European Symposium on Computer Aided Process Engineering – 15
L. Puigjaner and A. Espuña (Editors)

Construction of a Web–Based Library for Testing the Performance of Numerical Software for ODEs

Tova Alva[a], Mordechai Shacham[a*], Neima Brauner[b] and Michael B. Cutlip[c]
[a]Dept. Chem. Eng., Ben-Gurion University
Beer-Sheva, 84105, Israel
[b]School of Engineering, Tel-Aviv University,
Tel-Aviv, Israel
[c]Dept. Chem. Eng., University of Connecticut
Storrs, CT 06269, USA

Abstract

The development of an Ordinary Differential Equation (ODE) solver test library presents several challenges, as there is no accepted technique to verify the correctness of the solution obtained by a particular solver. A comparison of results obtained by different algorithms is also difficult due to the differences in the reporting intervals. In this paper we show that solving an ODE with both stiff and non-stiff algorithms can provide a validation to the correctness of the solution. The results obtained by different algorithms can be compared based on initial, minimal, maximal and final values of the differential variables. Cubic spline interpolation can be used to compare integrated values at equal intervals. In the example presented, the need and the potential benefits of an ODE test library are demonstrated. It is shown that even a widely used and well tested algorithm may yield incorrect or even absurd results under certain circumstances.

Keywords: ODE solver, test problem library, solution verification

1. Introduction

Steady state and dynamic simulations of chemical processes typically require numerical solutions of large systems of nonlinear algebraic equations (NLE), ordinary differential equations (ODE), and systems of differential-algebraic equations (DAE). It is often assumed by the user that the numerical solver will find the correct solution for a particular problem, or it will issue warning messages if there are uncertainties regarding the accuracy and/or correctness of the solution. Unfortunately, the solution provided by the numerical solver cannot always be trusted (e.g. Shacham, et al., 1995). The user of software packages also needs some reference concerning the error tolerance and the integration algorithm that should be used for a particular problem as inappropriate

[*]Author to whom correspondence should be addressed: shacham@bgumail.bgu.ac.il

values may cause exorbitantly long integration times in case of too tight tolerances, or incorrect or even absurd results in case of loose tolerances.

Confidence in the software's capability to correctly solve most problems can be gained by evaluation with a large set of test problems containing examples that are known to be difficult to solve. A set of test problems can also serve as reference for the software users to help with selection of the error tolerances and an appropriate integration algorithm. We have previously developed a web-based library for testing performance of NLE solvers (Shacham et al., 2002). This library has already proven its value as it is being used for rigorous evaluation of new algorithms for solving NLEs (see, for example, Lucia and Feng, 2003 and Bellavia et al., 2004).

In this paper we propose a technique for comparison of results obtained by different ODE solvers and a method for validation of these results. Necessary information is described for each library problem that is essential for comprehensive testing of the algorithms.

A practical example is used to demonstrate the benefits of the proposed techniques and the ODE library. The Polymath[1] and MATLAB[2] software packages were used to carry out the calculations reported in this paper.

2. The Information Included in the Library

The library contains information related to the problem definition and the solution trajectory of the variables that are defined by a differential equation. The model equations of the problem are stored in the same form as the input for a numerical solver (Polymath), in order to prevent introduction of typographical and other errors.

The correctness of the problem solution is verified by at least one non-stiff (such as the 4th order Runge-Kutta algorithm with error estimation and step size control) and one stiff algorithm. If the difference between the two solutions does not exceed 1% at any point, the solution is considered as correct. This solution verification is employed as the probability that the two algorithms will yield the same erroneous solution is very small because the algorithms are based on different principles. The initial, maximal, minimal and final values of the variables are compared first. Additional comparison is done using the variable values at equal intervals. Since the intervals at which the variable values are reported by the solution algorithms can be very different, the cubic spline interpolation is used to calculate the values at equal intervals. The tabular presentation of those values is included in the library as "residual plots" that display the difference between the variable values that were calculated using the two algorithms.

