The basic set of test problems for ODE system solvers

Vladimir Manichev, Dmitriy Zhuk and Emilii Feldman
BMSTU, Robotics and Complex Automation Department, Computer Aided Design and Engineering, 105005 Moscow, Russia
feldlime@yandex.ru

Abstract. Known mathematical packages, MATLAB, Maple, Mathematica, MathCAD and others may get wrong, often plausible, the result of numerical solution of ODE systems with low, given by default, the requirements for mathematical accuracy of the results of the numerical solution of ODE systems. Since ODE system parameters obtained usually experimental, with a low mathematical precision, so the requirements for the precision of the results of mathematical solutions of ODE systems is low (for example, in MATLAB package required precision is 0.001). The article offers a basic set of tests to assess the range of applicability of the relevant solvers. The basic set of test problems for ODE solvers systems include linear ODE systems with the known analytic solution and nonlinear systems with known graphics solution. Presented comparative results of solutions for proposed problems using MATLAB solvers and manzhuk program from a library of standard mathematical programs SADEL (Sets of Algebraic and Differential Equations solvers Library), which has been designed for reliable and accurate solving of systems of linear algebraic equations (LAE) and ODE systems. The results can be used in mathematical modeling of dynamic systems described by ODE systems.

1. Introduction
The programs for solving ODE systems use three classical problem statements for solving ODE systems:
1. The Cauchy normal form in the space of differential state variables (explicit form) [1]:
\[ \frac{dX}{dt} = F(X, t) \]
Where \( X \) is the vector of the coordinate basis of differential variables of the state of dimension \( m \); \( F \) is the vector function of the right parts of the dimension \( m \); \( t \) is the independent variable (usually time).
Initial conditions specified as \( X_0 = X(T_0) \), time of integration is \( t = [T_0, T_E] \), \( T_0 \) is the time of the beginning the integration, \( T_E \) – time of completion of integration.
2. Differential-algebraic form of different indices in the full coordinate basis (semi-explicit form) [2, 3]:
\[ \begin{cases} \frac{dX}{dt} = F(X, Y, t) \\ G(X, Y) = 0 \end{cases} \]
Where \( Y \) – vector of algebraic variables of the full coordinate basis of differential-algebraic variables of dimension \( k \), \( G \) is the vector function of dimension \( k \). The agreed initial conditions are now specified as \( X_0 = X(T_0) \), \( Y_0 = Y(T_0) \). This statement (1) of the task of ODE systems solutions are
widely used in the mathematical and numerical modeling automatic regulation and control systems, for example, MATLAB-SIMULINK [4], MVTU [5, 6] and others.

3. The differential-algebraic form in the full coordinate basis, i.e. system of differential-algebraic equations (DAE) (implicit form):

$$G(X, dX/dt, Y, t) = 0$$

(2)

Where $G$ – vector function of dimension $(m + k)$. The initial conditions are the same as in the previous case. Statement (2) is used in mathematical and numerical modeling of electronic and mechanical engineering systems, for example, the ODE systems solvers SPICE and ADAMS. The task (2) in the space of differential algebraic variables was set in the full coordinate basis by L. Petzold in 1982 and she developed the DASS (Differential Algebraic Systems Solver) program based on the Gear method with replacement of derivatives according to the Gear formula [7]:

$$dX/dt = BDF(X, h)$$

Where $BDF$ – Backward Differential Formula, $h$ – integration step. The resulting system of nonlinear algebraic equations was solved by the Newton method at each integration step.

The task of solving ODE systems in the extended space of differential-algebraic variables without transformations of the initial ODE systems is based on the statement (2) of the problem and has the form:

$$G(X, PX, Y, t) = 0$$

(3)

Here $PX = dX/dt$ – vector of derivatives of differential state variables with respect to time of dimension $m$.

Considering the fact that some matrix-vector transformations on a computer are not equivalent due to the limited bit grid of computers, the classical forms of representation of ODE systems should be used mainly for theoretical studies, provided that all transformations of the original DAE systems are performed analytically or in symbolic form.

The stability domain of any explicit method is limited, therefore, implicit methods are recommended for solving hard problems [1, 2]. But implicit methods also have their drawbacks. Using the implicit method for solving stiff and locally unstable problems may lead to qualitatively wrong solution [8]. It is undesirable to use explicit methods for modeling hybrid dynamic systems containing discrete and relay elements, since in this case the Jacobi matrix must be updated frequently. Finally, implementing implicit methods is much more complicated than explicit ones. Therefore, attempts to build explicit methods that are effective for hard problems do not stop [8]. This article discusses the basic set of test ODE systems, which includes both stiff and nonstiff ODE systems and comparing solvers of MATLAB with manzhuk program from the library of standard mathematical programs SADEL in this set of tests.

