UDC 519.622

E.A. Novikov

Institute of Computational Modeling SB RAS
50 Akademgorodok, ICM SD RAS, 660036, Krasnoyarsk, 660036, Russia

A SECOND-ORDER METHOD FOR ADDITIVE STIFF PROBLEMS

Е.А. Новиков

МЕТОД ВТОРОГО ПОРЯДКА ДЛЯ РЕШЕНИЯ АДДИТИВНЫХ ЖЕСТКИХ ЗАДАЧ

A second-order accuracy method for additive stiff systems of ordinary differential equations is developed. Inequalities for accuracy control are obtained. Numerical results are presented.

STIFF ADDITIVE PROBLEM, (M,K)-METHOD, ERROR ESTIMATION.

Построен метод второго порядка точности для решения жестких аддитивных систем обыкновенных дифференциальных уравнений. Получено неравенство для контроля точности вычислений. Приведены результаты расчетов.

ЖЕСТКАЯ АДДИТИВНАЯ ЗАДАЧА, (M,K)-МЕТОД, ОЦЕНКА ОШИБКИ

I. Introduction

For the numerical solution of the Cauchy problem

$$y' = f(t, y), \ \ y(t_0) = y_0, \ \ t_0 \le t \le t_k, \qquad (1)$$

for a stiff system of ordinary differential equations, *L*-stable methods are usually applied. Here *y* and *f* are real vector functions of dimension *N*, *t* is an independent variable.

For a large-scale problem (1), the total computational cost of a method with the unbounded stability domain in fact is completely defined by the time of the calculation and decomposition of Jacobi matrix of the system (1). In a number of algorithms, Jacobi matrix is frozen, i. e. one and the same matrix is used at several integration steps. This enables one to decrease a computational cost considerably. This approach is widely used when implementing semi-implicit and implicit methods of Runge – Kutta type and multistep methods of Adams and Gear type (see, for example, Ref. [9]). However, for iteration-free methods [2, 5, 8] the problem of the freezing or any other approximation of Jacobi matrix is rather more complicated. On the other hand, the problem (1) can be written in the form [1, 6]:

$$\begin{aligned} y' &= [f(t, y) - By] + By, \\ y(t_0) &= y_0, \ \ t_0 \le t \le t_k, \end{aligned} \qquad (2)$$

where *B* is an approximation of Jacobi matrix.

Assuming that the term *By* is completely responsible for stiffness, the expression in the brackets can be considered as a non-stiff part. Taking into account this fact when constructing an iteration-free method enables one, in particular, to freeze Jacobi matrix, which can be calculated analytically as well as numerically, in integration algorithms. For some problems, the symmetric part of Jacobi matrix or its diagonal approximation can be taken as the matrix *B*. Here we propose a second-order accuracy method which admits different types of approximation of Jacobi matrix. An error estimate and an inequality for the calculation accuracy control are obtained. Numerical results are presented.

II. A Numerical Scheme for Autonomous Problems

We consider the Cauchy problem for an autonomous system of the form

$$y' = \varphi(y) + g(y),\ \ y(t_0) = y_0,\ \ t_0 \le t \le t_k,\ (3)$$

where y, φ, and g are real vector functions of dimension N, and t is an independent variable.

We assume that the function $g(y)$ is completely responsible for stiffness and $\varphi(y)$ is a non-stiff part. For (3), we consider a method of the form

$$\begin{gathered} y_{n+1} = y_n + \textstyle\sum_{i=1}^{4} p_i k_i;\ \ D_n = E - ahg'_n; \\ k_1 = h\varphi(y_n);\ \ Dk_2 = h[\varphi(y_n) + g(y_n)]; \\ Dk_3 = k_2; \\ k_4 = h\varphi(y_n + \beta_{41}k_1 + \beta_{42}k_2 + \beta_{43}k_3), \end{gathered} \quad (4)$$

where E is the identity matrix; $g'_n = \partial g(y_n)/\partial g$; k_i, $1 \le i \le 4$ are the stages of the method; a, p_i, β_{4j}, k_i, $1 \le i \le 4$, $2 \le j \le 3$ are numerical coefficients which define accuracy and stability properties of (4).

