Explicit-Implicit Variable Structure Algorithm for Solving Stiff Systems

Eugeny A. Novikov, Anton E. Novikov

***Abstract*—** An algorithm of variable structure for solving stiff problems is constructed using L-stable and explicit methods. It is based on explicit and L-stable methods, both schemes of order two, and on an explicit method of the first order, which interval of stability is extended. On each step an efficient numerical scheme is chosen by criterion of stability. The numerical results of modeling the simplest oregonator and the modified oregonator exhibiting complicated limit cycle are given.

***Keywords*—** Stiff system, accuracy and stability control, variable structure algorithms.

I. Introduction

The Cauchy problem for stiff systems of ODEs arises in many applications [1, 2]. The main trends in construction of numerical methods are associated with expansion of their possibilities in solving large-scale problems.

In many cases calculations are required to be conducted within limits of so called engineering accuracy about 1% and lower. This is due to the fact, that measurement of constants in a right part of a system of differential equations is often quite rough. Sometimes, such accuracy of calculations is satisfactory in terms of a goal. It is well-known (see, e.g. [3]), that order of approximation of a numerical scheme should be associated with required accuracy of calculations. Therefore, below we shall consider only those numerical formulas, that have order of accuracy less or equal to two.

Modern methods for solving stiff problems usually use calculation and inversion of the Jacobi matrix of a system of differential equations. In case of a sufficiently large dimension, efficiency of numerical methods is almost completely determined by inversion (decomposition) of the matrix. To increase efficiency of calculations in a number of algorithms, the freezing of the Jacobi matrix is used, that means using same matrix on several integration steps [4]. This approach is the most successful in algorithms based on multistep methods and, in particular, in backward differentiation formulas [5]. This problem does not cause any particular difficulties in constructing integration algorithms based on other numerical schemes, if their stages are computed with the Jacobian matrix in some iterative process.

This is due to the fact, that in this case the Jacobian matrix does not affect accuracy order of a numerical scheme, but only determines rate of convergence of iterations. So, it needs to be recomputed, when there is a significant slowdown on convergence rate of the iterative process.

The situation is worse in an integration algorithm, based on the known noniterative methods, which include methods of the Rosenbrock type [6] and their various modifications [4]. It should be noted, that the noniterative method is much simpler in terms of computer implementation than algorithms, based on numerical formulas, which are evaluated with using iterations. However, in methods of form [6], the Jacobi matrix affects accuracy order of a numerical scheme and, therefore, difficulties with its freezing arise. If a problem of using same matrix on several steps of integration is left unsolved, then, obviously one is limited to solve only problems of low dimensions. In [7, 8], this problem is considered in relation to the Rosenbrock methods. It is proved, that maximum accuracy order of the Rosenbrock methods is equal to two, if in an integration algorithm the same Jacobi matrix is applied on several steps of integration. There is an algorithm with freezing the Jacobi matrix, based on L-stable numerical formulas of second accuracy order and results of calculations, confirming its high efficiency.

Another important requirement for modern integration algorithms is numerical approximation of the Jacobi matrix. This is due to the fact, that a right part of a system of differential equations often has large dimensions and quite complex form. A typical example is provided by problems of chemical kinetics, where complexity of a right part increases with number of elementary stages in a chemical reaction. Nowadays, simulation involves reactions, which contain dozens of reagents and hundreds of elementary stages. Therefore, in some cases, less effective numerical methods are more preferable, if their implementation does not require the analytical calculation of elements of the Jacobi matrix. This barrier can be removed if an integration algorithm includes possibility of numerical approximation of the Jacobi matrix. Note, that the problems of freezing and numerical approximation are in some sense close to each other and, therefore, can be solved simultaneously.

Some analog of freezing the Jacobi matrix is using in calculations integration algorithms, based on explicit and L-stable methods with automatic selection of a numerical scheme. In this case, efficiency of the algorithm can be

This work was supported by Russian Science Foundation (project 14-11-00147).

Eugeny A. Novikov is with the chief researcher at the Institute of Computational Modelling Siberian Branch of Russian Academy of Science, Krasnoyarsk, Russia (email: novikov@icm.krasn.ru).

Anton A. Novikov is the Senior lecturer of Siberian Federal University, Krasnoyarsk, Russia (e-mail: aenovikov@bk.ru).

improved by calculating transitive regions corresponding to a maximum eigenvalue of the Jacobi matrix by an explicit method. It is natural to apply an inequality for stability control [9] as a criterion for choosing an efficient numerical formula. Note, that using such hybrid algorithms does not fully eliminate the problem of freezing the Jacobi matrix, because the explicit method can be applied, generally speaking, only for a boundary layer solution, corresponding to a maximum eigenvalue of the Jacobi matrix.

Here, based on the explicit methods of the Runge-Kutta type of the first and second orders, as well as the L-stable (2,1)-method of second-order accuracy, an algorithm of variable structure is constructed, which allows both a numerical and an analytical Jacobi matrix. Numerical results confirm high efficiency of the integration algorithm.

To insert images in *Word*, position the cursor at the

II. *L*-STABLE (2,1)-METHOD

In [10] for numerical solution of the Cauchy problem for stiff systems of ordinary differential equations

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

where y and f are real N-dimensional vector functions, t is an independent variable, the class of (m,k)-methods is proposed. From the standpoint of computer implementation, (m,k)-methods are as simple as the Rosenbrock schemes. However, in contrast to the Rosenbrock methods, in this class it is much easier to solve a problem of freezing the Jacobi matrix and its numerical approximation. In addition, (m,k)-methods have more good properties of accuracy and stability with slight increase of computational cost. In traditional methods, number of stages m completely describes a numerical formula. In (m,k)-methods two constants are required to describe numerical schemes: number of stages m and number of calculations of a right part of the system (1) on an integration step k.

