

DOI: 10.17516/1997-1397-2020-13-2-242-252

УДК 519.622

First-Order Methods With Extended Stability Regions for Solving Electric Circuit Problems

Mikhail V. Rybkov*

Lyudmila V. Knaub†

Danil V. Khorov‡

Siberian Federal University

Russian Federation

Received 16.01.2020, received in revised form 06.02.2020, accepted 25.03.2020

Abstract. Stability control of Runge-Kutta numerical schemes is studied to increase efficiency of integrating stiff problems. The implementation of the algorithm to determine coefficients of stability polynomials with the use of the GMP library is presented. Shape and size of the stability region of a method can be preassigned using proposed algorithm. Sets of first-order methods with extended stability domains are built. The results of electrical circuits simulation show the increase of the efficiency of the constructed first-order methods in comparison with methods of higher order.

Keywords: stiff problem, explicit methods, stability region, accuracy and stability control.

Citation: M.V.Rybkov, L.V.Knaub, D.V.Khorov, First-Order Methods With Extended Stability Regions for Solving Electric Circuit Problems, J. Sib. Fed. Univ. Math. Phys., 2020, 13(2), 242-252.

DOI: 10.17516/1997-1397-2020-13-2-242-252.

Introduction

Systems of ordinary differential equations (ODEs) describe various dynamic processes in chemistry, physics, etc. One of the areas where ODEs may be effectively applied is the electric circuit theory. Any changes in electric circuit lead to transient processes where some voltage swells, electromagnetic oscillations, extra currents may occur. They can damage electrical devices. At the same time transient processes occur in electrical generators and other electric circuits. Many electric circuits problems are described by stiff systems of ODEs.

In some cases explicit methods are required to solve initial value problems of stiff ODEs because L -stable methods involve inversion of the Jacobi matrix of a system. This defines overall computational costs [1–2]. At the same time explicit methods do not require the Jacobi matrix computation and they are more preferable to use for problems with not so high stiffness ratio.

At present time various explicit and implicit methods were developed [3]. The former are used on transition regions where the integration step is restricted by the accuracy criterion and there is no requirements for large stability interval. The latter are for the regions where large stability interval gives an opportunity to pass the integration interval in "several steps". Nevertheless these

*mixailrybkov@yandex.ru <https://orcid.org/0000-0002-6560-1435>

†lvknaub@yandex.ru <https://orcid.org/0000-0003-4857-2078>

‡danilkhorov@gmail.com <https://orcid.org/0000-0001-8967-8341>

© Siberian Federal University. All rights reserved

algorithms are not so efficient to solve high dimension systems of ODEs because of mentioned above reasons.

Variable order algorithms based on explicit schemes were developed [4]. They are applied to the regions where there is no need to use high-order methods. Low computational cost can be achieved by using there low-order methods with extended stability intervals which in fact play part of implicit methods from the point of view of the length of stability interval.

Low-order methods with large stability interval are needed to develop such algorithms. In addition the greater number of stages of a method (and therefore the higher the degree m of stability polynomial), the large the stability interval is. The stability polynomials of degree up to $m = 13$ were constructed [2]. The algorithm to determine the stability polynomial coefficients was developed such that the corresponding explicit Runge-Kutta methods have a predetermined shape and size of the stability region [7].

Here implementation of the algorithm to obtain the stability polynomial coefficients with the use of the library for arbitrary precision arithmetic GMP is presented. Set of the first-order methods with extended stability intervals is developed. Numerical simulation of Van der Pol oscillator shows that proposed algorithms are more efficient in comparison with the Merson method of fourth order of accuracy.

1. Explicit Runge-Kutta methods

We consider the Cauchy problem for the stiff system of ordinary differential equations

$$y' = f(t, y), \quad y(t_0) = y_0, \quad t_0 \leq t \leq t_k, \tag{1}$$

where y и f are real N -dimantional vector functions, t is independent variable. To solve (1) the following explicit Runge-Kutta methods were proposed [2]

$$y_{n+1} = y_n + \sum_{i=1}^m p_{mi}k_i, \quad k_i = hf\left(t_n + \alpha_i h, y_n + \sum_{j=1}^{i-1} \beta_{ij}k_j\right), \tag{2}$$