2. Methods for solving ODE-DAE systems

There are implemented 8 methods for solving ODE-DAE systems in the MATLAB. The choice of methods for solving ODE-DAE systems in the SADEL library is explained below.

Since the explicit methods for solving systems of ODE and DAE are not universal and do not meet the requirements of the stability in the solution with variable step integration of the stiff ODE systems, then as basic integration methods for SADEL library were selected implicit methods of integration [2]. Implicit methods for solving ODE systems are reduced to iteratively solving nonlinear algebraic systems. In the software implementation of these methods, two main types of errors in the numerical solution of ODE systems should be distinguished – global error (or qualitative error) and local error (or quantitative error).

In theoretical terms, the properties of ODE-DAE systems and methods for their solution are considered in relation to linear inhomogeneous ODE systems of the form:

$$dX/dt = A * X + H(t)$$

(4)
Where $A$ is a constant real matrix with size $m \times m$, $H(t)$ is a known vector function of time. If the integration method is not suitable for solving the system (4), it is unsuitable for system (3) too. Denote the determinant of the matrix $A$ through $\det(A)$. The spectrum of eigenvalues is the set of all eigenvalues $\lambda_i$ of matrix $A$, $i = 1, 2, ..., m$. In the general case $\lambda_i = \text{Re}(\lambda_i) + j \cdot \text{Im}(\lambda_i)$ is a complex value. It is known that if the eigenvalues are different, then the general solution of system (4) will be a linear combination of fundamental solutions $c \cdot e^{\text{Re}(\lambda_i)t} \cdot \cos(\text{Im}(\lambda_i)t)$ and $c \cdot e^{\text{Re}(\lambda_i)t} \cdot \sin(\text{Im}(\lambda_i)t)$, $c$ is some constant. The analytical solution of system (4) can be represented as the sum of the individual fundamental solutions of a homogeneous system of ordinary differential equations with a matrix $A$ and the private solution of the heterogeneous system (4):

$$X(t) = \sum_{i=1}^{m} c_i e^{\lambda_i t} X_i + Z(t)$$

Where $c_i$ – constant values; $X_i$ – own vector of matrix $A$ corresponding to the eigenvalue $\lambda_i$; $Z(t)$ – private solution of the system (4). For these solutions five areas of the complex plane $\lambda_i$ reflecting the five classes of local problems of integration of differential equations (Figure 1) can be distinguished. In general, in a wide variety of ODE-DAE systems can be isolated five base classes of tasks for basic solutions of linear homogeneous ODE shown in Figure 1 and requiring the use of various methods for solving.

![Figure 1](image-url)

**Figure 1.** Five base classes for fundamental solutions of linear homogeneous systems of ODE.

The classical control of the local integration error will ensure the accuracy of the solution, and to control the global integration error it is necessary to ensure the appropriate stability of the integration methods. To ensure obtaining a qualitatively correct solution for the above classes of problems, the integration method must be stable for stable ODE systems (I and II classes of tasks) and should be unstable for unstable ODE systems (IV and V classes of problems), therefore it is necessary that the integration methods be appropriately AL-stable (AL-stability – Absolute stability in Left half-plane of the complex stability plane of numerical integration), i.e. absolutely stable strictly in the left half-plane of the complex plane of stability of the integration methods [9].

With low requirements to the precision of solving ODE systems, this area of absolute stability guarantees the preservation of a qualitatively correct solution with increasing integration step in the plane $h\lambda$, since with low integration accuracy the integration step can become so large that non-AL-stable integration methods can obtain incorrect solution trajectories. Implicit one-step Runge-Kutta integration methods meet the AL-stability requirement [10]. Algorithms for the implementation of implicit one-step integration methods of Runge-Kutta type are based on the joint solution of systems of nonlinear algebraic equations $F(X,Y,PX,t) = 0$ for solved DAE system of the form (3) and systems of linear algebraic equations for integration methods $H(X,PX,h_t) = 0$, formed at the corresponding stages of the integration methods, relative to differential-algebraic variables $X,Y$ and derivatives of differential variables $PX$ [11]. Were developed DAbc formulas (as the development of Abc Butcher methods [12]) for S-phasic implicit methods for solving DAE systems in an extended coordinate basis:

$$H_{ii}(PX_{i,i}, X_{i,i}, X_{i-1}, PX_{i-1}, h_t) = h_i \sum_{j=1}^{s} d_{ij} PX_j - \sum_{j=1}^{s} a_{ij} X_j - b_{ij} X_{i-1} - h_i c_{ij} PX_{i-1} = 0$$
\[ X_i = X_s, PX_i = PX_s, i = 1, \ldots, s, t_i = t_s \]