To solve the problem (1), we consider a (2,1)-scheme

$$\begin{aligned} &y_{n+1} = y_n + p_1 k_1 + p_2 k_2, \\ &D_n k_1 = hf\left(t_n + \beta h, y_n\right),\ D_n k_2 = k_1, \end{aligned} \tag{2}$$

where k_1 and k_2 are stages of the method; $D_n = E - ahA_n$, E is an identity matrix, h is an integration step, A_n is some matrix, which can be represented in a following form

$$A_n = f_n' + hB_n + O\left(h^2\right), \tag{3}$$

$f_n' = \partial f\left(t_n, y_n\right)/\partial y$ is the Jacobi matrix of the system (1), B_n is an independent of an integration step arbitrary matrix, a, β, p_1 and p_2 are numerical coefficients. Using the matrix A_n represented in form (3) allows us to apply (2) with freezing both an analytical and a numerical Jacobi matrix [11]. In case of using the Jacobi matrix f_{n-k}', calculated k steps back we have

$B_n = -kf_n'' f_n$, $f_n'' f_n = \partial^2 f(y_n)/\partial y^2$.

If the Jacobi matrix is computed numerically with a step $r_j = c_j h$, then elements $b_{n,ij}$ of the matrix B_n have a form

$b_{n,ij} = 0.5 c_j \partial^2 f_i\left(t_n, y_n\right)/\partial y_j^2$.

In calculations the step r_j is chosen according to a formula

$r_j = \max(10^{-14}, 10^{-7} \mid y_j \mid)$.

Let's obtain the coefficients of the L-stable numerical scheme (2) of second order and an inequality for accuracy control. An expansion of an exact solution $y(t_{n+1})$ in the Taylor series in a vicinity of a point t_n to terms with h^3 inclusive has a form

$$\begin{aligned} y\left(t_{n+1}\right) &= y\left(t_n\right) + hf + \frac{1}{2}h^2\left[f_t' + f_y' f\right] \\ &+ \frac{1}{6}h^3\left[f_{tt}'' + f_y' f_t' + 2f_{yt}'' f + f_y'^2 f + f_{yy}'' f^2\right] + O\left(h^4\right) \end{aligned} \tag{4}$$

where the elementary differentials are calculated on an exact solution $y(t_n)$. To find the coefficients β, a, p_1 и p_2 of the scheme (2) we write an expansion of the stages k_1 and k_2 in a Taylor series in a vicinity of a point y_n to terms with h^3 inclusive and substitute it in (2). We obtain

$$\begin{aligned} y_{n+1} &= y_n + \left(p_1 + p_2\right)hf_n + \beta\left(p_1 + p_2\right)h^2 f_{t,n}' \\ &+ a\left(p_1 + 2p_2\right)h^2 f_{y,n}' f_n + \frac{1}{2}\beta^2\left(p_1 + p_2\right)h^3 f_{tt,n}'' \\ &+ a\beta\left(p_1 + 2p_2\right)h^3 f_{y,n}' f_{t,n}' + a^2\left(p_1 + 3p_2\right)h^3 f_{y,n}'^2 f_n \\ &+ a\left(p_1 + 2p_2\right)h^3 B_n f_n + O\left(h^4\right), \end{aligned} \tag{5}$$

where the elementary differentials are calculated on an approximate solution y_n. Assuming, that $y_n = y(t_n)$ and comparing the expansions (4) and (5) to terms with h^2 inclusive, we obtain conditions of second-order accuracy of scheme (2), i.e.

$$p_1 + p_2 = 1,\ a(p_1 + 2p_2) = \frac{1}{2},\ \beta = \frac{1}{2}. \tag{6}$$

Let's investigate stability of a numerical formula (2). Applying it to a problem

$$y' = \lambda y,\ y(0) = y_0,\ \mathrm{Re}(\lambda) < 0, \tag{7}$$

we obtain $y_{n+1} = Q(x)y_n, x = h\lambda$, where a function of stability $Q(x)$ has the following form

$$Q(x) = \frac{1 + \left(p_1 + p_2 - 2a\right)x + a\left(a - p_1\right)x^2}{\left(1 - ax\right)^2}.$$

Then, the scheme (2) is L-stable, if $p_1 = a$. Substituting this relation in (6), we obtain a set of the coefficients

$$p_1 = a,\ p_2 = 1 - a,\ \beta = \frac{1}{2}, \tag{8}$$

where a is determined from a L-stability condition

$$a^2 - 2a + \frac{1}{2} = 0. \tag{9}$$

Comparing (4) and (5) to terms with h^3 inclusive, we find, that a local error δ_n of the numerical scheme (2) with the coefficients (8) have a form

$$\delta_n = \left(a-\frac{1}{3}\right)h^3 f_y'^2 f + \frac{1}{24}h^3 f_{tt}'' + \frac{1}{6}h^3 f_{yy}'' f^2 + \frac{1}{3}h^3 f_{yt}'' f - \frac{1}{2}h^3 f_y' f_t' - \frac{1}{2}h^3 B_n f + O\left(h^4\right). \quad (10)$$

The equation (9) has two roots $a_1 = 1-0.5\sqrt{2}$ and $a_2 = 1+0.5\sqrt{2}$. We choose $a = a_1$, as in this case the coefficient in the leading term $(a-1/3)h^3 f'^2 f$ of error (10) is less.

Let's consider simultaneously the numerical Rosenbrock formula with two calculations of the function f on an each step

$$y_{n+1} = y_n + p_1 k_1 + p_2 k_2,$$
$$D_n k_1 = hf\left(y_n\right),\ D_n k_2 = hf\left(y_n + ak_1\right). \quad (11)$$

According to [7], for $\gamma = a$, a set of coefficients (8) provides a second accuracy order of (11), and condition (9) provides its L-stability. It follows from [7] that the numerical formula (11) with the coefficients (8) is one of the most efficient among the methods of the Rosenbrock type, with two computations of a right part of a differential problem on an integration step. A local error δ_n^{roz} of the numerical formula (11) has a form

$$\delta_n^{roz} = h^3\left(a-\frac{1}{3}\right)f'^2 f + \left(\frac{1}{6}+\frac{1-\sqrt{2}}{2}a\right)h^3 f'' f^2 - ah^3 B_n f + O\left(h^4\right). \quad (12)$$

The scheme (2) with coefficients (8) as well as scheme (11) with coefficients (8) has second accuracy order and L-stability, and their local errors (10) and (12) differ slightly. At the same time, the scheme (2) requires one less calculation of function f than (11) on an each step, with other costs being equal, which makes it more preferable.