where k_i , $1 \leq i \leq m$, are stages of the method, h is an integration step, p_{mi} , α_i , β_{ij} , $1 \leq i \leq m$, $1 \leq j \leq i - 1$, are numerical coefficients that define stability and accuracy of scheme (2). For the sake of simplicity we consider the Cauchy problem for the autonomous system of ordinary differential equations

$$y' = f(y), \quad y(t_0) = y_0, \quad t_0 \leq t \leq t_k. \tag{3}$$

To solve (3) we can also write formulas (2) in the following form:

$$y_{n,i} = y_n + \sum_{j=1}^i \beta_{i+1,j}k_j, \quad 1 \leq i \leq m - 1, \quad y_{n+1} = y_n + \sum_{i=1}^m p_{mi}k_i, \tag{4}$$

where $k_i = hf(y_{n,i-1})$, $1 \leq i \leq m$, $y_{n,0} = y_n$. The results given below can be used for non-autonomous systems if we assume in (2) that

$$\alpha_1 = 0, \quad \alpha_i = \sum_{j=1}^{i-1} \beta_{ij}, \quad 2 \leq i \leq m. \tag{5}$$

Below we need matrix B_m with elements b_{ij} in the form [2]

$$\begin{aligned} b_{1i} &= 1, \quad 1 \leq i \leq m, \quad b_{ki} = 0, \quad 2 \leq k \leq m, \quad 1 \leq i \leq k-1, \\ b_{ki} &= \sum_{j=k-1}^{i-1} \beta_{ij} b_{k-1,j}, \quad 2 \leq k \leq m, \quad k \leq i \leq m, \end{aligned} \quad (6)$$

where β_{ij} are numerical coefficients of schemes (2) or (4).

The stability of one-step methods is usually investigated by applying a Runge-Kutta method to a linear scalar equation known as Dahlquist's test equation

$$y' = \lambda y, \quad y(0) = y_0, \quad t \geq 0, \quad (7)$$

with complex λ , $Re(\lambda) < 0$. Variable λ is considered as a certain eigenvalue of the Jacobi matrix of problems (1) or (3). Applying numerical scheme (4) to Dahlquist's equation we get

$$y_{n+1} = Q_m(z)y_n, \quad z = h\lambda, \quad Q_m(z) = 1 + \sum_{i=1}^m c_{mi}z^i, \quad c_{mi} = \sum_{j=1}^m b_{ij}p_{mj}, \quad 1 \leq i \leq m. \quad (8)$$

Denoting $C_m = (c_{m1}, \dots, c_{mm})^T$ and $P_m = (p_{m1}, \dots, p_{mm})^T$, we can rewrite the latter equality (8) in the form

$$B_m P_m = C_m, \quad (9)$$

where the elements of matrix B_m are defined in (6). For internal numerical schemes (4) we have

$$y_{n,k} = Q_k(z)y_n, \quad Q_k(z) = 1 + \sum_{i=1}^k c_{ki}z^i, \quad c_{ki} = \sum_{j=1}^k b_{ij}\beta_{k+1,j}, \quad 1 \leq k \leq m-1. \quad (10)$$

Using designations $\beta_k = (\beta_{k+1,1}, \dots, \beta_{k+1,k})^T$ and $c_k = (c_{k1}, \dots, c_{kk})^T$ we obtain that coefficients β_{ij} of internal schemes (4) and the coefficients of corresponding stability polynomials are related by the equation

$$B_k \beta_k = c_k, \quad 1 \leq k \leq m-1. \quad (11)$$

It follows from (6) and (10) that $b_{ki} = c_{i-1,k-1}$, i.e., the elements of $(k+1)$ -th column of matrix B_m are equal to coefficients of stability polynomial $Q_k(z)$. Hence, if the coefficients of the stability polynomials of the basic and intermediate numerical schemes are defined then the coefficients of methods (4) are uniquely determined from linear systems (9) and (11) with upper triangular matrices B_i , $1 \leq i \leq m$.

