Computer-aided simulation of high-dimensional event-continuous systems

Yu V Shornikov and E A Popov
Automated Control Systems Department, Novosibirsk State Technical University, 20 Prospekt K. Markska, Novosibirsk, 630073, Russia

E-mail: shornikov@inbox.ru

Abstract. A comparative analysis of efficiency of explicit methods with error and stability control, semi-implicit (m, k)-methods, and implicit methods implemented in the modelling and simulation environment ISMA is presented. The Akzo Nobel Central Research laboratory's model of a tumour penetrated by antibodies is used as the test problem. The computer model in the LISMA language after applying the method of lines is given. Performance of the numerical integration algorithms is measured for different spatial variable discretization steps.

1. Introduction
Event-continuous (ECSs) or hybrid dynamical systems [1–3] are characterized by having discontinuities of the first kind in the simulated time interval [0; T]. This feature is the reason why the Lipschitz conditions are not met, and application of symbolical analysis methods to systems of this class, especially high-dimensional, is therefore limited. In order to obtain an appropriate numerical solution, one has to use not only traditional and original numerical integration algorithms, but also effective algorithms [4, 5] detecting events at discrete time instants \( t^* \in [0; T] \). The integration step size in this case is computed taking into account the error, stability, and event function behaviour [5, 6]. Such computational routines form the basis of libraries of numerical algorithms in the architecture of modern computer modelling and simulation environments for complex dynamic systems. Another important aspect of studying ECSs is linguistic software supporting specialized visual and textual specification languages for describing mathematical models.

Continuous modes of ECSs, especially heterogeneous ECSs, that is combining processes of different physical nature, are often modelled by high-dimensional stiff ordinary differential equations (ODEs) [7, 8].

Let us conduct a comparative analysis of some traditional and original numerical methods, included in the library of the software environment ISMA [5], on event continuous problems of high dimension. ISMA is designed for modelling and simulation of complex dynamic processes from different applications, including chemical kinetics and power engineering problems. Problems are internally represented by ISMA as ECSs.

2. Mathematical model
Let us consider the Akzo Nobel Central Research laboratory's model [9] of a tumour penetrated by antibodies. The model is represented as the boundary value problem for a system of partial differential equations
Applying the method of lines, the initial conditions are defined separately from the initial values of the tissue, the spatial variable, and the reaction rate constant; $u, v$ are the concentrations of the radio-labelled antibodies and the tissue respectively, $x$ is the spatial variable,

$$
\phi(t) = \begin{cases} 
\phi_1(t), & t \in \left[0, t^*\right), \\
\phi_2(t), & t \in \left[t^*, T\right].
\end{cases}
$$

Let $\zeta = \frac{x}{x+c}$, $c > 0$. Applying the method of lines, we have

$$
\frac{dy}{dt} = f(t, y),
$$

where $y = (u, v, \ldots, u_N, v_N)^T$, $g = (0, v_0, 0, v_0, \ldots, 0, v_0)^T$, the functions $f(t, y)$ are defined as

$$
f_{2j-1} = \alpha_j \frac{y_{2j-1} - y_{2j-3}}{2\Delta\zeta} + \beta_j \frac{y_{2j+1} - y_{2j-1}}{(\Delta\zeta)^2} - k y_{2j-1} y_{2j},
$$

$$
f_{2j} = -k y_{2j} y_{2j+1},
$$

where

$$
\alpha_j = 2 \left( j\Delta\zeta - 1 \right)^3 / c^2, \quad \beta_j = \left( j\Delta\zeta - 1 \right)^3 / c^2, \quad 1 \leq j \leq N,
$$

$$
\Delta\zeta = \frac{1}{N}, \quad y_{-1}(t) = \phi(t), \quad y_{2N+1} = y_{2N-1}, \quad g \in \mathbb{R}^{2N}.
$$

The parameter values used further on are $k = 100$, $v_0 = 1$, $t^* = 5$, $T = 20$, $\phi_1(t) = 2$, $\phi_2(t) = 0$ and $c = 4$.

3. Computer model

The computer model of (2), (3) in the LISMA language is presented in figure 1.

Lines 1 and 2 define the model constants.

The right-hand side of (2) is algorithmically specified according to the language syntax and features of the hybrid problem.

The $2(N-2)$ differential equations are defined with a for cycle and the index notation. These language constructs allow to conveniently specify systems of equations, having a certain structure, and their initial conditions.

The 6 remaining equations, having special right-hand sides, and corresponding initial conditions are defined separately by lines 4–9 and 22–25.

Equations are specified in the LISMA language in a form similar to that of the language of mathematics. The prime symbol naturally denotes derivatives of functions.

Lines 27–32 describe the piecewise continuous function $\phi(t)$ by defining its form in the initial mode init and introducing a second mode activated at $t = 5$, where the function form is different.