Where \( s \) – the number of stages of the corresponding one-step method; \( h_i \) – \( i \)-th integration step; \( d_{ij} \), \( a_{ij}, b_{ij}, c_{ij} \) – method parameters. Based on these formulas, three integration methods were developed and implemented:

- **M1** – A-stable one-step implicit method of the first order of accuracy;
- **M2** – AL-stable one-step implicit method of the second order of accuracy;
- **M3** – AL-stable two-stage implicit method of the fourth order of accuracy.

The main drawback of AL-stable methods is that they do not satisfy the requirements of \( L \)-stability and sometimes give "false fluctuations" in the solution, so these methods are not widely used in practice [2]. The solution to this problem is given in [10].

3. **Basic set of test ODE systems**

3.1. **Test problems with a well-known analytical solution**

It is proposed to develop 5 test linear homogeneous ODE systems 1) – 5) with analytical solutions corresponding to the classes of tasks shown in Figure 1 as the basic set of test problems. Given the test linear system of ODE of the 2nd order:

\[ \frac{dX}{dt} = AX + B \]

Where \( X \) is the vector of unknowns of system, \( A \) is the matrix of constant coefficients of a system of size 2×2, \( B \) is a constant vector.

3.2. **Test problems with a graphical solution:**

6) An example of the calculation of a tough second-order ODE system (MU is a stiffness parameter) – Van der Pol's test. Correct solution for this test shown in Figure 2.

\[
\begin{align*}
\frac{dx_1}{dt} &= x_2 \\
\frac{dx_2}{dt} &= -x_1 + MU \times (1 - x_1^2) \times x_2 \\
x_1(0) &= -1, x_2(0) = 1 \\
t \in [0, 4.2MU]
\end{align*}
\]

**Figure 2.** Reliable solution results for test 6.
7) Example of electronic circuit simulation with a multi-period solution (5th order ODE system).

![Electronic Circuit Diagram](image)

Figure 3. Electronic circuit for test 7.

The ODE system for this test is:

\[
\begin{align*}
kr &= \frac{k_u}{k_l}, \\
kc &= kt * \frac{ki}{k_u}, \\
kl &= kt * \frac{ku}{ki} \\
\frac{dx_1}{dt} &= x_4 * kc \\
\frac{dx_2}{dt} &= x_5 * kc \\
\frac{dx_3}{dt} &= x_4 - x_5 \\
\frac{dx_4}{dt} &= ku - x_1 - x_3 - kr * x_4 * kl \\
\frac{dx_5}{dt} &= \frac{1001}{999} x_2 + x_3 - kr * x_5 * kl \\
x_1(0) &= 0, x_2(0) = 0, x_3(0) = 0 \\
x_4(0) &= 0, x_5(0) = 0 \\
t &\in [0, 12560 * kt]
\end{align*}
\]

The output voltage corresponds to \( x_5(t) \).

![Phase Portrait](image)

Figure 4. Correct solution for test 7.

8) Nonlinear stiff ODE system for mathematical modeling of real laser processes.

9) Nonlinear stiff ODE system for mathematical modeling of real laser processes.
\[
\begin{align*}
\frac{dx_1}{dt} &= -x_1 \cdot (\alpha \cdot x_2 + \beta) + \gamma \\
\frac{dx_2}{dt} &= x_2 \cdot (p \cdot x_1 - \sigma) + \tau \cdot (1 + x_1) \\
x_1(0) &= -1, x_2(0) = 0 \\
t \in [0, 10^6] \\
\alpha = 1.5 \times 10^{-18}, \beta = 2.5 \times 10^{-6}, \gamma = 2.1 \times 10^{-6}, \\
p = 0.6, \sigma = 0.18, \tau = 0.016
\end{align*}
\]

Figure 5. The correct solution for \(x_2(t)\) in test 8.

Waivers for problems with graphical solution are time of calculation more than 10 seconds or the wrong plot of the solution.