Expansions of the exact and approximate solutions in the Taylor series in powers of h have the form

$$\begin{aligned} y(t_{n+1}) &= y(t_n) + hf + \frac{h^2}{2} f' f + O(h^3), \\ y_{n+1} &= y_n + \left(\sum_{j=1}^m b_{1j} p_{mj} \right) hf + \left(\sum_{j=2}^m b_{2j} p_{mj} \right) h^2 f' f_n + O(h^3), \end{aligned} \quad (12)$$

where the elementary differentials are taken with respect to exact $y(t_n)$ and approximate y_n solutions, respectively. Comparing relations (12) under assumption that $y(t_n) = y_n$, one can show that numerical scheme (4) has the first order of accuracy if $\sum_{j=1}^m b_{1j} p_{mj} = 1$. Hence, to design m -stage methods of the first order of accuracy it is necessary to set $c_{m1} = 1$.

2. Stability polynomials

Let two integer numbers k and m , $k \leq m$ be given. Consider the polynomial

$$Q_{m,k}(x) = 1 + \sum_{i=1}^k c_i x^i + \sum_{i=k+1}^m c_i x^i, \quad (13)$$

where the coefficients c_i , $1 \leq i \leq k$, are given and c_i , $k+1 \leq i \leq m$, are free coefficients. The coefficients c_i , $1 \leq i \leq k$, are usually defined from the approximation requirements. Therefore, for definiteness we assume below that $c_i = 1/i!$, $1 \leq i \leq k$.

Denote extremum points of (13) as x_1, \dots, x_{m-1} , where $x_1 > x_2 > \dots > x_{m-1}$. Unknown coefficients c_i , $k+1 \leq i \leq m$ can be obtained from the condition that polynomial (13) takes on predefined values at extremum points x_i , $k \leq i \leq m-1$, i.e.,

$$Q_{m,k}(x_i) = F_i, \quad k \leq i \leq m-1, \quad (14)$$

where $F(x)$ is a preassigned function, $F_i = F(x_i)$. For this purpose let us consider the algebraic system of equations with respect to variables x_i , $k \leq i \leq m-1$, and c_j , $k+1 \leq j \leq m$,

$$Q_{m,k}(x_i) = F_i, \quad Q'_{m,k}(x_i) = 0, \quad k \leq i \leq m-1, \quad Q'_{m,k} = \sum_{i=1}^m i c_i x^{i-1}. \quad (15)$$

We rewrite (15) in the form that is convenient for computations. Let us introduce vectors y , z , g and r with components

$$y_i = x_{k+i-1}, \quad z_i = c_{k+i}, \quad g_i = F_{k+i-1} - 1 - \sum_{j=1}^k c_j y_i^j, \quad r_i = - \sum_{j=1}^k j c_j y_i^{j-1}, \quad (16)$$

$$1 \leq i \leq m-k,$$

We also introduce diagonal matrices E_1, \dots, E_5 with elements on the main diagonal

$$e_1^{ii} = k+i, \quad e_2^{ii} = 1/y_i, \quad e_3^{ii} = \sum_{j=1}^k j c_j y_i^{j-1} + \sum_{j=1}^{m-k} (k+j) z_j y_i^{k+j-1},$$

$$e_4^{ii} = \sum_{j=2}^k j(j-1) c_j y_i^{j-2} + \sum_{j=1}^{m-k} (k+j)(k+j-1) z_j y_i^{k+j-2}, \quad (17)$$

$$e_5^{ii} = (-1)^{k+i-1}, \quad 1 \leq i \leq m-k.$$

Consider matrix A with elements $a^{ij} = y_i^{k+j}$, $1 \leq i, j \leq m-k$. The elements of vectors (16), matrices (17) and A depend on numbers m and k , where

$$g = g(y), \quad r = r(y), \quad E_2 = E_2(y), \quad E_3 = E_3(y, z), \quad E_4 = E_4(y, z), \quad A = A(y).$$

Then, we can rewrite problem (15) in the form

$$Az - g = 0, \quad E_2 A E_1 z - r = 0. \quad (18)$$

System (18) is ill-conditioned that leads to some difficulties in applying the fixed-point iterations for its solution. For convergence of the Newton method it is necessary to obtain good initial values but in this case it is difficult problem in its own right.