4. Numerical results

It is possible to obtain systems of ODEs involving an arbitrary even number of differential equations $N = 2N$ by varying the parameter $N$. Let us consider the following 5 cases: $N = \{50, 100, 200, 400, 500\}$. Let the required accuracy of computations be $\epsilon = 10^{-3}$, the initial integration step size be $h_0 = 10^4$. 
The library of ISMA includes explicit, semi-implicit, and implicit algorithms of variable step. The purpose of this paper is to show how some of them behave as the dimension of the problem to be solved grows. For the comparison, the following algorithms were chosen:

1. RKF78st, an explicit modified eight-stage Fehlberg method of order 7 with stability control;
2. DP78st, an explicit modified eight-stage Dormand-Prince method of order 7 with stability control;
3. RKF78vo, an integration algorithm of variable order based on the explicit modified eight-stage 7th order Fehlberg method and an explicit eight-stage 1st order Runge-Kutta method with an extended stability interval;
4. MK22, a semi-implicit L-stable two-stage 2nd order method of the Rosenbrock type allowing to freeze the Jacobian matrix;
5. MK21, a semi-implicit L-stable two-stage 2nd order (2, 1)-method;
6. RADAU5, an implicit L-stable three-stage 5th order method of the Runge-Kutta type.

For the explicit methods, the maximum eigenvalue is estimated on each step using the power method, and the predicted step is computed as

\[
\text{max}(\lambda_\text{ac}, \text{min}(\lambda_\text{ac}, \lambda_\text{nn})) \times h_{\text{ac}}
\]

(4)

where \( h_{\text{ac}} \) is the last successful integration stepsize, \( h_{\text{ac}} \) and \( h_{\text{nn}} \) are the stepizes computed according to the requirements of accuracy and stability, respectively [6]. Formula (4) allows to stabilize the stepsize behaviour in settling regions, where the requirements of stability come into contradiction with those of accuracy. The plots of the solution components \( y_{79}, y_{172}, y_{199} \) and function \( \varphi(t) \) are given in figure 2.

**Figure 1.** Computer model in LISMA.
The computation time against the dimension of the system to be solved for the aforementioned methods is depicted in figure 3.

The semi-implicit non-iterative one-step \((m, k)\)-methods are the most efficient as the numerical results show. They have better stability properties than explicit Runge-Kutta methods. Their advantage over well-known implicit methods is that they allow to "freeze" the Jacobian matrix (i.e. to use the same matrix during several integration steps). This feature results in a significant reduction of computational costs, especially if the system to be solved involves a large number of equations. Moreover, \((m, k)\)-methods do not require running an iteration process that makes algorithms based on \((m, k)\)-methods more efficient and reliable.

**Figure 2.** Solution components \(y_{79}\), \(y_{199}\), \(y_{172}\) and function \(\phi(t)\).

**Figure 3.** Logarithmic dependence of the computation time on the problem dimension.
However, note that the use of explicit algorithms of variable order with stability control of the numerical formula is highly efficient as well. In figure 2 it can be seen that the time the implicit method spends on solving the problem almost exponentially depends on the dimension. On the other hand, the dependencies of the computation time on the system dimension for (m, k)-methods and RKF78vo are closer to linear.

Explicit methods with extended stability domains are the most efficient if the system to be solved involves less than 500 equations. In case of a higher dimension, the application of semi-implicit (m, k)-schemes is more preferable. Employing implicit methods is reasonable in case of smaller systems of differential equations (100-200 equations) since, as the problem dimension grows, the computational costs are almost entirely determined by the time taken to decompose the Jacobian matrix.

An integration algorithm should be always chosen taking into account the problem at hand. The presented results are only valid for the simulated problem and should be extrapolated to other problems with caution for their efficiency might be different.

References
[1] Goebel R, Sanfelice R G and Teel A R 2012 Hybrid Dynamical Systems: Modeling, Stability, and Robustness (Princeton, USA: Princeton University Press)
[2] Schauf A J van der and Schumacher H 2000 An Introduction to Hybrid Dynamical Systems (Springer-Verlag London)
[3] Urquia Moraleda A and Martín Villalba C 2018 Modeling and Simulation in Engineering Using Modelica (Madrid, Spain: UNED Editorial)
[4] Esposito J M and Kumar V 2007 A state event detection algorithm for numerically simulating hybrid systems with model singularities ACM Transactions on Modeling and Computer Simulation 17(1) 1
[5] Shornikov Yu V and Dostovalov D N 2018 Fundamentals of Event-Continuous System Simulation Theory (Novosibirsk, Russia: NSTU Publisher)
[6] Novikov E A and Novikov A E 2014 An Algorithm of Variable Order and Step, Based on the Stages of the Dormand-Prince Method Humanities and Science University Journal 10 140-7
[7] Butcher J C 2016 Numerical Methods for Ordinary Differential Equations (UK: Wiley)
[8] Hairer E and Wanner G 1996 Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems (Berlin, Germany: Springer-Verlag)
[9] Mazzia F and Magherini C 2008 Test set for initial value problem solvers, release 2.4, Report 4/2008 (Bari, Italy: University of Bari)