NUMERIC MODELLING OF HYBRID SYSTEMS WITH THE SECOND ORDER L-STABLE (2,1)-METHOD IN ISMA INSTRUMENTAL ENVIRONMENT

Evgeny A. Novikov[1], Yury V. Shornikov[2]

[1]Institute of computational modelling SB RAS, Department of Calculus mathematics
Akademgorodok, Krasnoyarsk, Russia
[2]Novosibirsk State Technical University, Faculty of Automatics and computer facilities
20 K. Marx Prospect, Novosibirsk, Russia

shornikov@inbox.ru (Yury V. Shornikov)

Abstract

The class of (m,k)-methods is described. A second order L-stable method is created for resolving stiff autonomous problems. An algorithm for hybrid systems is developed that takes into account the guard condition when selecting the integration step. To show how the algorithm works a hybrid system of two oscillating masses on springs is considered.

Keywords: (m,k)-method, L-stability, accuracy monitoring, freezing of Jacobi matrix, hybrid problems.

Presenting Author's biography

Yury V. Shornikov. Works in Novosibirsk State Technical University (NSTU), department «Automated control systems», a Dr. Sci. Tech., the professor of department «Automated control systems» NSTU. The general scientifically-pedagogical experience (without including 6 years of postgraduate study and doctoral studies in NSTU) – 31 years.
Has more than 70 printing works, from them 16 in leading reviewed magazines Higher Certifying Commission, 9 works are registered in Russian Patent Bureau and Branch Fund of Algorithms and Programs, seven manuals and the textbook «The theory and practice of language processors», 2004 recommended to interuniversity use.
Scientific specialty is Mathematical and the software of computers, complexes and computer networks, a scientific direction is computer modeling, scientific interests are programming languages, program tool means, mathematical and the software.

1 Introduction

Existence of events, meaning the existence of points of discontinuity in first derivatives of phase variables [1–3], is a feature of many engineering problems. Such combined discrete-continuous system are called hybrid or systems with switching [1]. Events fire in time moments matching to zeros of some algebraic function there. In many cases the situation is aggravated by stiffness and large size of a problem under consideration. When solving a Cauchy task for stiff systems of ordinary differential equations of large dimension, decomposition of Jacobi matrix consumes the main computing resources [4–6]. Therefore efficiency of the integration algorithm can be significantly enhanced by freezing Jacobi matrix, that is, by applying one matrix for several integration steps [7].

The second order of precision L-stable (2,1) method is developed here with freezing of both analytical, and numerical Jacobi matrix. The method is applicable to solving hybrid problems, including stiff ones.

2 The second order of accuracy L-stable method

A class of (m,k)-methods is proposed in [5–6] for numerical resolving of Cauchy task for stiff systems of ordinary differential equations

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

where y and f are real N-dimensional vector-functions, t is an independent variable. From the point of view of computer implementation (m,k)-methods are as simple as Rosenbrock type schemas [8]. However, unlike Rosenbrock type methods, it is significantly easier to carry out the Jacobi matrix freezing task in this class, as well as to make its numerical approximation. Besides, (m,k)-methods have better properties of accuracy and stability with a slight increase of computing resources. Using of an autonomous task (1) does not reduce commonness, because a non-autonomous task can always be transformed into the autonomous one by implementing an additional variable.

The (m,k)-methods class is defined as follows. Let positive integer numbers m and k be set up, $k \le m$. Let us denote a set of integers i, $1 \le i \le m$ as M_m, and as M_k and J_i we denote the subsets of M_m, $2 \le i \le m$, like

$$M_k = \{m_i \in M_m \mid 1 = m_1 < m_2 < \cdots < m_k \le m\},$$
$$J_i = \{m_{j-1} \in M_m \mid j > 1, m_j \in M_k, m_j \le i\},\ 2 \le i \le m.$$

Then (m,k)-methods can be presented as [5]

$$\begin{aligned} y_{n+1} &= y_n + \sum_{i=1}^{m} p_i k_i, \\ D_n &= E - ahf_n', \\ D_n k_i &= hf\Big(y_n + \sum_{j=1}^{i-1} \beta_{ij} k_j\Big) + \\ &\quad + \sum_{j \in J_i} \alpha_{ij} k_j, \quad i \in M_k, \\ D_n k_i &= k_{i-1} + \sum_{j \in J_i} \alpha_{ij} k_j,\ i \in M_m \setminus M_k \end{aligned} \qquad (2)$$