If we assume in (15) that $F_i = (-1)^i$, $k \leq i \leq m - 1$, we can find the polynomial with the maximal length of stability interval. In this case the problem of finding initial value y_0 is solved by using values of the Chebyshev polynomial at extremum points over interval $[-2m^2, 0]$, where m is the degree of polynomial (13). These values are

$$y_i = m^2[\cos(i\pi/m) - 1], \quad 1 \leq i \leq m - 1. \quad (19)$$

Substituting (19) in system (17), we obtain coefficients of the Chebyshev polynomial for which $|Q_{m1}(x)| \leq 1$ on $x \in [-2m^2, 0]$. Then for any k values given in (19) can be taken as the initial values, and according to numerical calculations there is good convergence rate in this case. If $F_i \neq (-1)^i$, $k \leq i \leq m - 1$, then the choice of initial values is quite a difficult problem.

Let us describe a way to solve (18) that does not require good initial values. One can apply relaxation to solve system (18). The main idea of relaxations is to solve unsteady-state problem which solution converges to the steady-state solution of the initial problem. Let us consider the Cauchy problem

$$y' = E_5(E_2AE_1A^{-1}g - r), \quad y(0) = y_0. \quad (20)$$

Apparently, upon finding the stationary point of (20), the coefficients of a stability polynomial can be determined from system (18). Let us notice that because matrix E_5 is used all eigenvalues of the Jacobi matrix of (20) have negative real parts, i.e., problem (7) is stable. It follows from numerical results that (20) is a stiff problem. Methods for solving such problems use calculation of the Jacobi matrix which cause difficulties in solving (20). Therefore, we apply the method of the second order of accuracy using numerical calculation and freezing the Jacobi matrix to solve (20) [5–6].

It can be shown that values of the polynomial coefficients tend to zero as m increases. Coefficients c_i , $k + 1 \leq i \leq m$, were calculated for the polynomial degree up to $m = 13$ using algorithm [2]. Moreover, the algorithm of obtaining polynomial coefficients on the interval $[-1, 1]$ was described in [7]. In this case coefficients c_i grow with slower rate and it is possible to construct polynomials of degrees $m > 13$.

3. Calculation of coefficients of stability polynomials using the GMP library

It is not difficult to see that coefficient c_m of stability polynomial (13) tends to zero as m increases and in particular if $m = 13$ and $k = 1$ the value of c_m is about 10^{-26} . Solution of problem (20) where $m > 13$ with double precision is very hard to realize because of round-off errors. In order to compute coefficients of polynomial of higher degrees m algorithm was implemented using *qd* library [8, 9].

The *qd* library allows one to perform computations with higher accuracy. Standard data type *double* which allows one to perform computations with double precision is restricted by 53 bites of binary mantissa and provides accuracy of 16 decimal digits. Whereas *qd* data type *dd_real* has 106-bit mantissa that provides accuracy of 32 decimal digits. In fact, the number of data type *dd_real* is a programmed concatenation of two *double* numbers, where mantissa becomes doubled but the range of values that can be represented using new data type stays the same (from 10^{-308} to 10^{308}). Despite this restriction accuracy of number representation increases.

With the use of this library the coefficients of polynomials up to degree $m = 35$ were computed [8]. Nevertheless, the *qd* library has some disadvantages. Firstly, accuracy of number

representation is restricted because of program implementation of data types. Secondly, it can be used only in Unix systems. Moreover, the *qd* library is written in the *C++* programming language. That is why numerical codes that use this library could be slower than codes written in low-level programming languages (for example, *C*).

Here we show numerical results of calculations of polynomial coefficients with the help of the library for arbitrary precision arithmetic *GMP*. This library provide accuracy of computations that is restricted only by the size of random access memory. It is cross-platform library, and it supports operations on integer, rational and real numbers. Besides, the *GMP* library is written in the *C* programming language which potentially increase the speed of computations.

Using the *GMP* library we obtain coefficients of polynomials up to degree $m = 40$. At higher degrees there are some difficulties that may be related to the choice of initial conditions for problem (20).

4. Construction of stability regions

Let us now describe the effect of function F on the size and shape of the stability region. If we assume $F_i = (-1)^i$, $k \leq i \leq m - 1$, than the stability interval length is $|\gamma_m| = 2m^2$. In this case, we have the maximum length of stability interval along the real axis for given m . The stability region of such methods is almost multiply connected which leads to the reduction of stability interval length because rounding errors may cause small imaginary parts of Jacobi matrix eigenvalues to appear. Fig. 1 shows the stability region of 5-stage method, where the stability interval length is $|\gamma_m| = 50$.