To study the scheme (4), we substitute Taylor expansion of the stages k_i, $1 \le i \le 4$ in the first formula of (4). This gives

$$\begin{gathered} y_{n+1} = y_n + (p_1 + p_2 + p_3 + p_4)h\varphi_n + \\ + (p_2 + p_3)hg_n + (\beta_{41} + \beta_{42} + \beta_{43})p_4h^2\varphi'_n\,\varphi_n + \\ + (\beta_{42} + \beta_{43})p_4h^2\varphi'_n\,g_n + a(p_2 + 2p_3)h^2g'_n\varphi_n + \\ + a(p_2 + 2p_3)h^2g'_ng_n + O(h^3), \end{gathered}$$

where the elementary differentials φ_n, g_n, $\varphi'_n\varphi_n$, φ'_ng_n, $g'_n\varphi_n$, and g'_ng_n are calculated at an approximate solution y_n.

Taylor expansion of the exact solution $y(t_{n+1})$ is of the form

$$\begin{gathered} y(t_{n+1}) = y(t_n) + h(\varphi + g) + \\ + 0.5h^2(\varphi'\varphi + \varphi'g + g'\varphi + g'g) + O(h^3), \end{gathered}$$

where the elementary differentials φ, g, $\varphi'\varphi$, $\varphi'g$, $g'\varphi$, and $g'g$ are calculated at the exact solution $y(t_n)$. Comparing the above expansions for the condition $y_n = y(t_n)$, we arrive at the second-order conditions for the scheme (4), i. e.

$$\begin{gathered} p_1 + p_2 + p_3 + p_4 = 1,\ \ p_2 + p_3 = 1, \\ a(p_2 + 2p_3) = 0.5, \\ (\beta_{41} + \beta_{42} + \beta_{43})p_4 = 0.5,\ \ (\beta_{42} + \beta_{43})p_4 = 0.5. \end{gathered}$$

This results in

$$\beta_{41} = 0,\ \ p_2 = \frac{4a-1}{2a},\ \ p_3 = \frac{1-2a}{2a},$$

$$p_1 + p_4 = 0,\ \ (\beta_{42} + \beta_{43})p_4 = 0.5. \quad (5)$$

Now we study the stability of scheme (4). In this case we may not use the test equation $y' = \lambda y$ with a complex value λ, $\mathrm{Re}(\lambda) < 0$ since there is no sense in splitting the right-hand side of the system of differential equations into stiff and non-stiff parts. Hence, in (3) we put $\varphi(y) = \lambda_1 y$ and $g(y) = \lambda_2 y$ where λ_1 and λ_2 are arbitrary complex numbers. Here λ_1 and λ_2 mean some eigenvalues of the Jacobi matrices of the functions $\varphi(y)$ and $g(y)$, respectively. Applying (4) to the scalar test problem

$$y' = \lambda_1 y + \lambda_2 y,\ \ y(0) = y_0,\ \ t \ge 0, \quad (6)$$

with the notations $x = \lambda_1 y$ and $z = \lambda_2 y$, we have $y_{n+1} = Q(x,z)y_n$ where

$$\begin{gathered} Q(x,z) = \{1 + (1-2a)z + x + \\ + [-2ap_1 - ap_2 + (\beta_{42} + \beta_{43} - 2a)p_4]xz + 0.5x^2 - \\ - a\beta_{42}p_4x^2z + [a^2p_1 + a^2p_4 - a\beta_{42}p_4]xz^2 + \\ + (a^2 - ap_2)z^2\} / (1 - az)^2. \end{gathered}$$

The necessary L-stability condition for the numerical formula (4) with respect to the function $g(y) = \lambda_2 y$ is the relation $Q(x,z) \to 0$ as $z \to -\infty$. From the form of $Q(x,z)$, it follows that this relation is valid provided that $p_2 = a$ and $\beta_{42} = 0$. As a result, taking into account (5), we obtain the set of the coefficients

$$\begin{gathered} \beta_{41} = \beta_{42} = 0,\ \ p_2 = a,\ \ p_3 = 1 - a, \\ p_4 = -p_1 = 0.5\beta_{43}^{-1} \end{gathered}$$

for the second-order accuracy scheme (4) where β_{43} is a free parameter and a is a root of the equation $a^2 - 2a + 0.5 = 0$. Then, the stability function $Q(x,z)$ of the scheme (4) has the form

$$\begin{gathered} Q(x,z) = [1 + x + 0.5x^2 + (1-2a)z + \\ + (1-2a)xz] / (1 - az)^2. \end{gathered}$$