The set J_i, $2 \le i \le m$, is intended to eliminate "extra" α_{ij} coefficients, which cannot affect the properties of the accuracy and stability (2) and which are linearly expressed through other coefficients. Note that consideration of $\sum_{i=1}^{j-1} \alpha_{ij} k_j$ instead of $\sum_{j \in J_i} \alpha_{ij} k_j$ leads to significant difficulties when building specific numerical formulas just because of the existence of the "extra" coefficients. Note also that one constant m, a number of stages, will be enough in traditional one-step methods to describe computational cost per step of integration, because in these methods each stage is accompanied by mandatory calculation of the right part of the task (1). There are two kinds of stages in the methods (2). Some require calculation of the right part, and the others do not. As a result, in (2) description of the computational cost per step requires two constants, m and k. Cost per step is as follows. Jacobi matrix is calculated once and decomposition of matrix D_n is carried out. The function f is calculated k times, and m times the reverse in the Gauss method is carried out. In the case of $k = m$ and $\alpha_{ij} = 0$ numerical schemes (2) match with Rosenbrock type methods [4, 8]. In remaining cases these are the other methods with better properties.

To solve the problem (1) let us consider the (2,1) schema looking like

$$\begin{aligned} &y_{n+1} = y_n + p_1 k_1 + p_2 k_2, \\ &D_n k_1 = hf(y_n), \quad D_n k_2 = k_1, \end{aligned} \qquad (3)$$

where k_1 and k_2 are the stages of the method; $D_n = E - ahA_n$, E is a single matrix, h is step, A_n is some matrix presentable in the form $A_n = f_n' + hB_n + O(h^2)$, $f_n' = \partial f(y_n)/\partial y$ is the Jacobi matrix of the system (1), B_n is step independent arbitrary matrix, and p_1 and p_2 are some numerical coefficients. Using of matrix A_n in the above presented form allows to apply (3) with the freezing both analytical and numerical Jacobi matrix [7]. In the case of usage of the Jacobi matrix f_{n-k}'

calculated k steps ago, we have the $B_n = -kf''_n f_n$, $f''_n = \partial^2 f(y_n)/\partial y^2$. If Jacobi matrix is calculated numerically with step $r_j = c_j h$, where c_j are some constants, then elements $b_{n,ij}$ of matrix B_n look like $b_{n,ij} = 0.5c_j \partial^2 f_i(y_n)/\partial y_j^2$. In calculations the numerical differentiation step r_j is defined according to the formula $r_j = \max\left(10^{-14}, 10^{-7}\left|y_j\right|\right)$.

We shall get coefficients of the second order L-stable numerical schema (3) and inequality for the calculation accuracy monitoring. Decomposition of the exact solution $y(t_{n+1})$ into Taylor sequence around the point t_n up to members with h^3 has a form

$$y(t_{n+1}) = y(t_n) + hf + 0.5h^2 f'f + \\ + [h^3/6](f'^2 f + f''f^2) + O(h^4), \quad (4)$$

where elementary differentials f, $f'f$, $f'^2 f$ and $f''f^2$ are calculated on the exact solution $y(t_n)$. To find coefficients a, p_1 and p_2 of the schema (3) we write decompositions of stages k_1 and k_2 into Taylor sequences around the point y_n up to members with h^3 inclusively and put into (3). Obtain

$$y_{n+1} = y_n + (p_1 + p_2)hf_n + a(p_1 + \\ + 2p_2)h^2 f'_n f_n + a^2(p_1 + 3p_2)h^3 f'^2_n f_n + \\ + a(p_1 + 2p_2)h^3 B_n f_n + O(h^4), \quad (5)$$

where elementary differentials f_n, $f'_n f_n$, $f'^2_n f_n$, $f''_n f_n^2$ and $B_n f_n$ are calculated on the approximate solution y_n. Assuming $y_n = y(t_n)$ and comparing (4) and (5) up to members with h^2 inclusively, we get the second order of accuracy conditions for the schema (3), that is

$$p_1 + p_2 = 1, ap_1 + 2ap_2 = 0.5. \quad (6)$$

Let us explore the stability of numerical formula (3). Applying it to the problem $y' = \lambda y$, $y(0) = y_0$, $\mathrm{Re}(\lambda) < 0$, we get $y_{n+1} = Q(x)y_n$, $x = h\lambda$, where the function of stability is $Q(x) = \left[1 + (p_1 + p_2 - 2a)x + a(a - p_1)x^2\right]/(1 - ax)^2$. Then the schema (3) will be L-stable if $p_1 = a$. Substituting this equation in (6), we get a set of coefficients $p_1 = a$, and $p_2 = 1 - a$, where a is determined from the condition of L-stability $a^2 - 2a + 0.5 = 0$. Comparing (4) and (5) up to members with h^3 inclusively we get that the local error δ_n of the numerical scheme (3) has the form

$$\delta_n = h^3[(a - 1/3)f'^2 f + f''f^2/6 - \\ - 0.5B_n f] + O(h^4) \quad (7)$$

Equation $a^2 - 2a + 0.5 = 0$ has two roots $a_1 = 1 - 0.5\sqrt{2}$ and $a_2 = 1 + 0.5\sqrt{2}$. Select $a = a_1$, as in this case the coefficient in the main member $(a - 1/3)h^3 f'^2 f$ of error (7) is less.