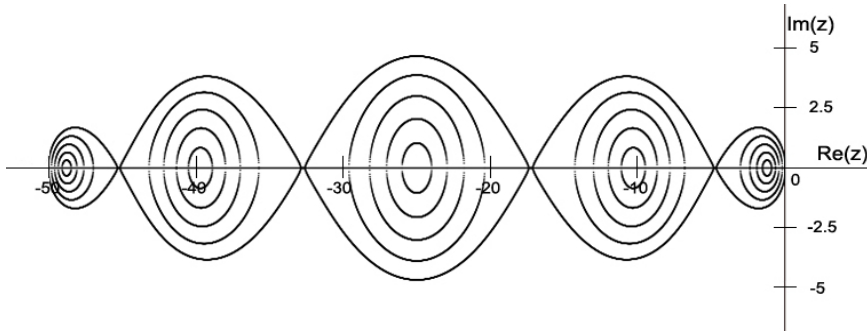


Fig. 1. Stability region at $m = 5$, $k = 1$, $F = \{-1, 1, -1, 1\}$, $|\gamma_m| = 50$

In order to avoid the stability region reduction because of rounding errors it should be "stretched" along the imaginary axis at the extremum points of the stability polynomial. For that we can assume $F_i = (-1)^i \mu$, $1 \leq i \leq m - 1$, $0 < \mu < 1$. For example, if we choose $\mu = 0.95$ then the stability interval length is reduced by only 3-4% in comparison with the maximal possible length that is equal to $2m^2$. Then it becomes equal to $|\gamma_m| = 48.39$ (Fig. 2). The stability region of the 5-stage method at $\mu = 0.8$ is shown in Fig. 3. In this situation, the stability interval length is reduced to $|\gamma_m| = 43.55$ with conjoined "stretching" along the imaginary axis. For better visualization of the roots of polynomial (13) level lines $|Q_{m,k}(x)| = 1$, $|Q_{m,k}(x)| = 0.8$, $|Q_{m,k}(x)| = 0.6$, $|Q_{m,k}(x)| = 0.4$, $|Q_{m,k}(x)| = 0.2$. in the complex plane Are shown in all figures.

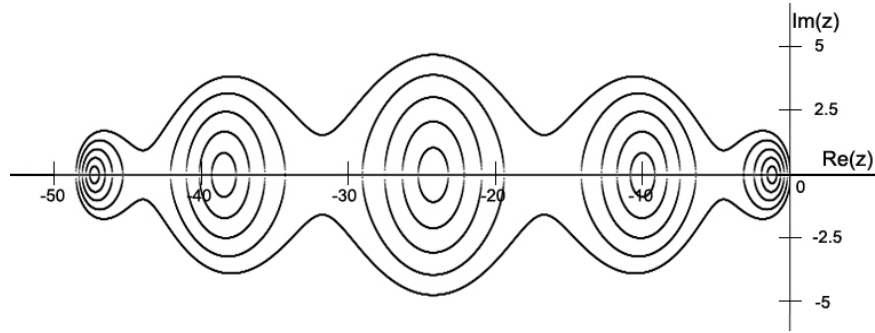


Fig. 2. Stability region at $m = 5$, $k = 1$, $F = \{-0.95, 0.95, 0.95, 0.95\}$, $|\gamma_m| = 48.39$

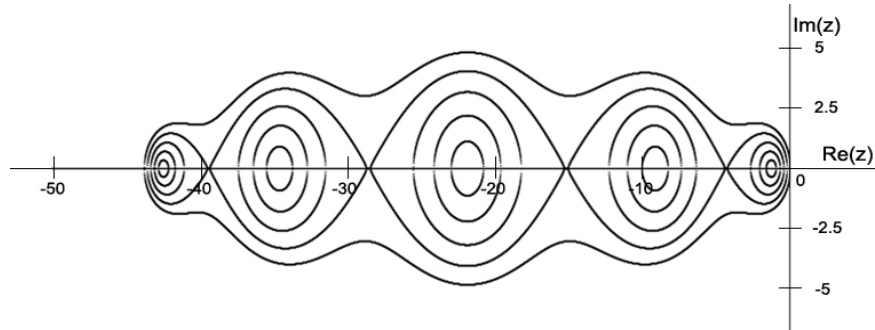


Fig. 3. Stability region at $m = 5$, $k = 1$, $F = \{-0.8, 0.8, -0.8, 0.8\}$, $|\gamma_m| = 43.55$

