{-3}$
5	$3.415 \cdot 10^{-2}$	$2.009 \cdot 10^{-2}$	$1.035 \cdot 10^{-2}$	$4.949 \cdot 10^{-3}$	$1.997 \cdot 10^{-3}$
7	$3.415 \cdot 10^{-2}$	$2.009 \cdot 10^{-2}$	$1.035 \cdot 10^{-2}$	$4.949 \cdot 10^{-3}$	$1.997 \cdot 10^{-3}$

Table 7: Average step sizes of RKmN method of order four using the extrapolation for estimating the local error

N	ϵ				
	10^{-5}	10^{-8}	10^{-11}	10^{-14}	10^{-17}
2	$7.762 \cdot 10^{-3}$	$1.447 \cdot 10^{-3}$	$2.613 \cdot 10^{-4}$	$3.754 \cdot 10^{-5}$	$4.069 \cdot 10^{-6}$
3	$2.009 \cdot 10^{-2}$	$5.991 \cdot 10^{-3}$	$1.626 \cdot 10^{-3}$	$4.206 \cdot 10^{-4}$	$9.500 \cdot 10^{-5}$
5	$2.439 \cdot 10^{-2}$	$8.330 \cdot 10^{-3}$	$2.457 \cdot 10^{-3}$	$6.593 \cdot 10^{-4}$	$1.686 \cdot 10^{-4}$
7	$2.439 \cdot 10^{-2}$	$8.330 \cdot 10^{-3}$	$2.457 \cdot 10^{-3}$	$6.593 \cdot 10^{-4}$	$1.686 \cdot 10^{-4}$

Table 8: Average step sizes of RKmN method of order six using the extrapolation for estimating the local error

N	ϵ				
	10^{-5}	10^{-8}	10^{-11}	10^{-14}	10^{-17}
2	$9.757 \cdot 10^{-3}$	$1.940 \cdot 10^{-3}$	$3.561 \cdot 10^{-4}$	$5.435 \cdot 10^{-5}$	$6.089 \cdot 10^{-6}$
3	$2.277 \cdot 10^{-2}$	$6.970 \cdot 10^{-3}$	$2.009 \cdot 10^{-3}$	$5.278 \cdot 10^{-4}$	$1.236 \cdot 10^{-4}$
5	$3.415 \cdot 10^{-2}$	$1.797 \cdot 10^{-2}$	$9.230 \cdot 10^{-3}$	$4.269 \cdot 10^{-3}$	$1.691 \cdot 10^{-3}$
7	$3.415 \cdot 10^{-2}$	$2.009 \cdot 10^{-2}$	$1.035 \cdot 10^{-2}$	$4.949 \cdot 10^{-3}$	$1.997 \cdot 10^{-3}$

high orders for small N is useless because it does not allow to increase step sizes in the numerical integration. So we can conclude that the numerical results confirm formula (3.21).

Tables 7 and 8 give the same result for the RKmN methods. The maximum precision and the optimal average step size take place if $N \geq 5$ for the method of order four and $N \geq 7$ for the method of order six. If the number of iterations is less then the average step size is decreased in many times. This is a good illustration for formula (3.22).

The last four tables give us the global errors and the average step sizes for the both RKN and RKmN methods in case if the local error have been computed by formula (3.18). We see that the the global errors are minimal if $N \geq 3$ for the RKN method and $N \geq 5$ for the RKmN method (see tables 9 and 10). Tables 11 and 12 allow to do the similar conclusion for the average step sizes that also

Table 9: Global errors of RKN method of order four using two RK formulae of different orders of convergence for estimating the local error

N	ϵ				
	10^{-5}	10^{-8}	10^{-11}	10^{-14}	10^{-17}
2	$7.127 \cdot 10^{-5}$	$2.203 \cdot 10^{-7}$	$8.751 \cdot 10^{-10}$	$3.509 \cdot 10^{-12}$	$5.677 \cdot 10^{-14}$
3	$1.318 \cdot 10^{-8}$	$3.519 \cdot 10^{-12}$	$1.416 \cdot 10^{-15}$	$3.608 \cdot 10^{-16}$	$1.790 \cdot 10^{-15}$
5	$1.485 \cdot 10^{-8}$	$3.552 \cdot 10^{-12}$	$1.547 \cdot 10^{-15}$	$4.718 \cdot 10^{-16}$	$3.539 \cdot 10^{-15}$
7	$1.485 \cdot 10^{-8}$	$3.552 \cdot 10^{-12}$	$1.547 \cdot 10^{-15}$	$4.718 \cdot 10^{-16}$	$3.567 \cdot 10^{-15}$