The number of function evaluations required to reach the solution using the different algorithms is also included so that the users can compare the efficiency of the various methods. The dependent variables are plotted as a function of the independent variables using all the integration step values of the Runge-Kutta algorithm. Stiff algorithms can

[1] Polymath is copyrighted by M. Shacham, M.B.Cutlip and M. Elly (http://www.polymath-software.com)

[2] MATLAB is a trademark of The Math Works, Inc. (http://www.mathworks.com)

use fewer steps to yield correct results at a particular point, but these may not follow correctly the curvature of the function (as was demonstrated by Shacham et al. 1995).

3. Multi-component, Semi-Batch Steam Distillation – an Example

This example is based on the work of Prenosil (1976). A simplified model and its solution are also presented by Ingham et al (1994). The process involves steam distillation of an *n*-octane and *n*-decane mixture. The organic phase is initially loaded into the still and is heated by a continuous feed of steam. The complete Polymath equation input and the MATLAB programs for the solution of this problem are presented in Appendix A[3].

There are two separate periods in the operation of the still. The first is the heating period where the sum of the vapour mole fractions of the three compounds (n-octane, n-decane and water) is less than one. In this period the indicator *boil* = 0. When the difference between the sum of the vapour mole fractions and 1 (unity) becomes less than 0.0001, the indicator *boil* gets a value greater than one. The differential equations included in the model are the mass balances on liquid *n*-octane, *n*-decane and water in the still and the equation that represents the change of the temperature in the still. The change of temperature in the still is represented by two different equations. For the heating period the energy balance equation is used for this purpose, while for the distillation period the temperature is changed in order to follow the bubble point curve using the controlled integration technique of Shacham et al.(1996). For this particular case the controlled integration technique can be described as follows. Let $\varepsilon = 1 - \sum_{i=1}^{3} y_i$, where y_i is the mole fraction of component *i* in the vapour phase. The controlled integration is initiated when ε approaches zero ($\varepsilon < 10^{-5}$). In order to prevent increase of the absolute value of ε, the temperature in the still is adjusted in a controlled manner, $dT / dt = K_c \varepsilon$, where K_c is the controller proportional gain.

The change of the mass of the water in the still is also represented by two equations. For the heating period it is assumed that the steam is completely condensed, thus the rate of change of the water mass is equal to the steam feed flow rate. For the distillation period, the rate of water evaporation is subtracted from the steam flow rate. Definition and description of the various variables and equations included in this problem are provided in Appendix A.

There are two potential causes for difficulties in the numerical solution of this problem. The first is the transition point from the heating period to the distillation period where two of the model equations are changed. Here the algorithm for adjustment of the integration step, which monitors the local truncation error, must reduce the step-size to a very small value to compensate for the non-smoothness of the functions. After passing this point, the step size must be increased to its normal value. Simple step-size adjustment algorithms (e.g. halving the step size when the error exceeds the desired tolerance) may require hundreds of function evaluations to pass the transition point. The

[3] Appendix A can be downloaded from the site ftp://ftp.bgu.ac.il/shacham/SemiBatch

other difficulty is related to the value of K_c, the proportional gain in the controlled integration scheme. Larger K_c, enables obtaining more precise temperature values (smaller $|\varepsilon|$), but makes the equation more stiff, thus forcing smaller step sizes.

This system of equations was solved using Polymath and MATLAB, with several different integration algorithms. The results obtained were very close in all cases except one case. Therefore, only the MATLAB solutions will be discussed in what follows.

In the MATLAB solution, the integration was carried out from t = 0 min up to t = 30 min, using K_c =10000. The integration algorithms that were used are *ode45*, *ode15s* and *ode23s*. The *ode45* is based on an explicit Runge-Kutta (4,5) formula, the Dormand-Prince pair (Dormand and Prince, 1980). The *ode15s* is a variable order solver based on the numerical differentiation formulas (NDFs). Optionally, it uses the backward differentiation formulas (BDFs), also known as Gear's method (Shampine and Reichelt ,1997). The *ode23s* is based on a modified Rosenbrock formula of order 2. Because it is a one-step solver, it may be more efficient than ode15s at crude tolerances. It can solve several types of stiff problems for which ode15s is not effective (Shampine and Reichelt ,1997).