5. First-order method

For numerical solution of Cauchy problem (1) we consider the explicit five-stage Runge-Kutta method

$$\begin{aligned}
 y_{n+1} &= y_n + p_1 k_1 + p_2 k_2 + p_3 k_3 + p_4 k_4 + p_5 k_5, \\
 k_1 &= hf(y_n), \quad k_2 = hf(y_n + \beta_{21} k_1), \\
 k_3 &= hf(y_n + \beta_{31} k_1 + \beta_{32} k_2), \\
 k_4 &= hf(y_n + \beta_{41} k_1 + \beta_{42} k_2 + \beta_{43} k_3), \\
 k_5 &= hf(y_n + \beta_{51} k_1 + \beta_{52} k_2 + \beta_{53} k_3 + \beta_{54} k_4),
 \end{aligned} \tag{21}$$

where y and f are real N -dimensional vector functions, t is independent variable, h is the integration step, k_1, k_2, k_3, k_4 and k_5 are stages of the method, $p_1, p_2, p_3, p_4, p_5, \beta_{21}, \beta_{31}, \beta_{32}, \beta_{41}, \beta_{42}, \beta_{43}, \beta_{51}, \beta_{52}, \beta_{53}, \beta_{54}$ are numerical coefficients that define accuracy and stability of (21). Applying the algorithm, we obtain coefficients of the stability polynomial:

$$\begin{aligned}
 c_{5,1} &= 0.1e1, \quad c_{5,2} = 0.164341322127140896342e0, \quad c_{5,3} = 0.948975952580473808808e - 2, \\
 c_{5,4} &= 0.223956930863224544258e - 3, \quad c_{5,5} = 0.18509727522235334153e - 5
 \end{aligned}$$

In this case the stability interval length is $|\gamma_m| = 48.39$. Upon solving (9) and (11), we obtain the coefficients of the first-order method:

$$\begin{aligned} \beta_{21} &= 0.0413243016210550, & \beta_{31} &= 0.0805823881610573, & \beta_{32} &= 0.0805823881610573, \\ \beta_{41} &= 0.11916681511228434, & \beta_{42} &= 0.1597820013984078, & \beta_{43} &= 0.0819394878966193, \\ \beta_{51} &= 0.1570787892802991, & \beta_{52} &= 0.2379583021959820, & \beta_{53} &= 0.1631711307360486, \\ \beta_{54} &= 0.0822916178203657, \\ p_1 &= 0.1945277188657676, & p_2 &= 0.3151822878089125, & p_3 &= 0.2437005934695969, \\ p_4 &= 0.1641555613805598, & p_5 &= 0.0824338384751631. \end{aligned}$$

Accuracy control of numerical scheme is based on local error estimation [10].

The magnitude of

$$A'_n = [(0.5 - c_{m2})/\alpha_2](k_2 - k_1) \quad (22)$$

is used as preliminary estimation of local error. To we estimate the final accuracy the magnitude of

$$A''_n = (0.5 - c_{m2})(hf(y_{n+1}) - k_1) \quad (23)$$

is used. Thus, the following inequalities

$$A'_n \leq \epsilon, \quad A''_n \leq \epsilon. \quad (24)$$

are used for the accuracy control and for the choice of of integration step. As k_1 linearly depends on integration step then omission of inequality (24) leads to just one additional computation of the right hand side of the problem. If the step of integration is successful the second inequality (24) does not lead to the increase of computational cost because $f(y_{n+1})$ is not used at the next step. At the same time if the second inequality (24) is used for accuracy control the repeat computations in the case of violating accuracy criterion are quite expensive. Moreover, the greater m the higher computational cost is. Nevertheless, in most cases preliminary estimation of A'_n allows one to avoid repeat computations. The following inequality

$$\nu_n \leq \gamma_{m,1} \quad (25)$$

is used for stability control of method (2), where

$$\nu_n = \left| \alpha_2 \beta_{32} \right|^{-1} \max_{1 \leq j \leq N} \left| [\alpha_2 k_3 + \alpha_3 k_2 - (\alpha_2 + \alpha_3) k_1]_j / [k_2 - k_1]_j \right|, \quad (26)$$

and positive constants $\gamma_{m,1}$ define the size of stability regions [10].

