Research Journal of Applied Sciences, Engineering and Technology 7(13): 2634-2638, 2014
DOI:10.19026/rjaset.7.579
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: June 17, 2013 Accepted: July 03, 2013 Published: April 05, 2014

Research Article

Evaluation of MATLAB Methods used to Solve Second Order Linear ODE

Yasin Al-Hasan
Albalqa Applied University, Amman, Jordan, Tel.: 00962777481434

Abstract: Most second-order Ordinary Differential Equations (ODES) arising in realistic applications such as applied mathematics, physics, metrology and engineering. All of these disciplines are concerned with the properties of differential equations of various types. ODES cannot be solved exactly. For these problems one does a qualitative analysis to get a rough idea of the behavior of the solution. Then a numerical method is employed to get an accurate solution. In this way, one can verify the answer obtained from the numerical method by comparing it to the answer obtained from qualitative analysis. In a few fortunate cases a second-order ode can be solved exactly. Because of the big efforts needed to solve second order linear ODES, some MATLAB methods were investigated, the result of these methods were studied and some judgment were done regarding the results accuracy and implementation time.

Keywords: Homogenous, inhomogeneous, MATLAB method, modeling, ODES

INTRODUCTION

Many physical phenomena and processes are modeled by second order ODE's such as Mechanical Systems/Vibrations (springs, pendulums, etc.), Electrical Circuits and One Dimensional Motion (Bryant *et al*., 2008a, b; Fels and Olver, 1997).

Most second-order ODES arising in realistic applications cannot be solved exactly. For these problems one does a qualitative analysis to get a rough idea of the behavior of the solution (Kruglikov, 2008). Then a numerical method is employed to get an accurate solution. In this way, one can verify the answer obtained from the numerical method by comparing it to the answer obtained from qualitative analysis. In a few fortunate cases a second-order ode can be solved exactly. So it is important to minimize the efforts needed to get the exact solution of second order ODE and to minimize the time needed to solve such equations.

The general second-order linear differential equation with independent variable t and dependent variable $x = x\ (t)$ is given by Eq. (1):

$$\ddot{x} + p\ (t)\ \dot{x} + q\ (t)\ x = g\ (t) \tag{1}$$

where, we have used the standard physics notation $\dot{x} = dx/dt$ and $\ddot{x} = d2x/dt2$.

A unique solution of (1) requires initial values $(t0) = x0$ and $\dot{x}\ (t0) = u0$.

The equation with constant coefficients-on which we will devote considerable effort-assumes that $p\ (t)$ and $q\ (t)$ are constants, independent of time. The second-order linear ode is said to be homogeneous if g $(t) = 0$ (Kruglikov and Lychagin, 2006a; Nurowski and Sparling, 2003; Yumaguzhin, 2008).

Second order ODE is considered to linear homogeneous (Hsu and Kamran, 1989; Ibragimov and Magri, 2004; Kruglikov, 2008) if the right hand side equal zero and inhomogeneous if not.

In practical life we can deal with different types (Kruglikov and Lychagin, 2006b, 2008) of second order linear equations depending on the roots of the characteristic equation which represent the differential equation and these roots may be:

- Real, distinct roots
- Complex conjugate, distinct roots
- Repeated roots

Table 1 shows some examples of these types and these equations will be analyzed and implemented later in the experimental part of this study.

METHODS AND TOOLS

The methods which were used in this study are different MATLAB functions used to solve second order linear ODES and SIMULINK models.

The following code was implemented several times using various types of ODES and SIMULINK models:

```
Close all
Clear
tic, Solution_dsolve = dsolve ('3*D2y + 3*Dy + 9*y - 15*t', 'y (0) = 3', 'Dy (0) + 2 = 0'); toc
display ('Symbolic MATH dsolve function output is: ')
disp (Solution_dsolve)
```