4. Calculation results

4.1. Test problems with a well-known analytical solution:

Failures for tasks with an analytical solution were recorded if the maximum relative error was more than 0.001 in a given integration interval. These tasks were used to test manzhuk solver and different MATLAB solvers.

The final test results are shown in Table 1.

Table 1. The final results of the tests with known analytical solution.

| Program for solving ODE systems | Tests |
|---------------------------------|-------|
|                                 | 1)    | 2) | 3) | 4) | 5) |
| manzhuk (M3)                   | +     | +  | +  | +  | +  |
| MATLAB ode45                    | +     | +  | -  | -  | -  |
| MATLAB ode23                    | +     | +  | -  | -  | -  |
| MATLAB ode113                   | +     | +  | -  | -  | -  |
| MATLAB ode15s                   | +     | +  | -  | -  | -  |
| MATLAB ode23s                   | +     | +  | -  | -  | -  |
| MATLAB ode23t                   | +     | +  | -  | -  | -  |
| MATLAB ode23tb                  | +     | +  | -  | -  | -  |
| MATLAB ode15i                   | +     | +  | -  | +  | +  |
Sign (+) means a correct solution of the test with accuracy $\text{eps} = 0.001$.
Sign (-) means failure with accuracy $\text{eps} = 0.001$.
It should be noted that the qualitative solution for all MATLAB methods was correct. Maximum errors were obtained on test 3.

### 4.2. Test problems with a graphical solution

In view of the results published in [10, 13, 14], Table 2 shows the summary comparison solvers of stiff ODE systems of the aforementioned SADEL library, from well-known libraries and packages of mathematical programs.

**Table 2. The results of the tests with known graphical solution.**

| Program for solving ODE systems | Tests | 6) | 6) | 7) | 7) | 8) |
|---------------------------------|-------|----|----|----|----|----|
|                                 | MU= 1e6 | MU=1e9 | Kt=1 | Kt=1e-103 | The real laser parameters |
| manzhuk (M3)                    | +      | +   | +   | +   | -   | +++|
| NAG C-Library                   | -      | -   | -   | -   | +   |   |
| IMSL C-Library                  | -      | -   | -   | -   | -   |   |
| Mathcad                         | +      | -   | +++ | -   | +   |   |
| MATLAB ode15s                   | +      | -   | +++ | -   | +   |   |
| Maple                           | -      | -   | +   | +   | -   | +++|
| Mathematica                     | +      | -   | +   | -   | -   | +   |
| SPICE-Multisim                  | -      | -   | +++ | -   | +   | -   |
| OrCAD- PSPICE                   | -      | -   | +   | -   | -   | -   |
| SPICE-SYMICA                    | -      | ND  | +   | -   | ND  | -   |
| OpenModelica                    | -      | -   | -   | -   | -   | -   |

The sign (+) means the correct solution of the test using the method recommended for solving stiff ODE systems, with an accuracy $\text{eps} = 0.001$.
The sign (-) indicates an incorrect test solution using the method recommended for solving hard ODE systems with default parameters without warning the user and refusing to get the right solution.
The sign (+) means an incorrect test solution by the method recommended for solving hard ODE systems, with default parameters without warning the user and obtaining the correct solution by setting appropriate methods and integration parameters and increasing the calculation time.
The sign (-++) means an incorrect test solution by the method recommended for solving hard ODE systems, with default parameters without warning the user and obtaining the correct solution by setting only the parameter specified accuracy of the method of integration and increasing the calculation time. ND – no data.

Figure 6 shows as an example the result of a qualitatively incorrect solution for the 1st variable of test 7) by the implicit method ode15s and correct solution by method M3 of program manzhuk. There were no warnings about the incorrect decision in the ode15s solver.
Figure 6. Solution for the 1st variable of test 7): by ode15s; by method M3 of program manzhuk (right).