For $\varphi(y) = 0$, the scheme (4) coincides with the L-stable (2,1)-method

$$y_{n+1} = y_n + ak_2 + (1 - a)k_3\ [7]$$

with the stability function $Q(0, z)$ of the form

$$Q(0, z) = [1 + (1 - 2a)z]/(1 - az)^2$$

and the local error $\delta_{n,L}$ of the form

$$\delta_{n,L} = (a - 1/3)h^3g'^2g + h^3g''g^2/6 + O(h^4).$$

The equation

$$a^2 - 2a + 0.5 = 0$$

has two roots $a_1 = 1 - 0.5\sqrt{2}$ and $a_2 = 1 + 0.5\sqrt{2}$. We take $a = a_1$ since in this case the coefficient of the principal term of the local error of the (2,1)-scheme is smaller. For $g(y) = 0$, (4) is degenerated into an explicit method of the Runge – Kutta type of the form

$$y_{n+1} = y_n + (1 - 0.5/\beta_{43})k_1 + 0.5k_4/\beta_{43}.$$

Notice that the local error $\delta_{n,RK}$ of this scheme can be written in the form

$$\delta_{n,RK} = h^3[\varphi'^2\varphi / 6 +$$
$$+(1/6 - 0.25/\beta_{43})\varphi''\varphi^2] + O(h^4).$$

Hence, the local error of the explicit formula is minimal provided that $\beta_{43} = 2/3$. Finally, we have the coefficients of the second-order accuracy scheme (4), i. e.

$$a = 1 - \sqrt{2}/2, \quad \beta_{41} = \beta_{42} = 0, \quad \beta_{43} = 2/3,$$
$$p_4 = -p_1 = 3/4, \quad p_2 = a, \quad p_3 = 1 - a.$$

III. Calculation Accuracy Control

Calculation control for the scheme (4) is performed with a first-order accuracy method. With the help of the stages of (4), we can construct a family of first-order numerical formulae of the form

$$y_{n+1,1} = y_n + b_1k_1 + b_2k_2 + b_3k_3 + b_4k_4 + b_5k_5, \quad (7)$$

where $k_5 = hg(y_n)$, b_i are numerical coefficients. Using Taylor expansion of the stages, we see that (7) is of first-order accuracy provided that

$$b_1 + b_2 + b_3 + b_4 = 1$$

and

$$b_2 + b_3 + b_5 = 1.$$

Then an error estimate ε_n for the scheme (4) can be calculated by the formula

$$\varepsilon_n = y_{n+1} - y_{n+1,1}.$$

When choosing the coefficients b_i, $1 \le i \le 4$, one can be guided by different considerations. For instance, if the function $g(y)$ is completely responsible for stiffness of the problem (3), which is the case for many problems (2) for $B = \partial f(y)/\partial y$, then it makes sense to take the set of coefficients

$$b_1 + b_3 + b_4 + b_5 = 0 \text{ and } b_2 = 1$$

or

$$b_1 + b_2 + b_4 + b_5 = 0 \text{ and } b_3 = 1.$$

This technique of the error estimation is used with advantage when implementing the (2,1)-method with analytical calculation of the Jacobi matrix. However, if, for instance, in the problem (2) a diagonal approximation of the Jacobi matrix is used, then for many problems (3) we may not consider the function $\varphi(y)$ as a non-stiff part. In this case, such an estimate may result in the loss of calculation accuracy due to arising instability of the explicit part of the numerical formula (4). From this reasoning, in (7) the coefficients $b_1 + b_5 = 1$ and $b_2 = b_3 = b_5 = 0$ are taken. In this case, (7) is rearranged to the form

$$y_{n+1,1} = y_n + h[\varphi(y_n) + g(y_n)].$$

Numerical results show that the application of this scheme in the estimate results is more reliable control of calculation accuracy.