Monitoring of calculation accuracy of the numerical scheme (3) we build on similar to [9]. We implement a denomination for this purpose $v(j_n) = D_n^{1-j_n}(k_2 - k_1)$, where k_1 and k_2 are calculated on formulas (3). Then according to [9] the inequality

$$\| v(j_n) \| \le \varepsilon, \quad 1 \le j_n \le 2, \quad (8)$$

is needed to be checked on each step for the calculation accuracy monitoring, where ε is the required precision of calculations, $\|\bullet\|$ is some norm in R^N, and the integer j_n is the smallest one, which meets this inequality.

3 Hybrid system

Let us consider the Cauchy task for system ordinary differential equations of type

$$y' = f(y), y(t_0) = y_0, g(y,t) \le 0, \quad (9)$$

where $g(y,t)$ – guard condition or nonlinear protector. Because many models of interest are linear ones, we shall consider them as most important class of guard conditions. Note that any nonlinear protector can be led to the linear view appending additional phase variable $x = g(y,t)$. As a result, task (9) can be rewritten as

$$y' = f(y), \\ x' = \frac{\partial g}{\partial y} f(y) + \frac{\partial g}{\partial t}, \ x \le 0. \quad (10)$$

Assumption here is that the guard condition is linear. For simplicity we will assume hereinafter that the original problem is scalar. However, all of the following considerations apply to the systems. Particular attention should be paid to the choice of method of integration. A fully implicit method cannot be used because it requires the calculation of $f(y)$ in potentially dangerous area, that is, where the model is not defined. Explicit methods are known as those of low stability. Therefore here we will use L-stable methods. For example, consider the implicit Euler method, which for the task (1) is

$y_{n+1} = y_n + hf(y_{n+1})$. In this case the guard condition dynamics is described by the expression

$$g_{n+1} = g\left(y_n + h^p_{n+1} f_{n+1}, t_n + h^p_{n+1}\right)$$

where $f_{n+1} = f(y_{n+1})$, h^p_{n+1} foreseeable step of integration. Note that f_{n+1} is calculated at a potentially dangerous point. Therefore, we will check the guard condition dynamics with the first order of accuracy (1,1)-method of $y_{n+1,1} = y_n + k_1$, where k_1 is defined in numerical formula (3). It is easy enough to see that Taylor sequence for $y_{n+1,1}$ has a view of $y_{n+1,1} = y_n + h^p_{n+1} f_n + O\left(h^p_{n+1}\right)^2$. As a result function g_{n+1} is

$$g_{n+1} = g\left(y_n + h^p_{n+1} f_n + O\left(h^p_{n+1}\right)^2, t_n + h^p_{n+1}\right).$$

Decomposing g_{n+1} into the Taylor sequence around the point (y_n, t_n) and given the linearity of $g(y,t)$, we have

$$g_{n+1} = g_n + h^p_{n+1}\left(\frac{\partial g_n}{\partial y} \cdot f_n + \frac{\partial g_n}{\partial t}\right), \qquad (11)$$

where

$$g_n = g(y_n, t_n), \quad \frac{\partial g_n}{\partial y} = \frac{\partial g(y_n, t_n)}{\partial y},$$
$$\frac{\partial g_n}{\partial t} = \frac{\partial g(y_n, t_n)}{\partial t}.$$

Eventually we received dependency of g_{n+1} on the foreseeable step h^p_{n+1}.

Theorem [1]. The selection of step by the formula

$$h^p_{n+1} = \frac{(\gamma - 1) g_n}{\frac{\partial g_n}{\partial y} \cdot f_n + \frac{\partial g_n}{\partial t}} \qquad (12)$$

where $\gamma \in [0, 1)$ provides behaviour of guard condition dynamics as of a stable linear system approaching the surface $g(y,t) = 0$. Besides, if $g(y_0, t_0) < 0$, then $g(y_n, t_n) \le 0$ for every n.