Table 10: Global errors of RKmN method of order four using two RK formulae of different orders of convergence for estimating the local error

N	ϵ				
	10^{-5}	10^{-8}	10^{-11}	10^{-14}	10^{-17}
2	$2.524 \cdot 10^{-3}$	$1.853 \cdot 10^{-5}$	$1.136 \cdot 10^{-7}$	$6.485 \cdot 10^{-10}$	$3.699 \cdot 10^{-12}$
3	$3.870 \cdot 10^{-5}$	$1.315 \cdot 10^{-7}$	$5.496 \cdot 10^{-10}$	$4.134 \cdot 10^{-12}$	$1.707 \cdot 10^{-14}$
5	$2.074 \cdot 10^{-7}$	$8.282 \cdot 10^{-11}$	$3.181 \cdot 10^{-14}$	$9.437 \cdot 10^{-16}$	$4.122 \cdot 10^{-15}$
7	$1.492 \cdot 10^{-8}$	$3.553 \cdot 10^{-12}$	$1.540 \cdot 10^{-15}$	$4.718 \cdot 10^{-16}$	$3.428 \cdot 10^{-15}$

Table 11: Average step sizes of RKN method of order four using two RK formulae of different orders of convergence for estimating the local error

N	ϵ				
	10^{-5}	10^{-8}	10^{-11}	10^{-14}	10^{-17}
2	$1.265 \cdot 10^{-2}$	$3.557 \cdot 10^{-3}$	$9.513 \cdot 10^{-4}$	$2.142 \cdot 10^{-4}$	$7.418 \cdot 10^{-5}$
3	$1.797 \cdot 10^{-2}$	$5.336 \cdot 10^{-3}$	$1.472 \cdot 10^{-3}$	$3.842 \cdot 10^{-4}$	$9.080 \cdot 10^{-5}$
5	$1.797 \cdot 10^{-2}$	$5.336 \cdot 10^{-3}$	$1.472 \cdot 10^{-3}$	$3.842 \cdot 10^{-4}$	$9.158 \cdot 10^{-5}$
7	$1.797 \cdot 10^{-2}$	$5.336 \cdot 10^{-3}$	$1.472 \cdot 10^{-3}$	$3.842 \cdot 10^{-4}$	$9.148 \cdot 10^{-5}$

Table 12: Average step sizes of RKmN method of order four using two RK formulae of different orders of convergence for estimating the local error

N	ϵ				
	10^{-5}	10^{-8}	10^{-11}	10^{-14}	10^{-17}
2	$7.589 \cdot 10^{-3}$	$1.626 \cdot 10^{-3}$	$3.125 \cdot 10^{-4}$	$5.650 \cdot 10^{-5}$	$1.013 \cdot 10^{-6}$
3	$1.265 \cdot 10^{-2}$	$3.557 \cdot 10^{-3}$	$9.566 \cdot 10^{-4}$	$2.219 \cdot 10^{-4}$	$7.318 \cdot 10^{-5}$
5	$1.797 \cdot 10^{-2}$	$5.336 \cdot 10^{-3}$	$1.472 \cdot 10^{-3}$	$3.842 \cdot 10^{-4}$	$9.144 \cdot 10^{-5}$
7	$1.797 \cdot 10^{-2}$	$5.336 \cdot 10^{-3}$	$1.472 \cdot 10^{-3}$	$3.842 \cdot 10^{-4}$	$9.134 \cdot 10^{-5}$

confirms the formulae (3.21) and (3.22).

5 Practical implementation of numerical variable step size methods

Thus, in the paper we have obtained and substantiated three classes of the stable numerical methods for solving problem (1.1) using variable step size and examined two strategies for the step size control. The same methods as have been used for DAEs may be applied to ODEs. In this case in formulae (2.3)-(2.5) it is necessary for the dimension of the y-components to be equal to zero. All results of the paper can be easily transfered on the case of ODEs.