We point out an important feature of the proposed error estimate. From L-stability of the scheme (4), it follows that for the stability function $Q(x,z)$ we have $Q(x,z) \to 0$ as $z \to -\infty$. For the exact solution

$$y(t_{n+1}) = \exp(x + z)y(t_n)$$

of the problem (6) a similar property holds, hence, it is natural to require that the error estimate ε_n approaches zero as $z \to -\infty$. However, for the proposed estimate we have $\varepsilon_n = O(z)$. Thus, to improve the asymptotic behaviour, instead of ε_n we consider the estimates $\varepsilon_n(j_n)$ of the form $\varepsilon_n(j_n) = D_n^{1-jn}\varepsilon_n$, $1 \le j_n \le 3$. Observe, that in the sense of the principal term, i. e. the first term of Taylor expansion of an error in powers of h, the estimates ε_n and $\varepsilon_n(j_n)$ coincide for any value of j_n, besides, $\varepsilon_n(3) \to 0$ as $z \to -\infty$. Now, for calculation accuracy control we can use the inequality $\|\varepsilon_n(j_n)\| \le \varepsilon$, $1 \le j_n \le 3$, where ε is the required accuracy of calculations. Notice, that the use of $\varepsilon_n(j_n)$ instead of ε_n does not result in considerable increase of a computational cost. For $z \to 0$, the estimate $\varepsilon_n(1) = \varepsilon_n$ is in agreement with the behaviour of an error and there is no need to check it for other values of j_n. With the drastic increase of the step, the behaviour of ε_n may become inadequate which manifests itself in unreasonable decrease of the step and repeated calculations of a solution. Thus, when implementing an integration algorithm, the inequality for accuracy control is used as follows. For each n the smallest value of j_n which provides this inequality is taken. If it does not hold for any value of j_n, then the step decreases and the solution is recalculated.

IV. A Numerical Scheme for Non-Autonomous Problems

We consider the Cauchy problem for a non-autonomous system of the form

$$y' = \varphi(t,y) + g(t,y), \quad y(t_0) = y_0, \quad t_0 \le t \le t_k,$$

where y, φ, and g are real vector functions of dimension N and t is an independent variable.

Further, we again assume that the function $g(t, y)$ is completely responsible for the stiffness, and $\varphi(t, y)$ is the non-stiff part. For the numerical solution of a non-autonomous problem we consider a method of the form

$$y_{n+1} = y_n + p_1k_1 + \ldots + p_4k_4, \quad D = E - ahg'_n,$$

$$k_1 = h\varphi(t_n, y_n),$$

$$Dk_2 = h[\varphi(t_n, y_n) + g(t_n + ch, y_n)], \quad Dk_3 = k_2, \quad (8)$$

$$k_4 = hf(t_n + [\beta_{41} + \beta_{42} + \beta_{43}]h,$$
$$y_n + \beta_{41}k_1 + \beta_{42}k_2 + \beta_{43}k_3),$$

where E is the identity matrix; $g'_n = \partial g(t_n,y_n)/\partial y$; k_i are the stages of the method; c, a, p_i, β_{4j} are numerical coefficients.

To study the scheme (8), we use Taylor expansion of the stages k_i in powers of h up to the terms of order of h^2. Substituting these series in the first formula of (8), we get

$$y_{n+1} = y_n + (p_1 + p_2 + p_3 + p_4)h\varphi_n +$$
$$+ (p_2 + p_3)hg_n + (\beta_{41} + \beta_{42} + \beta_{43})p_4h^2\varphi'_{tn} +$$
$$+ c(p_2 + p_3)g'_{tn} + (\beta_{41} + \beta_{42} + \beta_{43})p_4h^2\varphi'_{yn}\varphi_n +$$
$$+ (\beta_{42} + \beta_{43})p_4h^2\varphi'_{yn}g_n + a(p_2 + 2p_3)h^2g'_{yn}\varphi_n +$$
$$+ a(p_2 + 2p_3)h^2g'_{yn}g_n + O(h^3),$$

where the elementary differentials are calculated at an approximate solution.

Taylor expansion of the exact solution $y(t_{n+1})$ in the neighbourhood of the point t_n has the form

$$y(t_{n+1}) = y(t_n) + h(\varphi + g) + \frac{1}{2}h^2(\varphi'_t + g'_t +$$
$$+ \varphi'_y\varphi + \varphi'_y g + g'_y\varphi + g'_y g) + O(h^3),$$

where the elementary differentials are calculated at the exact solution. Comparing the series for the condition $y_n = y(t_n)$, we arrive at the second-order accuracy conditions for the scheme (7), i. e.