Table 1: Examples of second order linear ODE

Equation No.	Roots	Example
1	Real, distinct roots homogeneous	$\ddot{x} + 5\dot{x} + 6x = 0$ with x (0) = 2, $\dot{x}$ (0) = 3
2	Real, distinct roots inhomogeneous	$\ddot{x} + 5\dot{x} + 6x = 15t$ with x (0) = 2, $\dot{x}$ (0) = 3
3	Real, distinct roots inhomogeneous	$\ddot{x} + 5\dot{x} + 6x = 3 \exp(4t)$ with x (0) = 2, $\dot{x}$ (0) = 3
4	Complex conjugate, distinct roots homogeneous	$\ddot{x} + x = 0$ with x (0) = 1 and $\dot{x}$ (0) = -2
5	Complex conjugate, distinct roots inhomogeneous	$\ddot{x} + x = 15t$ with x (0) = 1 and $\dot{x}$ (0) = -2
6	Complex conjugate, distinct roots inhomogeneous	$\ddot{x} + x = 3 \exp(4t)$ with x (0) = 1 and $\dot{x}$ (0) = -2
7	Repeated roots homogeneous	$\ddot{x} - 3\dot{x} - 4x = 0$ with x (0) = 1 and $\dot{x}$ (0) = 0
8	Repeated roots inhomogeneous	$\ddot{x} - 3\dot{x} - 4x = 2t$. x (0) = 1, $\dot{x}$ (0) = 0
9	Repeated roots inhomogeneous	$\ddot{x} - 3\dot{x} - 4x = 3e2t$ with x (0) = 1 and $\dot{x}$ (0) = 0

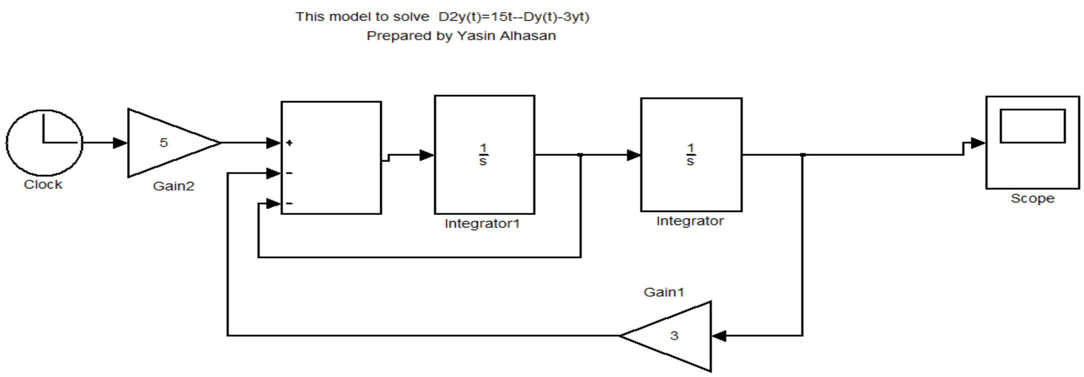

Fig. 1: SIMULINK model second order ODE

```
figure;
subplot (211)
ezplot (Solution_dsolve)
title ('Solution found with DSOLVE '); grid
syms t s Y
ODE2nd = '3*D (D (y)) (t) + 3*D (y) (t) + 9*y (t) - 15*t';
lt_A = laplace (ODE2nd, t, s);
lt_A = subs (lt_A, {'laplace (y(t), t, s)', 'y (0)', 'D (y) (0)'},{Y, 3, -2});
Y = solve (lt_A, Y);
display ('Laplace Transforms of the given 2nd Order ODE with ICs')
disp (Y)
tic, Solution_Laplace = ilaplace (Y); toc
display ('Solution found using Laplace Transforms')
disp (Solution_Laplace)
subplot (212)
ezplot (Solution_Laplace); grid
title ('Solution found with Laplace Transforms')
hold off
t = 0: pi/100: 6*pi;
SDsol = eval (vectorize (Solution_dsolve));
LTsol = eval (vectorize (Solution_Laplace));
figure;
plot (t, SDsol, 'bx-'); grid
hold on; xlabel ('t'); ylabel ('solution values')
plot (t, LTsol, 'ro-');
ICs = [3, -2];
time SPAN = 0: pi/100: 6*pi;
options_ODE = odeset ('RelTol', 1e-6, 'AbsTol', 1e-8);
tic, [t, y_ODE45] = ode 45 (@compare ODEsols, time SPAN, ICs, options_ODE); toc
plot (t, y_ODE45 (:,1), 'm+-');
tic, [t, y_ODE113] = ode113 (@compare OD Esols, time SPAN, ICs, options_ODE); toc
plot (t, y_ODE113(:,1), 'k<-');
title ('Comparison analysis of ODE sols with ODE45, ODE23, dsolve, Laplace Transforms and SIMULINK')
sim ('ODEsols_COMPARE.mdl');
tic, SIM_sols = ODEsols_SIM.signals. values; time = tout; toc
plot (time, SIM_sols, 'cd-');
legend ('dsolve', 'Laplace Transforms', 'ODE45', 'ODE113', 'SIMULINK', 0)
hold off
function dydt = compareODEsols (t, y)
% this is a 2nd order ODE IVP example
% 3*y"+ 3*y' + 9*y = 15*t with ICs: y (0) = 3, y' (0) = -2;
dydt = [y (2); - (y (2) + 3*y (1)) + 5*t ];
return
```

This program was used to solve the following ODE:

3*D2y + 3*Dy + 9*y = 15*t, 'y (0) = 3', 'Dy (0) = -2') (2)

The following SIMULINK model was built for this equation (Fig. 1) and implanted by the previous program.