The implicit numerical variable step size methods obtained in Section 3 for solving DAEs (1.1) are also applicable to ODEs. Moreover, the estimates for the optimal numbers of iterations which are analogous to (3.21)-(3.23) for ODEs have the form:

$$N_0 \geq s+1 \qquad \text{for the RKSI method,}$$

$$N_0 \geq (\log_2(s+3))/2-1 \qquad \text{for the RKN method,}$$

$$N_0 \geq s/2 \qquad \text{for the RKmN method.}$$

Thus, the constructed numerical methods allow us to solve both ODEs and DAEs of the form (1.1) quite effectively. But the numerical integration by implicit variable step size methods requires considerable expenditures of computing time because the Richardson extrapolation implies that we must solve problem (2.2) at least three times per step and the way based on RK formulae of different orders implies that we must solve (2.2) at least two times. We will give some recommendations how to reduce the total number of arithmetical operations and consequently to shorten computing time expenditures.

Since the main expenditures of computing time are used in the calculation of the inverse matrices $\partial\bar{F}_k^\tau(Z)^{-1}$ we will try either to reduce the dimension of inversed matrices or to modify methods of inversing of matrices in order to reduce total number of arithmetical operations or to decrease the number of inversions.

To solve problem (2.2) it is not necessary to apply the iterative process to all equations at the same time. We can notice that equations (2.2a,b) do not depend on equations (2.2c,d). So we solve system (2.2a,b) consisting of $(m+n)l$ equations. Then we find the solution of system (2.2c) being a known function of the solution of system (2.2a,b). After that we solve the last system (2.2d) depending on the solution of system (2.2c) and consisting in the general case of n equations. Thus, we have replaced each inversion of the square matrix of dimension $(m+n)(l+1)$ by inversion of two matrices of dimensions $(m+n)l$ and n, respectively, that requires roughly $(1+\frac{1}{l})^3$ times less operations

Table 13: Expenditures of computing time for numerical methods using the extrapolation for estimating the local error

Method	Order	Expenditures of computing time (sec.)		
		Full Gauss for one system	Full Gauss for two systems	Modified Gauss for two systems
RKN	4	1007	373	274
RKN	6	555	204	121
RKmN	4	9401	1059	901
RKmN	6	12487	1689	1234

of multiplication and division[4]. Hence, for equations (3.15) expenditures in computing time can be reduced approximately 1.95 times and for equations (3.16) 1.59 times. The numerical experiments on two methods (3.15) and (3.16) confirm the strength of the above given estimates in reducing the expenditures in computing time (see table 13).

Another way for reducing the number of arithmetical operations is obtained by using the structure of the matrices $\partial\bar{F}_k^T(Z)$. When carrying out the numerical experiments in Section 4 we used the Gauss method and chose the pivot element of the matrix. This method gives high precision and requires $(m+n)^3l^3$ operations of multiplication and division for inversing a matrix of dimension $(m+n)l \times (m+n)l$ and approximately $(m+n)^3l^3/3$ operations for solving a system of linear algebraic equations with a matrix of the same dimension. We will now consider a modification of the Gauss method allowing to keep the high precision and to reduce the number of arithmetical operations for the matrices of the form $\partial\bar{F}_k^T(Z)$. For these purposes we will change the choice of the pivot element of the matrix. We will choose the pivot element among some subset of elements of the active submatrix only. This strategy of choosing the pivot element will be more effective and it will permit keeping the same structure of matrix $\partial\bar{F}_k^T(Z)$.

We can see that the matrix $\partial\bar{F}_k^T(Z)$ has the following block structure:

$$\partial\bar{F}_k^T(Z) = \begin{pmatrix} \partial\bar{F}_k^T(Z)_1 \\ \partial\bar{F}_k^T(Z)_2 \\ \cdots \\ \partial\bar{F}_k^T(Z)_l \end{pmatrix}, \tag{5.1}$$

where each block $\partial\bar{F}_k^T(Z)_i, i = 1, 2, ..., l$, of dimension $(m+n) \times (m+n)l$ has the block structure

$$\partial\bar{F}_k^T(Z)_i = (A, B, C, D),$$

[4] Here and later we will carry out all the calculations on the number of operations of multiplication and division only because they require more time than operations of addition and subtraction do

and A is the matrix of dimension $(m+n)\times(m+n)(i-1)$

$$A=\begin{pmatrix} O(\tau) & \cdots & O(\tau)\\ \vdots & \ddots & \vdots\\ O(\tau) & \cdots & O(\tau)\\ 0 & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & 0 \end{pmatrix},$$