$$p_1 + p_2 + p_3 + p_4 = 1, \quad p_2 + p_3 = 1,$$

$$(\beta_{41} + \beta_{42} + \beta_{43})p_4 = \frac{1}{2},$$

$$c(p_2 + p_3) = \frac{1}{2}, \quad (\beta_{42} + \beta_{43})p_4 = \frac{1}{2},$$

$$a(p_2 + 2p_3) = \frac{1}{2}.$$

This yields $c = 0.5$. With considerations similar to those for the scheme (4), we obtain the coefficients for the numerical formula (8) of the form

$$a = 1 - \frac{\sqrt{2}}{2}, \ \beta_{41} = \beta_{42} = 0, \ c = \frac{1}{2}, \ \beta_{43} = \frac{2}{3},$$

$$p_4 = -p_1 = \frac{3}{4}, \ p_2 = a, \ p_3 = 1 - a.$$

An inequality for calculation accuracy control is constructed similarly to that for the scheme (4), where in the estimate ε_n an approximation to a solution, obtained with the second-order accuracy method (8), and an approximate solution, calculated by a first-order method of the form

$$y_{n+1,1} = y_n + h[\varphi(t_n,y_n) + g(t_n + 0.5h,y_n)]$$

are used. The choice of the step with respect to the accuracy is performed in the same way as in the case of an autonomous system.

V. The Analysis of Numerical Results

In what follows, the proposed algorithm is called ASODE2. Freezing the Jacobi matrix, i.e. the use of the matrix

$$D_n = E - ahg'_{yn}$$

at several integration steps, is performed by the following rule. If the Jacobi matrix is not recalculated, then the integration step remains the same to keep the stability of the numerical scheme. An attempt to use the former matrix is performed after each successful integration step. The following three reasons result in unfreezing:

1) violation of an inequality for calculation accuracy control;

2) the number of steps with a frozen matrix exceeds i_h;

3) the forecasted integration step exceeds the last successful step by the factor of q_h.

The parameters i_h and q_h can be used to adjust the method to a specific problem. If $i_h \to \infty$ and $q_h \to \infty$, then the number of integration steps with one and the same Jacobi matrix increases. If $i_h = 0$ and $q_h = 0$, then the matrix is not frozen. Hence, for a large-scale system of ordinary differential equations, it makes sense to take i_h and q_h sufficiently large. In the numerical results presented below, $i_h = 20$ and $q_h = 2$.

All the examples below are rearranged to the form (2). The required calculation accuracy is $\varepsilon = 10^{-2}$. The calculations were performed with PC Intel(R) Core i7-3770S CPU@3.10GHz with double precision. The scheme (4) is of second-order accuracy, hence, there is no sense in higher-order precision in this case. The norm $\|\zeta\|$ in inequalities for accuracy control is calculated by the formula

$$\|\zeta\| = \max_{1\le i\le N}\left\{\frac{|\zeta_i|}{|y_n^i|+r}\right\},$$

where i is the number of components, and r is a positive parameter.

If the inequality $|y_n^i| < r$ holds for the i-th component of a solution, then an absolute error εr is controlled, otherwise the relative error ε is controlled. In the calculations, the parameter r is taken so that an actual accuracy for all components of the solution is not lower than the required one. Below is, if, idec, isol denote the total number of integration steps, of right-hand sides of the system (1), of decompositions of the Jacobi matrix, and of calculations of backward Gauss, respectively.

A. Example 1 [4]

$$y_1' = -0.013y_1 - 1000y_1y_3,\quad y_2' = -2500y_2y_3,$$

$$y_3' = -0.013y_1 - 1000y_1y_3 - 2500y_2y_3, \qquad (9)$$

$$t \in [0,50],\quad y_1(0) = 1,\quad y_2(0) = 1,\quad y_3(0) = 0,$$

$$h_0 = 2.9\cdot 10^{-4}.$$

The problem (9) is solved by the method (4) with a diagonal approximation of the Jacobi matrix, i. e. in the numerical formula (4) the diagonal Jacobi matrix with the diagonal entries b_{ii} of the form

$$b_{11} = -0.013 - 1000y_3;\ b_{22} = -2500y_3;$$

$$b_{33} = -1000y_1 - 2500y_2$$

is used.

Since in this case the computational cost of the method (4) is close to that of explicit methods, this method is compared in terms of efficiency with the well-known Merson method [3] of the fourth-order accuracy. Calculating an approximate solution with accuracy $\varepsilon = 10^{-2}$ by the ASODE2 algorithm requires 687 steps, the remaining costs are calculated from the form of the scheme (4). The solution of the problem by Merson method requires 400,627 calculations of the right-hand side. In the case of full Jacobi matrix of system (9), the ASODE2 algorithm without freezing Jacobi matrix for the problem (9) requires 38 steps, 38 decompositions of the matrix, and 108 calculations of backward Gauss. The remaining costs are calculated from the form of the scheme (4). In calculations with freezing the Jacobi matrix, the computational cost is as follows:

is = if = 98, idec = 15, isol = 288.