We construct accuracy control of the numerical scheme (2) by analogy with [12]. For this purpose, we denote

$$v\left(j_n\right) = D_n^{1-j_n}\left(k_2 - k_1\right), \quad (13)$$

where k_1 and k_2 are calculated by the formulas (2). Then, according to [12], in order to control the accuracy on an each step, one has to control an inequality

$$\left\|v(j_n)\right\| \le \varepsilon,\ 1 \le j_n \le 2, \quad (14)$$

where ε is required accuracy of calculations, $\|\cdot\|$ is some norm in R^N, and the integer variable j_n is selected as the lowest, for which inequality (14) holds.

Note one important feature of error estimation (13). The scheme (2) is L-stable, that is, for its stability function $Q(x)$, the relation $Q(x) \to 0$ for $x \to -\infty$ holds. Since for an exact solution $y(t_{n+1}) = \exp(x)y(t_n)$ of the problem (7) a similar property holds, it is natural to require convergence to zero of the error estimation for $x \to -\infty$. However, for $v(1)$ this property is not satisfied — this estimation has an A-stable manner.

To correct asymptotic behavior of the estimated error we introduced an estimation $v(j_n), 1 \le j_n \le 2$ instead of $v(1)$. In this case, behavior of error estimations for $j_n = 2$ will be coordinated with behavior of the exact solution of the test problem for $x \to -\infty$. We emphasize, that in sense of a general member, estimations $v(1)$ and $v(2)$ coincide. Using $v(j_n)$ actually does not lead to increase of computational costs. This is due to the fact, that $v(j_n)$ for $j_n = 2$ is checked only when it is violated for $j_n = 1$. This situation appears rarely, mainly when an integration step grows rapidly. However, this allows us to choose the step more accurately and thereby reduce the number of unnecessary recomputing solutions (returns).

An estimation of a maximum eigenvalue $w_{n,0} = h\lambda_{n,\max}$ of the Jacobi matrix of the system (1), necessary to switch to an explicit formula, is estimated through its norm $w_{n,0} = h\left\|\partial f\left(t_n, y_n\right)/\partial y\right\|$. Below, this estimation will be used for automatic selection of a numerical scheme.

III. EXPLICIT SECOND ORDER RUNGE-KUTTA METHOD

For solution the problem (1) we consider an explicit two-stage Runge-Kutta method [13]

$$y_{n+1} = y_n + p_1 k_1 + p_2 k_2,$$
$$k_1 = hf\left(y_n\right),\ k_2 = hf\left(y_n + \beta k_1\right). \quad (15)$$

Let's consider the autonomous problem (1) to simplify formulas. In case of a non-autonomous system $y' = f(t, y)$ the scheme (15) is written as

$$y_{n+1} = y_n + p_1 k_1 + p_2 k_2,$$
$$k_1 = hf\left(t_n, y_n\right),\ k_2 = hf\left(t_n + \beta h, y_n + \beta k_1\right).$$

We obtain relations for the coefficients of the method (15) of second accuracy order. For this purpose, we expand the stages k_1 and k_2 in Taylor series in powers of h up to terms with h^3 inclusive, and substitute them in the first formula (15). As a result, we obtain

$$y_{n+1} = y_n + \left(p_1 + p_2\right)hf_n + \beta p_2 h^2 f_n' f_n + \frac{1}{2}\beta^2 h^3 p_2 f_n'' f_n^2 + O\left(h^4\right),$$

where the elementary differentials are calculated on the approximate solution y_n. Comparing this expression with (4) to terms with h^2 inclusive, assuming, that $y_n = y(t_n)$, we write conditions $p_1 + p_2 = 1$ and $\beta p_2 = 0.5$ of second accuracy order of the scheme (15). In these relations, the local error δ_n of the scheme (15) can be written as follows

$$\delta_n = h^3\left[\frac{1}{6}f'^2 f + \frac{2-3\beta}{12}f'' f^2\right] + O\left(h^4\right).$$

We construct an inequality for accuracy control. For this purpose, we consider an auxiliary scheme $y_{n+1,1} = y_n + k_1$ of first accuracy order. Using the idea of nested methods, estimation of error $\varepsilon_{n,2}$ of the second-order method can be calculated by formula [12]

$\varepsilon_{n,2} = y_{n+1} - y_{n+1,1} = p_2 (k_2 - k_1)$.

To improve reliability of this estimation, we choose $\beta = 1$. Then, stage k_1 is computed at the point t_n, and k_2 is computed at the point t_{n+1}. Calculations show, that using information in extreme points of a step leads to more reliability. For $\beta = 1$, coefficients of the method of second order are uniquely determined $p_1 = p_2 = 0.5$ and a local error and an accuracy control inequality are, respectively, given below

$\delta_n = \frac{h^3}{12}\left[2f'^2 f - f''f\right] + O(h^4)$, $0.5\|k_2 - k_1\| \le \varepsilon$.

Now, we construct an inequality for stability control of (15) by method proposed in [9]. For this purpose, we consider an auxiliary stage $k_3 = hf(y_{n+1})$. Note, that k_3 coincides with the stage k_1, which is used on a next integration step and, therefore, its applying does not lead to additional computing of the right part of (1). We write the stages k_1, k_2 and k_3 to a problem $y' = Ay$, where A is a matrix with constant coefficients. A result is given below

$k_1 = Xy_n$, $k_2 = \left(X + X^2\right) y_n$,

$k_3 = \left(X + X^2 + 0.5X^3\right) y_n$,

where $X = hA$. It is easy to see, that

$k_2 - k_1 = X^2 y_n$, $2(k_3 - k_2) = X^3 y_n$.