B is the matrix of dimension $(m+n)\times m$

$$B=\begin{pmatrix} 1+O(\tau) & O(\tau) & \cdots & O(\tau) & O(\tau)\\ O(\tau) & 1+O(\tau) & \cdots & O(\tau) & O(\tau)\\ \vdots & & \ddots & & \vdots\\ O(\tau) & O(\tau) & \cdots & 1+O(\tau) & O(\tau)\\ O(\tau) & O(\tau) & \cdots & O(\tau) & 1+O(\tau)\\ x & & \cdots & & x\\ \vdots & & \ddots & & \vdots\\ x & & \cdots & & x \end{pmatrix},$$

C is the matrix of dimension $(m+n)\times n$

$$C=\begin{pmatrix} O(\tau) & \cdots & O(\tau)\\ \vdots & \ddots & \vdots\\ O(\tau) & \cdots & O(\tau)\\ x & \cdots & x\\ \vdots & \ddots & \vdots\\ x & \cdots & x \end{pmatrix},$$

and D is the matrix of dimension $(m+n)\times(m+n)(l-i)$

$$D=\begin{pmatrix} O(\tau) & \cdots & O(\tau)\\ \vdots & \ddots & \vdots\\ O(\tau) & \cdots & O(\tau)\\ 0 & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & 0 \end{pmatrix}.$$

Here x means in the general case a nontrivial element. According to the Gauss method we will reduce the first block $\partial\bar{F}_k^\tau(Z)_1$ to the upper triangular form with the unitary diagonal.

Let us consider the first m rows of the block $\partial\bar{F}_k^\tau(Z)_1$: Since the square matrix of dimension m at the upper left corner of $\partial\bar{F}_k^\tau(Z)_1$ satisfies the condition of diagonal dominance so it is nonsingular if τ is small enough. Thus, choosing the pivot element we can limit ourselves by viewing the elements of this matrix only in the first m steps of the elimination. Besides, taking into account the diagonal dominance of the first m rows of the matrix $\partial\bar{F}_k^\tau(Z)_1$ on the first m steps of the Gauss method it is enough to make sure that the modulus of the diagonal element is larger than some ϵ. We choose the pivot element only if this condition does not hold.

Making the first step of the Gauss elimination we can take into account that $n(l-1)$ elements in the first column are already equal to zero. Thus, using the structure of the matrix $\partial\bar{F}_k^\tau(Z)$ we must make only $ml+n-1$ subtractions of rows. Then repeating the elimination $m-1$ times we convert the first m rows of the block $\partial\bar{F}_k^\tau(Z)_1$ to the triangular form and make m normalizations and $m(l-1)+n+\sum_{i=1}^{m}(m-i)$ subtractions of the rows multiplied by a coefficient.

Let us now consider the last n rows of the block $\partial\bar{F}_k^\tau(Z)_1$. After m steps of the Gauss method all elements of the first m columns are equal to zero and the elements of the next n columns form a matrix of the dimension $n\times n$ of the form $I_n-\partial f_y(z)+P(z)$. Since the nonsingularity condition (II) hold and $P(z)=O(\tau)$ then a τ exists such that the matrix $I_n-\partial f_y(z)+P(z)$ is nonsingular. Hence, the Gauss elimination and the choice of the pivot element among the elements of this matrix allow to obtain the upper triangular form for the last n rows of the matrix $\partial\bar{F}_k^\tau(Z)_1$. It requires n normalizations and $m(l-1)+n+\sum_{i=1}^{n}(n-i)$ subtractions of the rows multiplied by a coefficient.

So, we have converted the first block of the matrix $\partial\bar{F}_k^\tau(Z)$ to the required form. Repeating the given algorithm for the block $\partial\bar{F}_k^\tau(Z)_2$, and using the fact that the first $m+n$ columns of this block contain the zero elements only we change the second block of the matrix $\partial\bar{F}_k^\tau(Z)$ to the triangular form. After $l-2$ repetitions of the algorithm the remaining blocks $\partial\bar{F}_k^\tau(Z)_i, i=3,...,l$, are transformed to the upper triangular form. Thus, we obtain the LU-factorization of the matrix $\partial\bar{F}_k^\tau(Z)$.