Proof. Substituting (12) in (11), we have $g_{n+1} = \gamma g_n$. Converting recurrently this expression we get $g_{n+1} = \gamma^{n+1} g_0$. Given that $\gamma < 1$, then $g_n \to 0$ takes place when $n \to \infty$. Besides, from ratio $\gamma \ge 0$ follows that function g_n does not change its sign. Therefore, when $g_0 < 0$, $g_n \le 0$ will be valid for every n. Then the guard condition will never cross potentially dangerous area $g(y_n, t_n) = 0$, that completes the proof.

Let the solution y_n in the point tn be calculated with step h_n. Besides, values of method (3) stages k_1 and k_2 are known. Then the integration algorithm taking into account the forecast of step through the guard condition is as follows:

1. $g_n = g(y_n, t_n)$, $\partial g_n / \partial y = \partial g(y_n, t_n)/\partial y$, $\partial g_n / \partial t = \partial g(y_n, t_n)/\partial t$ are calculated.

2. Step h^p_{n+1} is calculated by formula (12).

3. A new step h_{n+1} is calculated by the formula $h_{n+1} = \min\left(h^p_{n+1}, h^{pr}_{n+1}\right)$, where h^{pr}_{n+1} is found from inequality (8). As soon as $v(j_n) = O\left(h^2\right)$, then foreseeable step h^{pr}_{n+1} on accuracy will be defined by formula $h^{pr}_{n+1} = q h_n$, where q is set up by the equation $q^2 v(j_n) = \varepsilon$.

4. Runs the next step of integration.

To show how the algorithm works a typical hybrid system of two oscillating masses on springs is considered [11]. System can be in one of the two local states: "Separately" and "Together". A System of algebra-differential equations describes behaviour of the system in each of the states.

Provided $s < abs\left(k_1 n_1 - k_2 n_2 - x_1(k_1 - k_2)\right)$ we have:

$$\begin{aligned} x_1' &= v_1; \\ v_v' &= k_1(n_1 - x_1)/m_1; \\ a_1 &= k_1(n_1 - x_1)/m_1; \\ x_2' &= v_2; \\ v_2' &= k_2(n_2 - x_2)/m_2; \\ a_2 &= k_2(n_2 - x_2)/m_2. \end{aligned} \qquad (13)$$

Provided $(x_1 = x_2)\ and\ (v_1 > v_2)$ we have:

$$\begin{aligned} a_1 &= \left(k_1 n_1 + k_2 n_2 - x_1(k_1 + k_2)\right)/(m_1 + m_2); \\ v_1' &= \left(k_1 n_1 + k_2 n_2 - x_1(k_1 + k_2)\right)/(m_1 + m_2); \\ x_1' &= v_1; \\ a_2 &= \left(k_1 n_1 + k_2 n_2 - x_1(k_1 + k_2)\right)/(m_1 + m_2); \\ v_2' &= \left(k_1 n_1 + k_2 n_2 - x_1(k_1 + k_2)\right)/(m_1 + m_2); \\ x_2' &= v_2; \\ s' &= -s, \end{aligned} \qquad (14)$$

where m_1, m_2 are masses of loads; k_1, k_2 are stiffness of springs; n_1, n_2 are neutral coordinates of

loads; x_1, x_2 are coordinates of loads; v_1, v_2 are speeds of loads; a_1, a_2 are accelerations of loads; s is total stiffness of springs in the state "Together".

The computer model of system of two oscillating masses on springs, written down in language LISMA [12], is presented in Fig. 1.

```
k1=1; k2=2; // stiffness of springs
n1=1; n2=2; // neutral coordinates of loads
m1=1; m2=1; // masses of loads
x1=0;
x2=3; // initial coordinates
separate [s < abs(k1*n1-k2*n2-x1*(k1-k2))] is
 s~=10;
 x1'=v1;
 v1'=k1*(n1-x1)/m1;
 a1~=k1*(n1-x1)/m1;
 x2'=v2;
 v2'=k2*(n2-x2)/m2;
 a2~=k2*(n2-x2)/m2;
from;
together [ (x1>=x2) and (v1>=v2) ] is
 s=10;
 v1=(m1*v1+m2*v2)/(m1+m2);
 v2=v1;
 v1'=(k1*n1+k2*n2-x1*(k1+k2))/(m1+m2);
 a1~=(k1*n1+k2*n2-x1*(k1+k2))/(m1+m2);
 x1'=v1;
 v2'=(k1*n1+k2*n2-x2*(k1+k2))/(m1+m2);
 a2~=(k1*n1+k2*n2-x2*(k1+k2))/(m1+m2);
 x2'=v2;
 s'=-s;
from separate;
```

Fig. 1 Computer model of the hybrid two-mass system in LISMA

Results of analysis computer model by ISMA instrumental environment [11] with the detection algorithm developed (Fig. 2) match with results of calculation of the reference model in the system HyVisual [10].