Then, according to [9] an estimation of a maximum eigenvalue $w_{n,2} = h\lambda_{n,max}$ of the Jacobi matrix of the system (1) can be calculated by a following formula

$$w_{n,2} = 2\max_{1\le i\le N}\left\{\frac{\left|k_3^i - k_2^i\right|}{\left|k_2^i - k_1^i\right|}\right\}. \quad (16)$$

A stability domain of the scheme (15) is shown on a Fig. 1.

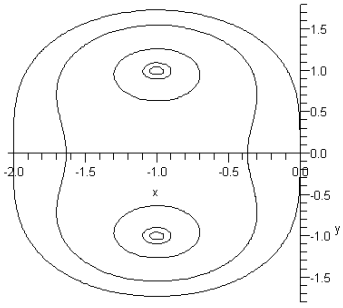

Fig. 1 Stability domain of the scheme (15)

A stability interval of (15) of second accuracy order is approximately equal to two. Therefore, an inequality $w_{n,2} \le 2$ can be applied for stability control. In case of using this inequality for step selection, roughness of estimation (16) should be considered, because a maximum eigenvalue may not be strongly separated from rest, in a power method few iterations are applied and additional distortions occur, because of nonlinearity of the problem (1). Therefore, stability control is used to limit size of an integration step. As a result, we will calculate the projected step h_{n+1} as follows. We define a new step h^{ac} by criterion of accuracy according to the formula $h^{ac} = qh_n$, where h_n is the last successful step of integration and q, taking into account a relation

$k_2 - k_1 = O(h_n^2)$,

is given by an equation

$q^2\|k_2 - k_1\| = \varepsilon$.

A step h^{st} by criterion of stability is given by a formula $h^{st} = dh_n$, where d, taking into account a relation $w_{n,2} = O(h)$, is determineв from an equation $dw_{n,2} = 2$. Then, the projected step h_{n+1} is calculated by a formula

$$h_{n+1} = \max\left[h_n, \min\left(h^{ac}, h^{st}\right)\right]. \quad (17)$$

Note, that the formula (17) is used to predict size of the integration step h_{n+1} after successful computation of solution with the previous step h_n and, therefore, does not actually lead to increase of computational cost. If the step by criterion of stability is less than the last successful one, it will not be reduced, because it may be caused by roughness of estimation of a maximum eigenvalue. However, the step will not be increased, because there is a possibility of instability of the numerical scheme. If the step should be reduced by criterion of stability, then the latest successful step h_n is applied again. As a result, the formula (17) is proposed to select a step. This formula allows to stabilize step behavior on a settling region of solution, where stability has a defining role. Indeed, existence of this region limits possibilities of applying explicit methods for solving stiff problems.

IV. FIRST ORDER RUNGE-KUTTA METHOD

For numerical solution of the problem (1), we consider a scheme

$$y_{n+1} = y_n + r_1 k_1 + r_2 k_2,$$
$$k_1 = hf(y_n),\ k_2 = hf(y_n + k_1). \quad (18)$$

Note, that when $r_1 = r_2 = 0.5$, the numerical formula (18) has second order of accuracy, and coincides with (15) with coefficients $p_1 = p_2 = 0.5$. We construct a less accurate scheme with a maximum interval of stability. For this purpose, we use (18) for solution the scalar test equation (7). We obtain $y_{n+1} = Q(x)y_n$, where the function of stability $Q(x)$ has a form

$Q(x) = 1 + (r_1 + r_2)x + r_2 x^2$, $x = h\lambda$.

Requirement of first order of accuracy leads to the relation

$r_1 + r_2 = 1$,

which, below, we will assume to be satisfied. Now, we choose r_2 so that the method (18) has a maximum stability interval. For this purpose, we consider the Chebyshev polynomial

$T_2(z) = (2z^2 - 1)$

on an interval [-1,1]. We carry out change of variables, setting

$z = 1 - \frac{2}{\gamma} x$.

We obtain

$T_2(x) = 1 - \frac{8}{\gamma} x + \frac{8}{\gamma^2} x^2$,

and the interval $[\gamma, 0]$ passes to [-1,1]. It is easy to show, that among all polynomials of a form $P_2(x) = 1 + x + c_2 x^2$

for $T_2(x)$ the inequality $|T_2(x)| \le 1$ is satisfied at a maximum interval $[\gamma, 0]$, $\gamma = -8$. We require, that the coefficients of $Q(x)$ and $T_2(x)$ coincide at $\gamma = -8$. This leads to relations

$r_1 + r_2 = 1$, $r_2 = \frac{1}{8}$.

As a result, we have coefficients

$r_1 = \frac{7}{8}$, $r_2 = \frac{1}{8}$

of the first accuracy order method with a maximum stability interval, with a local error

$\delta_n = \frac{3}{8} h^2 f f' + O\left(h^3\right)$.

We will use estimation of the local error to control accuracy of the numerical formulas of the first order. Taking into account, that

$k_2 - k_1 = h^2 f_n' f_n + O\left(h^3\right)$

and a form of the local error, an inequality for accuracy control can be written as

$\|k_2 - k_1\| \le \frac{8}{3} \varepsilon$,

where $\|\cdot\|$ is some norm in R^N, ε is required accuracy of calculations.

We construct an inequality for stability control for a first order method. For this purpose, we consider an auxiliary stage $k_3 = hf(y_{n+1})$.

We write k_1, k_2 and k_3, applied to the problem $y' = Ay$, where A is the matrix with constant coefficients. As a result, we obtain

$k_1 = Xy_n$, $k_2 = \left(X + X^2\right) y_n$,

$k_3 = \left(X + X^2 + \frac{1}{8} X^3\right) y_n$,

where $X = hA$. It is easy to see, that

$k_2 - k_1 = X^2 y_n$, $8\left(k_3 - k_2\right) = X^3 y_n$.

Then, according to [9], estimation of a maximum eigenvalue $w_{n,1} = h\lambda_{n,max}$ of the Jacobi matrix of the system (1) can be calculated by a formula

$w_{n,1} = 8 \max\limits_{1 \le i \le N} \left\{ \frac{\left|k_3^i - k_2^i\right|}{\left|k_2^i - k_1^i\right|} \right\}$.