6. Merson method

One of the most effective explicit fourth-order Runge-Kutta type methods is the Merson method [8]

$$\begin{aligned} y_{n+1} &= y_n + \frac{1}{6}k_1 + \frac{2}{3}k_4 + \frac{1}{6}k_5, \\ k_1 &= hf(y_n), \quad k_2 = hf\left(y_n + \frac{1}{3}k_1\right), \quad k_3 = hf\left(y_n + \frac{1}{6}k_1 + \frac{1}{6}k_2\right), \\ k_4 &= hf\left(y_n + \frac{1}{8}k_1 + \frac{3}{8}k_3\right), \quad k_5 = hf\left(y_n + \frac{1}{2}k_1 - \frac{3}{2}k_3 + 2k_4\right). \end{aligned} \quad (27)$$

The fifth computation of function f does not result in the fifth order of accuracy but allows one to extend the stability interval to 3.5 and estimate truncation error $\delta_{n,4}$ using stages k_i i.e.,

$$\delta_{n,4} = (2k_1 - 9k_3 + 8k_4 - 2k_5)/30.$$

We use inequality $\|\delta_{n,4}\| \leq 5\epsilon^{5/4}$ for accuracy control. Despite the fact that inequality for accuracy control is obtained with the help of a linear equation, it shows high reliability in solving non-linear problems.

Now let us construct the inequality for stability control. Applying to $k_3 - k_2$ the first order Taylor's formula with the remainder term written in the Lagrangian form, we have

$$k_3 - k_2 = h[\partial f(\mu_n)/\partial y](k_2 - k_1)/6,$$

where vector μ_n is calculated in some vicinity of solution $y(t_n)$. Taking into account that

$$k_2 - k_1 = h^2 f'_n f_n / 3 + O(h^3),$$

the inequality

$$\nu_{n,4} = 6 \cdot \max_{1 \leq j \leq N} \left| \frac{k_3^j - k_2^j}{k_2^j - k_1^j} \right| \leq 3.5 \quad (28)$$

can be used for stability control of (27), where 3.5 is the approximate length of stability interval. Let $\epsilon_{n,4} = \delta_{n,4}/5$. Then inequalities $\epsilon_{n,4} \leq 5\epsilon^{5/4}$ and $\nu_{n,4} \leq 3.5$. can be used for accuracy and stability control of scheme (27), respectively.

7. Numerical results

The computations were performed on Intel(R) Core(TM) i7-8550U CPU. However, coefficients of stability polynomial were computed with the help of the *GMP* library whereas solution of differential problem was determined with double precision. The norm $\|\xi_n\|$ in the inequalities for the accuracy control was calculated by the formula

$$\|\xi_n\| = \max_{1 \leq j \leq N} |\xi_n^i| / (|y_n^i| + r),$$

where i is a number of vector component, r is a positive parameter. If inequality $|y_n^i| < r$ is satisfied for the component with number i then the absolute error $r \cdot \epsilon$ is controlled. Otherwise we control the relative error ϵ , where ϵ is the required accuracy.

We chose the Van der Pol oscillator (29) as a test example. This problem has the stiffness ratio approximately equal to 10^6 :

$$\begin{aligned} y_1' &= y_2, & y_2' &= \left((1 - y_1^2)y_2 - y_1 \right) / 10^{-6}, \\ 0 \leq t \leq 1, & h_0 = 10^{-3}, & y_1(0) &= 2, & y_2(0) &= 0, & \epsilon &= 10^{-2}. \end{aligned} \quad (29)$$

The efficiency of two algorithms are compared. The first algorithm is the first order 5-stage Runge-Kutta method described in Section 5. The second algorithm is the traditional 5-stage Merson method of the fourth order of accuracy (27). Both algorithms were applied in two modes: with stability control and without it. We counted total numbers of steps, repeat computations of a solution (due to omission of the defined accuracy), and the number computations of the

right hand side the problem. The accuracy $\varepsilon = 10^{-2}$ was supported with the Merson method, whereas for the first-order method we needed to use $\varepsilon = 10^{-5}$ in order to provide 10^{-2} in fact. Nevertheless, under these conditions the constructed algorithm shows better efficiency (Fig. 4).