B. Example 2 [4]

$$y_1' = -55y_1 + 65y_2 - y_1y_2,$$

$$y_2' = 0.0785(y_1 - y_2),\quad y_3' = 0.1y_1, \qquad (10)$$

$$t \in [0,500],\quad y_1(0) = y_2(0) = 1,\quad y_3(0) = 0,$$

$$h_0 = 2\cdot 10^{-2}.$$

The problem (10) is solved by the method (4) with the diagonal approximation of Jacobi matrix where

$$b_{11} = -55 - y_3;\ b_{22} = -0.0785y_3;\ b_{33} = 0.$$

An approximate solution with accuracy $\varepsilon = 10^{-2}$ by the ASODE2 algorithm is calculated in 4,953 steps. The solution of the problem by Merson method requires 80,713 calculations of the right-hand side. In the case of the full Jacobi matrix of system (10), the ASODE2 algorithm without freezing Jacobi matrix requires 81 steps, 81 decompositions of Jacobi matrix, and 388 calculations of backward Gauss. In calculations with freezing the matrix, the computational cost is as follows:

is = if = 338, idec = 24, isol = 1,124.

C. Example 3 [8]

$$\begin{gathered} y_1' = 77.27[y_1(1 - 8.375\cdot 10^{-6} y_1 - y_2) + y_2], \\ y_2' = (y_3 - (1 + y_1)y_2)/77.27, \\ y_3' = 0.161(y_1 - y_3), \\ t \in [0, 360],\ \ y_1(0) = 1,\ \ y_2(0) = 2, \\ y_3(0) = 3,\ \ h_0 = 10^{-6}. \end{gathered} \tag{11}$$

The problem (11) is solved by the method (4) with the diagonal approximation of Jacobi matrix where

$$b_{11} = 77.27(1 - 1.675{\cdot}10^{-7}y_1 - y_2);$$
$$b_{22} = -(1+y_1)/77.27;\ b_{33} = -0.161.$$

An approximate solution with accuracy $\varepsilon = 10^{-2}$ is calculated by the ASODE2 algorithm in 19,964 steps. Solving the problem by Merson method requires 23 700,664 calculations of the right-hand side. In the case of the full Jacobi matrix of the system (11), the ASODE2 algorithm without freezing Jacobi matrix requires 2,449 steps, 2,652 decompositions of Jacobi matrix, and 6,964 calculations of backward Gauss. The computational cost of the calculations with freezing the matrix is as follows:

is = if = 19,807, idec = 3,431, isol = 50,924.

VI. Conclusion

The proposed integration algorithm serves for the numerical solution of the problems of mechanics of continua after the space discretization by the finite element or finite difference method. In this case, in the problem (3) splitting into the functions $g(y)$ and $\varphi(y)$ is natural, $g(y)$ is a symmetric part related to the second-order differentiation operator and $\varphi(y)$ is a nonsymmetric part (convective terms) related to the first-order differentiation operator. The implementation of the numerical formula (4) requires solving a linear system of algebraic equations twice. In the problems of mechanics of continua, efficiency of an integration algorithm can be improved by means of special methods for linear systems with a symmetric matrix that is positive in many cases.

The scheme (4) can also be applied to locally unstable problems. In this case $\varphi(y)$ is responsible for the eigenvalues of Jacobi matrix with a positive real part. Contrary to *A*-stable or *L*-stable methods which usually have a small instability domain and are *A*-stable or *L*-stable not only in the left half-plane but in the right half-plane of the plane $\{h\lambda\}$ as well, explicit methods of Runge – Kutta type are unstable practically in the whole right half-plane, hence, they are preferable for determining an unstable solution. For locally unstable problems, splitting the right-hand side of a system of ordinary differential equations into functions $\varphi(y)$ and $g(y)$ from physical considerations usually does not involve difficulties.

The presented numerical results are not oriented to the solution of the problems of mechanics of continua or locally unstable problems, but are related to the study of possibilities of the integration algorithm for some conventional test examples.