A stability domain of the scheme (18) is shown on a Fig. 2.

An interval of stability of the numerical scheme (18) is equal to eight. Therefore, an inequality $w_{n,1} \le 8$ can be applied to control stability.

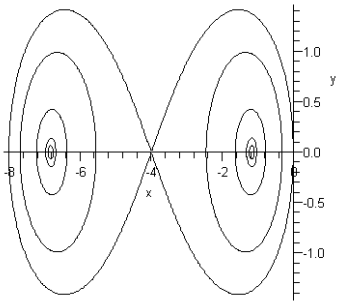

Fig. 2 Stability domain of the scheme (18)

V. ALGORITHM WITH AUTOMATIC SELECTION OF A NUMERICAL SCHEME

An algorithm of alternating order and step can be easily formulated on a base of constructed explicit methods of first and second orders of accuracy. Calculations are always begun with the second order method as it is more accurate. Transition to the first order scheme is carried out in case of violation an inequality $w_{n,2} \le 2$. Reverse transition to the second order method is carried out if an inequality $w_{n,1} \le 2$ holds. On calculations by the first order method in addition to accuracy control there is stability control, and choice of a projected step is carried out in the same manner as in the second order method applying a formula of type (17).

Using the scheme (2) does not present difficulties. Violation of an inequality $w_{n,1} \le 8$ causes a transition to the scheme (2). Transfer to explicit methods is carried out if an inequality $w_{n,0} \le 8$ holds.

The numerical formula (2), without the loss of the accuracy order, can be applied with the frozen matrix D_n. Note, that during the freezing the Jacobi matrix, size of an integration step remains constant. An attempt to freeze the matrix D_n is carried out after each successful step. The matrix thaws in the following cases: (1) violation of accuracy of calculations; (2) if number of steps with a frozen matrix reaches a defined maximum number i_h; (3) if a projected step is greater than the last one by q_h times. By the numbers i_h and q_h we can affect the redistribution of computational cost. When $i_h = 0$ and $q_h = 0$, freezing does not occur; with increasing i_h and q_h, the number of calculations of a right part increases, while the number of inversions of the Jacobi matrix decreases.

The norm in a left part of inequality for accuracy control is calculated by a formula

$$\|k_2 - k_1\| = \max_{1 \le i \le N} \left\{ \frac{|k_2^i - k_1^i|}{|y_n^i| + r} \right\},$$

where i is a number of component, r is a positive parameter. If, for the i-th component of solution, an inequality $|y_n^i| < r$ holds, then an absolute error $r\varepsilon$ is controlled, otherwise, the relative error ε is controlled. Below, the algorithm of alternating order and step with automatic selection of an explicit or a L-stable numerical scheme is called RKMK2.

VI. Differential Equations of Chemical Kinetics

The kinetic scheme of any chemical reaction includes the following elementary reactions (stages)

$$\begin{gathered} \alpha_{1,1}x_1 + \ldots + \alpha_{\mathrm{NR},1}x_{\mathrm{NR}} \to \beta_{1,1}x_1 + \ldots + \beta_{\mathrm{NR},1}x_{\mathrm{NR}} \\ \ldots\ \ldots\ \ldots \\ \alpha_{1,1}x_1 + \ldots + \alpha_{\mathrm{NR},1}x_{\mathrm{NR}} \to \beta_{1,1}x_1 + \ldots + \beta_{\mathrm{NR},1}x_{\mathrm{NR}} \end{gathered} \tag{19}$$

where x_i, $1 \le i \le \mathrm{NR}$ are chemical reagents, NR and NS are numbers of the reagents and the stages in the reaction, respectively; α_{ij} and β_{ij}, $1 \le i \le \mathrm{NR}$, $1 \le j \le \mathrm{NS}$ are the stoichiometric coefficients. For each elementary reaction a corresponding stage velocity constant k_j, $1 \le j \le \mathrm{NS}$ is given. For process (19) under the lumped model of an isothermal reactor of constant capacity the corresponding ordinary differential equations system has the following form

$$C' = A^T V,\ C(0) = C_0.$$

Here, A^T is a stoichiometric matrix, C and V are vectors of reagents concentrations and stage velocities, respectively. When a reaction proceeds in non-isothermal conditions, this system also involves the heat balance equation

$$T' = \frac{Q^T V - \alpha(T - T_{01})}{C_V^T C},$$

where T is mixture temperature in a reactor, T_{01} is temperature of walls of a reactor, Q^T is the vector of calorific capacities of the stages, C_V^T is a vector of heat capacities of the reagents, $\alpha = \tilde{\alpha} s / r$, where $\tilde{\alpha}$ is a heat-conduction coefficient, s and r are square of an area and volume of the reactor, respectively. The superscript T of the vectors Q^T and C_V^T denotes the transposition. The heat capacities of the reagents and the heat-conduction coefficient may be functions of the reagents concentrations c_i, $1 \le i \le \mathrm{NR}$, α may depend on temperature.

If a reaction proceeds in an isothermal reactor of constant capacity with substance exchange (an open system, a perfect-mixing reactor), a system of ordinary differential equations has a form

$$C' = A^T V + \frac{1}{\Theta}(C_p - C),\ C(0) = C_0.$$

where C_p is a vector of inlet reagent concentrations, $\Theta = r/u$ is a stay period of mixture in a reactor, u is mixture space velocity. If a reaction proceeds in non-isothermal conditions, then this system also involves a heat balance equation

$$T' = \frac{Q^T V - \alpha(T - T_{01})}{C_V^T C} - \frac{1}{\Theta}(T - T_{02}),$$

where T_{02} is inlet mixture temperature in a reactor. The temperature of reaction mixture can be described by a function of time t and concentrations of reagents c_i, $1 \le i \le \mathrm{NR}$, i.e. $T = T(t, C)$.