5. Conclusion
Most of the problems of solving systems of ordinary differential equations arising in practice cannot be solved analytically [15]. Numerical methods descend to the seventeenth century – to Newton, who proposed using them to calculate the trajectories of comets. Manual calculations continued to be used until the 60s of the twentieth century, before the appearance of the first solver programs written in FORTRAN for the first computers [16]. Since that time, development of test problems for solving ODE systems has been carried out. Classical sets of test considered tests from books [1, 2, 17]. The first test comparison of the MATLAB ODE solvers is given in [18]. In MATLAB we can always choose the method and adjust its parameters to obtain qualitatively correct results, but most users are not experts in numerical methods, so the comparison should be made for the recommended default parameters of integration ODE solvers.
For specific classes of problems to be solved, it is necessary to have sets of test practical and mathematical problems with a known deliberately reliable and accurate solution for the correct selection and adjustment of the corresponding solvers of ODE systems and the correct evaluation of the obtained results of numerical simulation.
This article presents the results obtained in the continuation of the above works. Further studies focused on developing effective methods and algorithms for the automatic integration step selection for AL-stable implicit methods for the numerical solution of high- and ultra-high-dimensional ODE-DAE systems and testing these new solvers. The relevance and importance of these works is indicated in the [19, 20].

References
[1] E. Hairer, S.P. Norsett, G. Wanner, Solving ordinary differential equations I. Nonstiff Problems. Springer, 528 (2008)
[2] E. Hairer, G. Wanner, Solving ordinary differential equations II. Stiff and Differential-Algebraic Problems. Springer, 614 (2010)
[3] A.D. Polyanin, I.K. Shingareva, Non-linear blow-up problems for systems of ODEs and PDEs: Non-local transformations, numerical and exact solutions. International Journal of Non-Linear Mechanics 111, p. 28-41 (2019)
[4] I.V. Chernykh, Modeling of electrical devices in MATLAB, SimPowerSystems and Simulink. 288 (2007)
[5] O.S. Kozlov, D.E. Kondakov, L.M. Skvortsov, K.A. Timofeev, V.V. Khodakovsky, The program complex “Modeling in technical devices” (PC "MVTU"), http://model.exponenta.ru/mvtu/20050615.html
[6] O.S Kozlov, L.M. Skvortsov, MVTU software package in scientific research and applied developments. Mathematical Models and Computer Simulations 8(4), p. 358-368 (2016)
[7] G. Mao, L.R. Petzold, Efficient integration over discontinuities for differential-algebraic
systems. Computers & Mathematics with Applications, 43, Issues 1–2, p. 65-79 (January 2002)

[8] L.M. Skvortsov. The explicit multi-step method for the numerical solution of stiff differential equations // Journal of Computational Mathematics and Mathematical Physics. T. 47, No. 6. p. 959-967 (2007)

[9] Y.A. Evstifeev, V.B. Manichev. An effective A-stable method for integrating ordinary differential equations for electronic circuit analysis programs. Radio electronics. T.29, № 11. p. 31–35 (1986)

[10] D.M. Zhuk, V.B. Manichev, A.O. Ilnitsky. Methods and algorithms for solving differential-algebraic equations for modeling systems and objects in the time domain. Information Technologies. №7, part1. p. 16–24 (2010)

[11] L.M. Skvortsov, Implicit Runge–Kutta Methods with Explicit Internal Stages. Computational Mathematics and Mathematical Physics 58(3), p. 307-321 (2018)

[12] J.C. Butcher, Numerical methods for ordinary differential equations. John Wiley & Sons Ltd, 484 (2008)

[13] D.M. Zhuk, V.B. Manichev, M.K. Sakharov, SADEL as a library of “ultra-precise” solvers for the PA10 software package (SADEL-PA10). MES, p. 147-153 (2012)

[14] V.B. Manichev, D.M. Zhuk, F.A. Vityukov, Method of mathematical testing of the analysis of transient processes in CAD electronic. MES, p. 166-172 (2014)

[15] V.N. Bakulin, Y.I. Vinogradov, Analytical and asymptotic solution of boundary value problems in the mechanics of deformed shells under concentrated loading. Russian Aeronautics 60(1), p. 13–20 (2017)

[16] D. Kahaner, C. Moler, S. Nash, Numerical methods and software, Prentice Hall, 494 (1989)

[17] F. Mazzia, C. Magherini, F. Iavernaro, Test set for initial value problem solvers. http://pitagora.dm.uniba.it/~testset

[18] O.S. Kozlov, L.M. Skvortsov, Test comparison of MATLAB ODE Solvers. IPU RAS Publishing House, p. 53–60 (2002)

[19] Y.P. Petrov, Ensuring the reliability and robustness of computer calculations. BHV-Petersburg, 160 (2008)

[20] E.F. Mitenkova, E.V. Solov’eva, V.B. Manichev, E.O. Fel’dman, Isotopic Kinetics Problems on a Complete Elemental Basis of the Fission-Product Yield. Atomic Energy 124(1), p. 69-73 (2018)