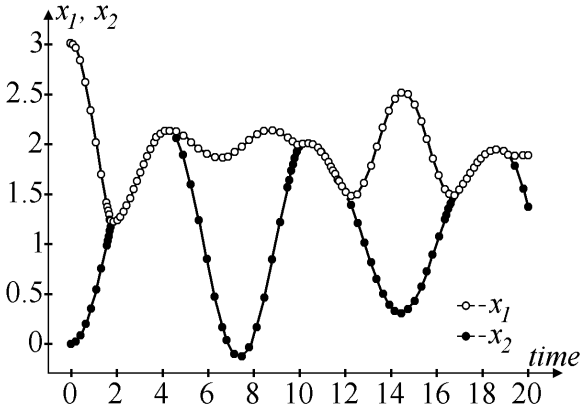

Fig. 2 Dynamic of the hybrid two-mass system

Traditional analysis of the system without detection algorithm leads to low-quality results (Fig. 3).

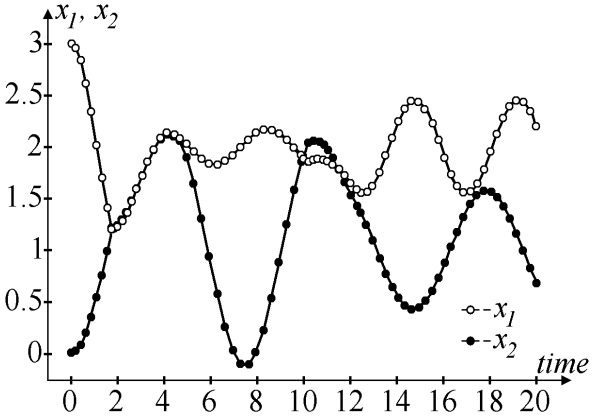

Fig. 3 Results of analysis two-mass system without detection algorithm

4 Conclusion

The proposed way of predicting the step taking into account guard conditions extends to all (m,k)-schemas and Rosenbrock type methods. For prediction of step taking into account the guard condition the second order of accuracy L-stable method (3) can be used, because it needs information only at the current point. Constructively proved efficiency of the developed and integrated into ISMA environment algorithms.

5 References

[1] J. Esposito, V. Kumar, G. J. Pappas. Accurate event detection for simulating hybrid systems. In: Hybrid systems: Computation and control (HSCC). Volume LNCS 2034. Springer-Verlag. 1998.

[2] E.A. Novikov, Yu.V. Shornikov. Numeric modelling of hybrid systems by the second order of accuracy Runge-Kutta methods. *Computing Technology*, 2, t. 13:98-104. 2008.

[3] A.E. Novikov, E.A. Novikov, Yu.V. Shornikov. Modelling of hybrid systems by explicit methods. *Messenger Krasgau: Resource-conserving technology of agricultural mechanization*, 5:88-92. 2009.

[4] E. Hairer, G. Wanner. Solution of ordinary differential equations. Stiff and differential-algebraic problems. Moscow, Mir. 1999.

[5] E.A. Novikov, Yu.A. Shytov, Yu.I. Shokhin. One-step non-iterative methods of solving stiff systems. *DAN USSR*, 6, t. 301:1310-1314. 1988.

[6] E.A. Novikov, Yu.A. Shytov, Yu.I. Shokhin. On a class of (m,k)-methods of solving stiff systems. *Siberian Journal of Numerical Mathematics*, 2, t.29:194-201, 1989.

[7] E.A. Novikov, Yu.A. Shytov. The algorithm of integrating stiff systems based on the second order of accuracy (m,k)-method with numeric

calculation of Jacobi matrix. Preprint No. 20. USSR, Krasnoyarsk. 1988.

[8] H.H. Rosenbrock Some general implicit processes for the numerical solution of differential equations. *Computer*, 5:329-330, 1963.

[9] E.A. Novikov. Explicit methods for stiff systems. Novosibirsk, Science. 1997.

[10] E.A. Lee, H. Zhenq. Operational Semantics of Hybrid Systems. Proc. of Hybrid systems: Computational and control (HSCC) LNCS 3414. Zürich, Switzeland. 2005.

[11] Yu.V. Shornikov. Modelling of complex dynamic and hybrid systems in ISMA. *Scientific Messenger NSTU*, 1 (26):79-88, 2007.

[12] Yu.V. Shornikov, I.N. Tomilov. The program of language processor from language LISMA. Official registration license for computers № 2007611024. Moscow, Rospatent. 2007.