VII. Algorithm for Generating Chemical Kinetic Equations

If an elementary reaction is balancing, its velocity W_s is equal to a difference of its forward W_s^+ and backward W_s^- reactions, i.e. $W_s = W_s^+ - W_s^-$, $1 \le s \le \mathrm{NS}$. If there is some third entity in a reaction, the velocity V_s is calculated by formulas

$$V_s = P_s W_s,\ P_s = \sum_{i=1}^{\mathrm{NR+NI}} \varepsilon_{si} c_i,\ 1 \le s \le \mathrm{NS},$$

where ε_{si}, $1 \le s \le \mathrm{NS}$, $\mathrm{NR}+1 \le i \le \mathrm{NR+NI}$, are efficiencies of the third entities, NI is number of inert substances, ε_{si} and c_i are efficiencies and concentrations of the inert substances, respectively. Values of components of the vector W_s are determined from a scheme of a chemical reaction (19) by relations

$$W_s^+ = k_s \prod_{i=1}^{\mathrm{NR+NI}} c^{\alpha_{ij}},\ W_s^- = k_{-s} \prod_{i=1}^{\mathrm{NR+NI}} c^{\beta_{ij}},$$

where k_s and k_{-s}, $1 \le s \le \mathrm{NS}$, are forward and backward velocity stage constants, respectively. Velocity constants of the stages are calculated using the formulas

$$k_j = A_j T^{n_j} \exp\left(-\frac{E_j}{RT}\right),$$

where T is temperature of mixture in a reactor; A_j, n_j and E_j / R are given constants. Note, that, in general, the velocity constants values are not constant in case of a non-isothermal reactor – they depend on temperatures. However, historically, an isothermal reactor was considered earlier than non-isothermal one and thus k_j, $1 \le j \le \mathrm{NS}$, at the present time, are still called constants. The stoichiometric matrix A^T with elements a_{ij} is formed from the kinetic scheme (19) conforming to the following rule. A number of a stage is aligned with a column number and a reagent number is aligned with a row number of a matrix A^T. If x_i is an initial reagent, then $a_{ij} = -\alpha_{ij}$, if x_i is a product, then $a_{ij} = \beta_{ij}$. If x_i is both an initial reagent and a product, then $a_{ij} = -\alpha_{ij} + \beta_{ij}$. Usually, a few amount of reagents are reacting in a stage, i.e. a stoichiometric matrix is sparse.

VIII. NUMERICAL RESULTS

Calculations were carried out on PC Intel(R) Core(TM) i7-3770S CPU@3.10GHz with double precision. In the calculations, the parameter r was chosen so that practical accuracy of all components of solution was not worse than required accuracy. Calculations were performed with defined accuracy $\varepsilon = 10^{-2}$. This is due to the fact, that the algorithm is based on low accuracy order schemes, and, therefore, it is impractical to carry out calculations with higher accuracy with this method. A comparison of its efficiency was carried out with the well-known Gear method in the implementation of A. Hindmarsh named DLSODE from the ODEPACK collection [5].

Below i_f and i_j denote, respectively, total numbers of calculations of a right part and number of inversions (decompositions) of the Jacobi matrix of the problem (1), which allow us to evaluate objectively the efficiency of the integration algorithm.

As the first test, the simplest model of the Belousov-Zhabotinsky reaction [14] was chosen

$$y_1' = 77.27\left(y_2 - y_1 y_2 + y_1 - 8.375\cdot 10^{-6} y_1^2\right),$$

$$y_2' = \frac{1}{77.27}\left(-y_2 - y_1 y_2 + y_3\right),$$

$$y_3' = 0.161\left(y_1 - y_3\right), \qquad (20)$$

$$y_1(0) = y_3(0) = 4,\ y_2(0) = 1.1,$$

$$t \in [0, 300],\ h_0 = 2\cdot 10^{-3}.$$

Calculations were carried out with the numerical Jacobi matrix.

A solution of this problem by the algorithm RKMK2 was calculated with costs $i_f = 1\,214$ and $i_j = 65$. Calculations only by the L-stable scheme (2) give $i_f = 926$ and $i_j = 88$. Practical accuracy of the calculations at the end of the interval of integration is not worse than required one. Solution of (20) was calculated by explicit methods of alternating order and step with cost $i_f = 2\,112\,678$. This problem is too stiff for shown here to demonstrate principal possibility of application explicit methods for solving stiff enough examples, those in solving some high-dimensional problems may be more efficient than L-stable methods. For calculations by the DLSODE program, the required accuracy 10^{-2} is achieved, when the defined accuracy equal is 10^{-4} with costs $i_f = 1\,129$ and $i_j = 107$. On calculations with higher accuracy, DLSODE is more efficient than the constructed algorithm. This is a sequence of low accuracy order of the constructed numerical formulas. On defined accuracy equal to 10^{-2}, the algorithm RKMK2 is more efficient than the well-known method DLSODE in 1.5 times in number of inversions of the Jacobi matrix, while number of computations of the right part of (20) for RKMK2 and DLSODE vary slightly. In case of the large-scale problem (1), the constructed integration algorithm may be more efficient than DLSODE in calculating time. The time dependence of y_1 is showed on a Fig. 3.

A second example describes the modified oregonator exhibiting complicated limit cycle. This reaction includes following six elementary stages [15]

$\mathrm{A+Y \to X+P}$, $k_1 = 0.084$, $k_{-1} = 10^4$,

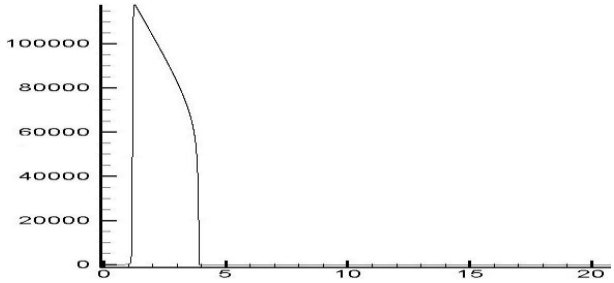

Fig. 3 Time dependence of y_1 (fragment)