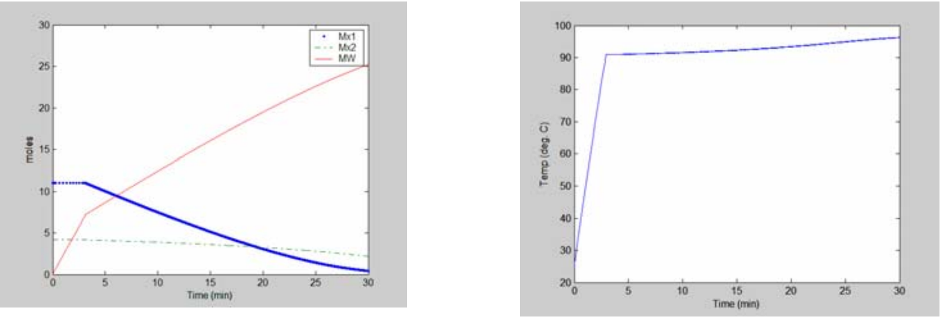
Table 2. Partial results, using the ode45 and ode23s routines, for the steam distillation example.

Value	Initial	Minimal		Maximal		Final	
Variable		ode45	ode23s	ode45	ode23s	ode45	ode23s
T	25	25	25	95.36	95.34	95.36	95.34
Mx_1	10.991	0.389	0.389	10.991	10.991	0.389	0.389
Mx_2	4.169	2.18	2.18	4.169	4.169	2.18	2.18
M_W	0	0	0	25.203	25.204	25.203	25.204

Partial results for the steam distillation problem are shown in Table 2. The initial, minimal, maximal and final values for the temperature, the mass of n-octane, n-decane and water in the still are shown. The differences between the results obtained by the two techniques are less than 0.5%; therefore, the solutions are essentially the same. To investigate the efficiency of the various methods, integration statistics can be examined. For the ode45 technique, some 3790 integration steps were required, while for ode23s only 144 steps were required. Thus for this problem, the efficiency of the stiff technique is higher by more than an order of magnitude.

Plots of the temperature in the still, and the masses of n-octane, n-decane and water are shown in Figures 1and 2 respectively. During the heating period, the temperature changes linearly with time. When boiling starts there is a sharp reduction in the slope and the line becomes more curved. The mass of liquid in the still increases continuously. However, the slope is reduced when boiling starts and there the plot becomes more curved toward the end of the distillation. This is associated with a decrease of the mass of *n*-octane and of *n*-decane, with the more volatile (n-octane) being depleted much faster. The n-octane's concentration in the liquid becomes lower than that of *n*-decane after 19 minutes.

Further validation of these results can be achieved by plotting the differences between the values of the various variables obtained by the *ode45* and the *ode23s* algorithms at equal time intervals. Cubic spline extrapolations are used to generate the variable values at equal time intervals of one minute in this case. A plot of the temperature differences obtained by the two integration techniques vs. the temperature obtained by the ode45 method is shown in Figure 3. While above 90 °C (near boiling temperature) the differences become somewhat larger, they do not exceed 0.1% at any point. This reconfirms that the two solutions are practically identical.