It is not too difficult to verify that this algorithm requires the following number of operations of multiplication and division with an accuracy up to the members of lower order (see appendix A):

$$\frac{(m+n)^3l^3}{3}-\frac{n(m+n)^2l^3}{3}. \tag{5.2}$$

Thus, from (5.2) it follows that taking into account the structure of the matrix $\partial\bar{F}_k^\tau(Z)$ for solving a linear algebraic equations with this matrix we can decrease the number of arithmetic operations approximately $1+\frac{m}{n}$ times. For the RKN methods based on equations (3.15) and (3.16) it reduces the computing time expenditures two times (see table 13).

Taking into consideration that the structure of the matrix $\partial\bar{F}_k^\tau(Z)$ does not influence the back substitution with an accuracy up to the members of low order in l, m and n we obtain the number of operations of multiplication and division needed for an inversion of this matrix [18, p.69]:

$$(m+n)^3l^3 - \frac{n(m+n)^2l^3}{3}. \tag{5.3}$$

From (5.3) it follows that using the structure of the matrix $\partial \bar{F}_k^T(Z)$ for the inversion we can decrease the number of operations of multiplication and division approximately $(3m+3n)/(3m+2n)$ times. For the RKmN methods based on equations (3.15) and (3.16) it reduces the computing time expenditures 1.2 times (see table 13).

The last recommendation allowing to decrease the total number of arithmetical operations is referred to the numerical variable step size methods, which use RK formulae of different orders to estimate the local error. If these methods were constructed using embedded RK formulae [6] then we have to solve system (2.2a,b) one time and system (2.2c,d) two times to define the local error. Thus, it is possible to reduce the computing time by a factor of approximately two.

References

[1] V.O. Belash, A.L. Glebov, N.Ya. Mar'yashkin, Ye.E. Ovchinnikov, *The solution of differential-algebraic systems for the circuit analysis of large integrated circuits*, VTs Akad. Nauk S.S.S.R., Moscow, 1991.

[2] Yu.Ye. Boyarintsev, V.A. Danilov, A.A. Loginov, V.F. Chistyakov, *Numerical methods of solving singular systems*, Nauka, Novosibirsk, 1989.

[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] C.W. Gear, *Differential-algebraic equations index transformations*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 39-47.

[5] A.C. Guyton, T.G. Coleman, H.J. Gardner, *Circulation: overall regulation*, Ann. Rev. Phisiol., 34 (1972), pp. 13-41.

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

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

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

[9] N. Ikeda, F. Muramo, M. Shiratare, T. Sato, *A model of overall regulation of body fluids*, Ann. Biomed. Eng., 7 (1979), pp. 135-166.

[10] 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.

[11] 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., 1 (1992), pp. 14-19.

[12] 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., 3 (1993), pp. 6-10.

[13] 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), pp. 522-540.

[14] G.Yu. Kulikov, *The practical implementation and efficiency of numerical methods for solving the autonomous Cauchy problem with an algebraic relation between the phase variables*, Zh. Vychisl. Mat. Mat. Fiz., 34 (1994), pp. 1617-1631.

[15] G.Yu. Kulikov, *The numerical solution of the Cauchy problem with algebraic constrains on the phase variables (with applications in medical cybernetics)*, The dissertation of candidate of sciences in mathematics, Computational Center of Russian Academy of Sciences, Moscow, 1994.

[16] 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.

[17] L.R. Petzold, P. Lotstedt, *Numerical solution of nonlinear differential equations with algebraic constrains II: Practical implications*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 720-733.

[18] A.A. Samarskiy, A.V. Gulin, *Numerical methods*, Nauka, Moscow, 1989.

A Number of operations of multiplication and division for modified Gauss method

In this Section we will now calculate the number of operations of multiplication and division needed for transforming the matrix $\partial\bar{F}_k^{\tau}(Z)$ to the upper triangular form with the unitary diagonal. In our case the number of operations of division can be calculated in the same way as that for the ordinary Gauss method [18, p.53]. It is equal to

$$\sum_{i=1}^{(m+n)l}\left((m+n)l-i\right)=\frac{(m+n)l((m+n)l-1)}{2}. \qquad (A.1)$$