$\mathrm{X+Y \to 2P}$, $k_2 = 4\cdot 10^8$, $k_{-2} = 5\cdot 10^{-5}$,

$\mathrm{A+X \to 2W}$, $k_3 = 2\cdot 10^3$, $k_{-3} = 2\cdot 10^7$,

$\mathrm{C+W \to X+Z}$, $k_4 = 1.3\cdot 10^5$, $k_{-4} = 2.4\cdot 10^7$,

$\mathrm{2X \to A+P}$, $k_5 = 4\cdot 10^7$, $k_{-5} = 4\cdot 10^{-11}$,

$\mathrm{Z \to C+0.462Y}$, $k_6 = 0.65$,

where k_i, $1 \le i \le 6$, velocity constants of forward (with positive indices) and backward (with negative indices) elementary stages. There are 7 entities in this reaction, denoted by

$\mathrm{A=BrO_3^-}$, C=M(n), P=HOBr,

$\mathrm{W=BrO_2}$, $\mathrm{X=HBrO_2}$, $\mathrm{Y=Br^-}$, Z=M(n+1).

In these notations M(n) is an ion of metal accelerant, M(n+1) is an oxygenated form of the ion. Let's denote concentrations of reagents by

$c_1 = [\mathrm{BrO_3^-}]$, $c_2 = [\mathrm{Br^-}]$, $c_3 = [\mathrm{M(n)}]$,

$c_4 = [\mathrm{HBrO_2}]$, $c_5 = [\mathrm{HOBr}]$, $c_6 = [\mathrm{BrO_2}]$, $c_7 = [\mathrm{M(n+1)}]$.

This reaction proceeds in an isothermal reactor with constant capacity with substance exchange. A corresponding system of equations is given below

$$c_1' = -v_1 - v_3 + v_5 + \frac{1}{\Theta}\left(c_{p1} - c_1\right),$$

$$c_2' = -v_1 - v_2 + 0.462 v_6 + \frac{1}{\Theta}\left(c_{p2} - c_2\right),$$

$$c_3' = -v_4 + v_6 + \frac{1}{\Theta}\left(c_{p3} - c_3\right),$$

$$c_4' = v_1 - v_2 - v_3 + v_4 - 2v_5 + \frac{1}{\Theta}\left(c_{p4} - c_4\right), \qquad (21)$$

$$c_5' = v_1 + 2v_2 + v_5 + \frac{1}{\Theta}\left(c_{p5} - c_5\right),$$

$$c_6' = 2v_3 - v_4 + \frac{1}{\Theta}\left(c_{p6} - c_6\right),$$

$$c_7' = v_4 - v_6 + \frac{1}{\Theta}\left(c_{p7} - c_7\right),$$

where $\Theta = 125.5$ and velocities v_1, v_2, …, v_6 of stages are defined by formulas

$v_1 = k_1 c_1 c_2 - k_{-1} c_4 c_5$, $v_2 = k_2 c_2 c_4 - k_{-2} c_5^2$,

$v_3 = k_3 c_1 c_4 - k_{-3} c_6^2$, $v_4 = k_4 c_3 c_6 - k_{-4} c_4 c_7$,

$v_5 = k_5 c_4^2 - k_{-5} c_1 c_5$, $v_6 = k_6 c_7$.

Integration of the system (21) was made on an interval [0,1000] with an initial step equal to 10^{-5}. Inlet concentrations of reagent are given below

$c_{p1} = 0.14$, $c_{p2} = 0.151 \cdot 10^{-5}$,

$c_{p3} = 0.125 \cdot 10^{-3}$, $c_{p4} = c_{p5} = c_{p6} = c_{p7} = 0$.

Initial values of reagent concentrations are equal to

$c_1 = 0.1387$, $c_2 = 0.1534 \cdot 10^{-6}$,

$c_3 = 0.1176 \cdot 10^{-3}$, $c_4 = 0.3165 \cdot 10^{-7}$,

$c_5 = 0.1956 \cdot 10^{-3}$, $c_6 = 0.5814 \cdot 10^{-6}$,

$c_7 = 0.631 \cdot 10^{-5}$.

Time dependence of [BrO_2] is showed on a Fig. 4.

Calculations were performed with the numerical Jacobi matrix. Solution of the problem was calculated by the algorithm

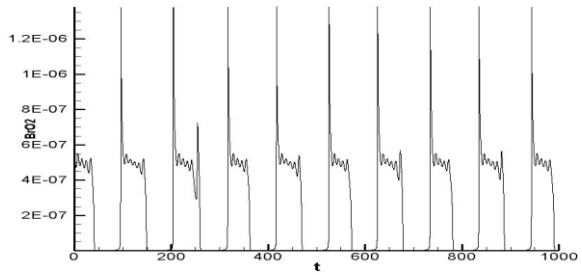

Fig. 4. Time-dependence of [BrO2] concentration

RKMK2 with i_f = 5 623 and i_j = 533. Calculations with the L-stable scheme (2) only give i_f =5 371 and i_j =591. Practical accuracy in the end of the integration interval is as good as defined one. In calculations by DLSODE required accuracy 10^{-2} is achieved for defined accuracy equal to 10^{-4} with computational costs i_f =7 806 and i_j =542. On higher accuracy of calculations DLSODE is more efficient than the constructed algorithm. It is a result of low order of accuracy of constructed numerical formulas. In case of high dimension of the problem (1) the constructed algorithm may be more efficient in time than DLSODE.

IX. Conclusion

The constructed algorithm RKMK2 is designed for low precision calculations — about 1% and lower. In this case, its maximum efficiency is reached.

In RKMK2, with its parameters, one can specify different modes of calculations:

(1) explicit methods of first or second order of accuracy with or without stability control;

(2) explicit methods of alternating order and step;

(3) L-stable method with or without freezing, both an analytical and a numerical Jacobi matrix.

(4) with automatic selection of a numerical scheme.

This allows us to apply this algorithm to solving both stiff and non-stiff problems. In calculations with automatic selection of a numerical scheme, the integration algorithm makes a decision whether a problem is stiff or not by itself.