The comparison of two algorithms shows that stability control increases efficiency because extra repeat computations of solution originating from instability of numerical scheme are eliminated. In addition, the constructed first-order algorithm has less computational cost estimated by the number of computations of the right hand side. Simulations of other test examples show similar pattern.

Criterion	First-order method without stability control	First-order method with stability control	Merson method without stability control	Merson method with stability control
Number of integration steps	69 433	51 414	549 241	556 114
Number of repeated solution calculations	20 001	1 052	187 120	6 464
Number of right part computations	452 683	309 948	3 494 685	2 806 426

Fig. 4. Numerical results for the Van der Pol oscillator problem

Conclusion

Implementation of the algorithm to obtain coefficients of stability polynomial with the use of the *GMP* library allowed one to build stability polynomial up to degree $m = 40$. It provides a possibility to develop methods with extended stability regions with respective number of stages. The greater number of stages the larger stability interval is, and therefore the higher efficiency of numerical scheme is achieved in the case of stiff problems.

Comparing two five-stage methods (the proposed first-order method and the Merson method), one can see that at the same number of stages extending the stability interval decreases the overall computational cost.

It is important to say that the first-order methods with extended stability regions allow one to significantly increase the efficiency in the region where the step is restricted by stability. So methods described here can be used in adaptive algorithms where number of stages may vary from one integration step to another. It provides large stability interval where it is needed and decreases computational cost when numerical scheme is unconditionally stable.

The study was funded by Russian Foundation for Basic Research (project no. 18-31-00375).

References

- [1] E.Hairer, G.Wanner, Solving ordinary differential equations II. Stiff and differential-algebraic problems, Berlin, Springer, 1996.
- [2] E.A.Novikov, Explicit methods for stiff systems, Novosibirsk, Nauka, 1997 (in Russian).
- [3] E.A.Novikov, A.E.Novikov, *International Journal of Mathematical Models and Methods in Applied Sciences*, **9**(2015), no. 1, 62–70.

- [4] E.A.Novikov, Yu.V.Shornikov, Computer simulation of stiff hybrid systems, Novosibirsk: Publisher of NSTU, 2012 (in Russian).
- [5] A.E.Novikov, E.A.Novikov, L-stable (2,1)-method for stiff nonautonomius problem solving, *Computing technologies*, **13**(2008), 477–482 (in Russian).
- [6] E.A.Novikov, Yu.A.Shitov, Integration algorithm for stiff systems based on a second-order accuracy (m, k)-method with numerical calculation of the Jacobi matrix, Krasnoyarsk: Preprint of the Exhibition Center of the Siberian Branch of the USSR Academy of Sciences no. 20, 1988 (in Russian).
- [7] E.A.Novikov, M.V.Rybkov, *Bulletin of the Buryat State University*, (2014), no. 9–2, 80–85 (in Russian).
- [8] E.A.Novikov, M.V.Rybkov, The numerical algorithm of constructing of stability regions for explicit methods, *Control systems and information technologies*, **55**(2014), no. 1.1, 173–177 (in Russian).
- [9] Yozo Hida, Xiaoye S Li, David H Bailey, Quad-double arithmetic: algorithms, implementation, and application, Technical Report LBNL–46996, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, 2000.
- [10] L.V.Knaub, P.S.Litvinov, A.E.Novikov, M.V.Rybkov, *University Scientific Journal*, (2016), no. 22, 49–58.
- [11] R.H.Merson, *An operational methods for integration processes*, Proc. of Symp. on Data Processing, Weapons Research Establishment, Salisbury, Australia, 1957.

Методы первого порядка с расширенными областями устойчивости для расчета задач электрических цепей

Михаил В. Рыбков

Людмила В. Кнауб

Данил В. Хоров

Сибирский федеральный университет

Российская Федерация

Аннотация. Исследуется применение контроля устойчивости численных схем типа Рунге-Кутты для повышения эффективности при интегрировании жестких задач. Приведена реализация алгоритма определения коэффициентов полиномов устойчивости, при которых метод имеет заданную форму и размер области устойчивости, с помощью библиотеки GMP. Построены наборы методов первого порядка с расширенными областями устойчивости. Приведены результаты расчетов задач из теории электрических цепей, показывающие повышение эффективности построенных методов первого порядка точности в сравнении с методом более высокого порядка.

Ключевые слова: жесткая задача, явные методы, контроль точности, контроль устойчивости.