(a) (b)

Figure 1.Mass of n-octane (Mx1), n-decane (Mx2) and water (MW)(a) and temperature (b) in the distillation still

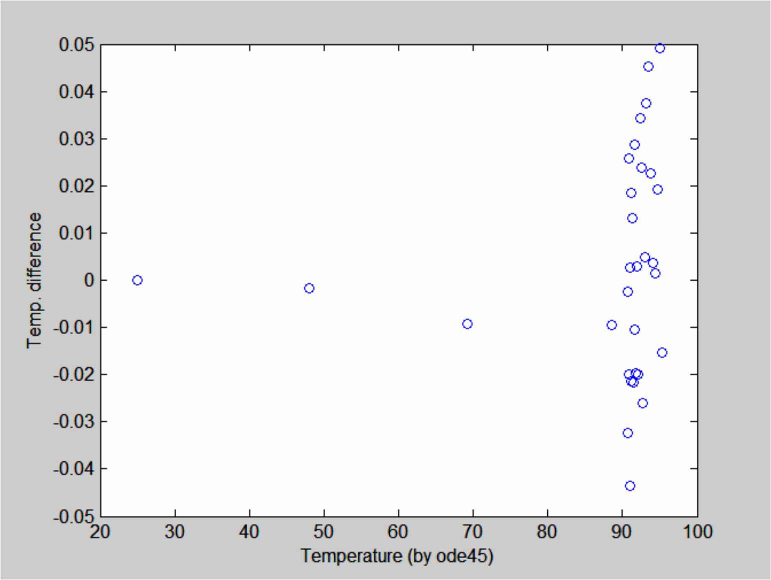


Figure 3. Difference between solutions obtained by ode45 and ode23s algorithms for the temperature in the still

The proposed testing of various algorithms is further demonstrated by solving the same problem using the *ode15s* routine. The results summarized in Table 3 clearly indicate that this algorithm yields an incorrect solution. The comparison in Table 3 indicates that most values obtained by *ode15s* are very similar to those obtained by *ode45*, except that the minimal temperature and water mass values are both negative. While the inability of the ode15s routine to solve correctly the problem requires further investigation, this example demonstrates very clearly the need to use a test problem library with appropriate test procedures in order to detect such failures.

Table 3. Partial results, using the ode45 and ode15s routines, for the steam distillation example

Value	Initial	Minimal		Maximal		Final	
Variable		ode45	ode15s	ode45	ode15s	ode45	ode15s
T	25	25	-84.95	95.36	96.72	95.36	96.72
Mx_1	10.991	0.389	9.15e-5	10.991	10.991	0.389	9.15e-5
Mx_2	4.169	2.18	0.4265	4.169	4.169	2.18	0.4265
M_W	0	0	-8.6885	25.203	7.187	25.203	5.52

4. Conclusions

Testing of algorithms for the solution of ODE problems can be achieved by

- Validating the correct solution of ODE problems using both stiff and non-stiff algorithms
- Reporting initial, minimal, maximal and final values for the differential variables
- Plotting the difference between the values obtained at equal intervals by different techniques
- Plotting the dependent variable values versus the independent variable

The particular example in this paper demonstrates the need for an ODE solver test library, as even a widely accepted and well-tested algorithm is shown to yield incorrect or even absurd results under certain conditions. Work is in progress for developing such a library based on the evaluation principles described in this paper.

References

Bellavia, S., Macconi, M. and Morini, B., 2004, Comput. Optim. Appl. 28(1), 31.

Dormand, J. R. and P. J. Prince, 1980, J. Comp. Appl. Math, 6, 19..

Ingham, J. Dunn,I. J., Heizle, E. and J.E. Prenosil, 1994, Chemical Engineering Dynamics, VCH, Weinheim, pp. 616-622.

Lucia, A. and Feng, Y., 2003, AIChE J, 49(10), 2553.

Prenosil, J. E., 1976, The Chemical Engineering Journal, 12, 59.

Shacham, M., N. Brauner and M. Pozin, 1995, Acta Chimica Slovenica, 42(1), 119.

Shacham, M., N. Brauner, and M. Pozin, 1996, Computers Chem. Engng. 20(Suppl), s1329.

Shacham, M., N. Brauner and M. B. Cutlip, 2002, Computers Chem. Engng. 26(4-5), 547.

Shampine, L. F. and M. W. Reichelt, 1997, SIAM Journal on Scientific Computing, 18, 1.