Using the inequality for stability control does not actually lead to increase of computational cost, because estimation of a maximum eigenvalue of the Jacobi matrix of (1) is carried out through a previously computed stages and does not lead to increase of number of computed values of function f. This estimation is rough. However, using stability control for limiting the step growth allows us to avoid the negative effects of roughness of estimation. Moreover, in some cases this leads to an exceptionally high growth of efficiency of the algorithm. In a settling region , the old errors tend to zero due to stability control and new ones are low, due to small values of derivatives of a solution. In some cases, following eigenvalue is estimated instead of a maximum one. An integration step becomes greater than a maximal admissible one, and with such a step the integration is carried out as long as the inequality for accuracy control is not violated. Typically, the number of such steps is small. However, the step may be an order of magnitude greater than the maximum step by a criterion of stability. After violation of the inequality for accuracy control, the step is reduced to a maximum admissible. This effect may occur many times, depending on length of a region of settling. As a result, an average integration step may be greater than the maximum admissible one.

Application of the explicit first order method with an extended stability domain on a settling region allows us to increase step size by 4 times in comparison with the explicit second order method without increasing of computational cost. On transition regions, where accuracy of calculations has a defining role, the second accuracy order method is more efficient, though it has rather small stability domain.

Combining methods of low and high orders applying an inequality for stability control, improves efficiency of calculations.

Acknowledgment

This work was supported by Russian Science Foundation (project 14-11-00147).

References

[1] A. U. Moyi, “Wah June Leong. A New Inexact Line Search Method for Convex Optimization Problems analysis,” *Proceedings of the 2013 International Conference on Applied Mathematics and Computational Methods* (AMCM-2013), pp. 59-61, September 2013.

[2] F. A. Rihan, “Delay Differential Equations in Biosciences: Parameter estimation and sensitivity analysis,” *Proceedings of the 2013 International Conference on Applied Mathematics and Computational Methods* (AMCM-2013), pp. 50-58, September 2013.

[3] S. S. Artem’ev, G. V. Demidov, and E. A. Novikov, and L. A. Yumatova, “The Way to Optimize the Numerical Solution of Cauchy Problem for Ordinary Differential Equations,” *Numerical methods of continuum mechanics*, no. 2, pp. 5–14, 1984.

[4] E. Hairer, and G. Wanner, *Solving Ordinary Differential Equations*. *Stiff and Differential-Algebraic Problems*, Cambridge: Cambridge Univ. Press, 1987, ch. IV.
[5] G. D. Byrne, and A. C. Hindmarsh, "ODE Solvers: a Review of Current and Coming Attractions," *J. Comput. Phys*., no. 70, pp. 1–62, 1987.
[6] H. H. Rosenbrock, "Some General Implicit Processes for the Numerical Solution of Differential Equations," *Computer*, no. 5, pp. 329–330, 1963.
[7] E. A. Novikov, V. A. Novikov, and L. A. Yumatova, "Improving the Efficiency of an Integration Algorithm Based on the Rosenbrock Type Formula with the Second Accuracy Order by Freezing the Jacobi Matrix," *preprint no. 592*, Institute of Computational Mathematics and Mathematical Geophysics of SB RAS, 1985.
[8] E. A. Novikov, and A. L. Dvinskii, "Jacobi Matrix Freezing for Rosenbrock Type Methods," *Computational technologies*, no. 10, pp. 108–114, 2005.
[9] E. A. Novikov, *Explicit Methods for the Stiff Systems*, Novosibirsk: Nauka, 1997, ch. III.
[10] E. A. Novikov, Yu. A. Shitov, and Yu. I. Shokin, "Single-Step Methods for Solving the Stiff Systems," *Dokl. Akad. Nauk SSSR*, vol. 301, no. 6, pp. 1310–1314, 1988.
[11] E. A. Novikov, and Yu. A. Shitov, "Algorithm for Solving Stiff Systems Based on (m,k)-Method with Second Accuracy Order with Numerical Evaluation of the Jacobi Matrix", *preprint no. 20*, Institute of Computational Mathematics and Mathematical Geophysics of SB RAS, 1988.
[12] G. V. Demidov, and E. A. Novikov, "Error Estimation for One_step Methods for Integrating the Ordinary Differential Equations," *Numerical methods of continuum mechanics*, vol. 16, no. 1, pp. 27–42, 1985.
[13] L. V. Knaub, Yu. M. Laevskii, and E. A. Novikov, "Integration Algorithm of Variable Order and Step that is Based on the Explicit Two_Stage Runge-Kutta Method," *Siberian Journal of Numerical Mathematics*, vol. 10, no. 2, pp. 177–185, 2007.
[14] W. H. Enright, and T. E. Hull, "Comparing Numerical Methods for the Solutions of Systems of ODE's, " *BIT*, vol. 15, pp. 10–48, 1975.
[15] K. Showalter., and R. M. Noyes, "Bar-Eli K. A Modified Oregonator Model Exhibiting Complicated Limit Cycle Behavior in a Flow System," *J. Chem. Phys.*, vol. 69, pp. 2514–2524, 1978.

Eugeny A. Novikov. Dr. Sc. (1991) in computational mathematics. A chief researcher of the Institute of computational modeling of SB RAS, Krasnoyarsk. Research interests: computational mathematics, methods for solving stiff problems, chemical kinetics, simulation of problems in mechanics and electronics. He has more than 200 publications, mostly on methods for solving stiff problems. **Address:** ICM of SB RAS, Akademgorodok, Krasnoyarsk 660036 Russia. Tel.: 8(391)249-47-24, e-mail: novikov@icm.krasn.ru.

Anton E. Novikov. Cand. Sc. (2014) in computational mathematics. A teacher in the Siberian Federal University, Krasnoyarsk. Research interests: computational mathematics, methods for solving stiff problems, chemical kinetics, simulation of problems in mechanics and electronics. He has more than 40 publications, mostly on methods for solving stiff problems. **Address:** SFU, Svobodny pr., Krasnoyarsk 660041 Russia. Tel.: 8(963)955-80-90, e-mail: aenovikov@bk.ru.