It is simple to calculate that in order to convert the first block $\partial\bar{F}_k^\tau(Z)_1$ to the required form we need

$$\sum_{i=1}^{m+n}\big(m(l-1)+m+n-i\big)\big((m+n)(l-1)+m+n-i\big)$$

multiplications. To factor the second block $\partial\bar{F}_k^\tau(Z)_2$ we need

$$\sum_{i=1}^{m+n}\big(m(l-2)+m+n-i\big)\big((m+n)(l-2)+m+n-i\big)$$

multiplications. Thus, to transform the j-th block $\partial\bar{F}_k^\tau(Z)_j, j=3,...,l$, we need

$$\sum_{i=1}^{m+n}\big(m(l-j)+m+n-i\big)\big((m+n)(l-j)+m+n-i\big) \qquad (A.2)$$

operations of multiplication. So from (A.2) for the LU-factorization of the matrix $\partial\bar{F}_k^\tau(Z)$ we have the following number of multiplications:

$$\sum_{j=1}^{l}\sum_{i=1}^{m+n}\big(m(l-j)+m+n-i\big)\big((m+n)(l-j)+m+n-i\big). \qquad (A.3)$$

Let us calculate (A.3). We can notice that

$$\begin{aligned}&\sum_{j=1}^{l}\sum_{i=1}^{m+n}\big(m(l-j)+m+n-i\big)\big((m+n)(l-j)+m+n-i\big)\\ &=\sum_{j=1}^{l}\sum_{i=1}^{m+n}\big((m+n)(l-j)+m+n-i\big)^2 \qquad (A.4)\\ &-\sum_{j=1}^{l}\sum_{i=1}^{m+n}n(l-j)\big((m+n)(l-j)+m+n-i\big).\end{aligned}$$

For the first item of the right side of formula (A.4) it is known [18, p.53] that

$$\begin{aligned}&\sum_{j=1}^{l}\sum_{i=1}^{m+n}\big((m+n)(l-j)+m+n-i\big)^2\\ &=\frac{\big((m+n)l-1\big)(m+n)l\big(2(m+n)l-1\big)}{6}. \qquad (A.5)\end{aligned}$$

Let us calculate the second item of the right side of (A.4). For any $j = 1, 2, ..., l$ from the formula for sum of an arithmetic progression we have

$$\begin{aligned}
&\sum_{i=1}^{m+n} n(l-j)((m+n)(l-j)+m+n-i) \\
&= n(l-j)\sum_{i=1}^{m+n}((m+n)(l-j)+m+n-i) \qquad (A.6) \\
&= \frac{n(m+n)}{2}(2(m+n)(l-j)^2+(m+n-1)(l-j)).
\end{aligned}$$

Substituting (A.6) into the second item of the right side of (A.4) we obtain

$$\begin{aligned}
&\sum_{j=1}^{l}\frac{n(m+n)}{2}(2(m+n)(l-j)^2+(m+n-1)(l-j)) \\
&= n(m+n)^2\sum_{j=1}^{l}(l-j)^2+\frac{n(m+n)(m+n-1)}{2}\sum_{j=1}^{l}(l-j) \qquad (A.7) \\
&= \frac{n(m+n)^2(l-1)l(2l-1)}{6}+\frac{n(m+n)(m+n-1)l(l-1)}{4}.
\end{aligned}$$

The forward substitution requires $(m+n)l$ operation of division and

$$\begin{aligned}
&\sum_{j=1}^{l}\sum_{i=1}^{m+n}(m(l-j)+m+n-i) = \sum_{j=1}^{l} m(m+n)(l-j) \\
&+\frac{1}{2}\sum_{j=1}^{l}(m+n)(m+n-1) = \frac{m(m+n)l(l-1)}{2} \\
&+\frac{(m+n)(m+n-1)l}{2} = \frac{(m+n)l(m(l-1)+m+n-1)}{2}
\end{aligned}$$

operations of multiplication. Summing these numbers we obtain

$$\frac{(m+n)l(m(l-1)+m+n+1)}{2}. \qquad (A.8)$$

Thus, using (A.1), (A.4), (A.5), (A.7) and (A.8) we have the following total number of operations multiplication and division needed for both the LU-factorization and the forward substitution taking into account the structure of the matrix $\partial\bar{F}_k^{\tau}(Z)$:

$$\begin{aligned}
&\frac{(m+n)l\big((m+n)l-1\big)}{2}+\frac{\big((m+n)l-1\big)(m+n)l\big(2(m+n)l-1\big)}{6}\\
&\qquad-\frac{n(m+n)^2(l-1)l(2l-1)}{6}-\frac{n(m+n)(m+n-1)l(l-1)}{4}\\
&\qquad\qquad+\frac{(m+n)l\big(m(l-1)+m+n+1\big)}{2}.
\end{aligned}\tag{A.9}$$

With an accuracy up to the members of lower order from (A.9) we obtain (5.2) and (5.3).