The test examples are taken in order to demonstrate different features of the integration algorithm. In the case of similar behavior of several test problems, the simplest example is taken.

The purpose of the calculations is to verify working efficiency of the algorithm with variable step and freezing Jacobi matrix, reliability of the inequality for computation accuracy control and to study possibility of calculations with a diagonal approximation of Jacobi matrix. In the last case the computational cost per step for the proposed algorithm is close to that for explicit methods. In particular, from the analysis of numerical results for stiff problems, it follows that, in the case where methods with unbounded stability domain may not be applied, the algorithm (4) is considerably more efficient than the Merson method being the most popular among explicit numerical schemes of Runge – Kutta type.

The work is supported by Russian Foundation of Fundamental Researches (projects 11-01-00106 and 11-01-00224).

REFERENCES

1. **Cooper G.J., Sayfy A.** Additive Runge – Kutta methods for stiff ordinary differential equations. *Mathematics of Computation*, 1983, Vol. 40, No. 161, pp. 207–218.

2. **Kaps P., Rentrop P.** Generalized Runge – Kutta methods of order four with step size control for stiff ordinary differential equations. *Numerical Math.*, 1979. No. 33, pp. 55–68.

3. **Merson R.H.** An operational methods for integration processes. *Proc. Symp. on Data Proc.* Weapons Research Establishment, Salisbury, Australia. 1957, pp. 329–330.

4. **Novikov E.A.** Explicit methods for stiff systems. Novosibirsk: Nauka, 1997. (rus)

5. **Novikov E.A., Shitov Yu.A., Shokin Yu.I.** One-step iteration-free methods for solving stiff systems. *Soviet Math. Dokl.*, 1989, Vol. 38, No. 1, pp. 212–216. (rus)

6. **Novikov E.A.** The additive third of order method for solving stiff of nonautonomous problems. *Journal of Applied and Industrial Mathematics*, 2010, Vol. 4, No. 4, pp. 1–12.

7. **Novikov E.A., Shornikov Yu.V.** Computer simulation of stiff hybrid systems. Novosibirsk: Publishing house NGTU, 2012, 450 p.

8. **Rosenbrock H.H.** General implicit processes for the numerical solution of differential equations. *Computer J.* 1963, No. 5, pp. 329–330.

9. **Hairer E., Wanner G.** Solving ordinary differential equations. Stiff and differential-algebraic problems. Berlin: Springer-Verlag, 1991. 528 p.

СПИСОК ЛИТЕРАТУРЫ

1. **Cooper G.J., Sayfy A.** Additive Runge – Kutta methods for stiff ordinary differential equations. *Mathematics of Computation*, 1983, Vol. 40, No. 161, pp. 207–218.

2. **Kaps P., Rentrop P.** Generalized Runge – Kutta methods of order four with step size control for stiff ordinary differential equations. *Numerical Math.*, 1979. No. 33, pp. 55–68.

3. **Merson R.H.** An operational methods for integration processes. *Proc. Symp. on Data Proc.* Weapons Research Establishment, Salisbury, Australia. 1957, pp. 329–330.

4. **Новиков Е.А.** Явные методы для жестких систем. Новосибирск: Наука, 1997. 192 с.

5. **Новиков Е.А., Шитов Ю.А., Шокин Ю.И.** Одношаговые безытерационные методы решения жестких систем // ДАН СССР. 1988. Т. 301. № 6. С. 1310–1314.

6. **Novikov E.A.** The additive third of order method for solving stiff of nonautonomous problems. *Journal of Applied and Industrial Mathematics*, 2010, Vol. 4, No. 4, pp. 1–12.

7. **Новиков Е.А., Шорников Ю.В.** Компьютерное моделирование жестких гибридных систем. Новосибирск: НГТУ, 2012. 450 с.

8. **Rosenbrock H.H.** General implicit processes for the numerical solution of differential equations. *Computer J.* 1963, No. 5, pp. 329–330.

9. **Hairer E., Wanner G.** Solving ordinary differential equations. Stiff and differential-algebraic problems. Berlin: Springer-Verlag, 1991. 528 p.

НОВИКОВ Евгений Александрович – *доктор физико-математических наук, профессор, главный научный сотрудник отдела вычислительной математики Института вычислительного моделирования Сибирского отделения РАН.*

660036, Россия, Красноярск, Академгородок, 50
novikov@icm.krasn.rr