All the methods used in the previous code show the same results as shown in Fig. 2, which means that the solutions were accurate and meat the real analytical and numerical solution.

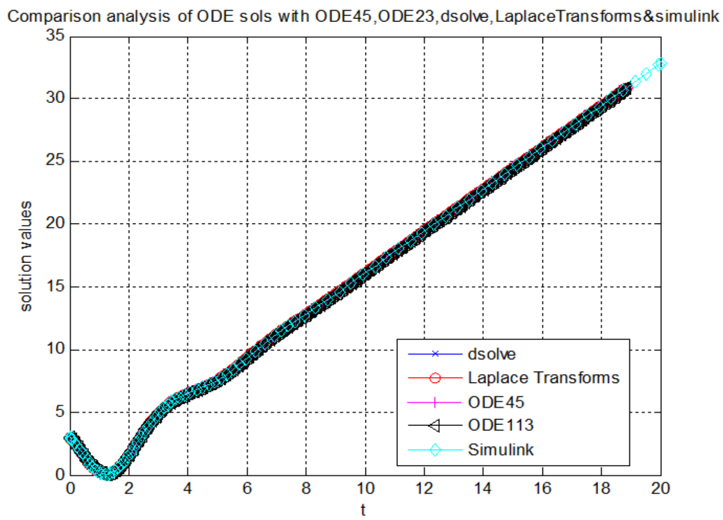

Fig. 2: Different methods implementation results

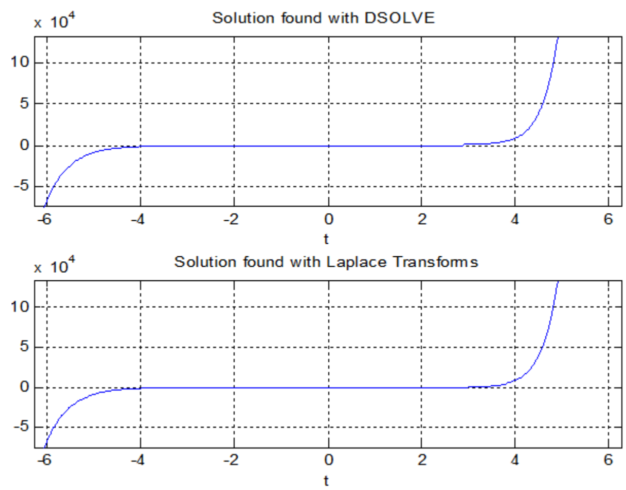

Fig. 3: Experiment 1 result matching

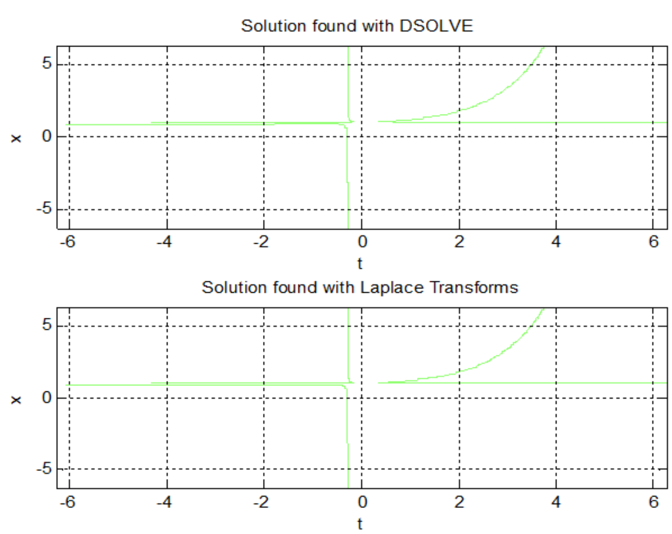

Fig. 4: Experiment 2 result matching

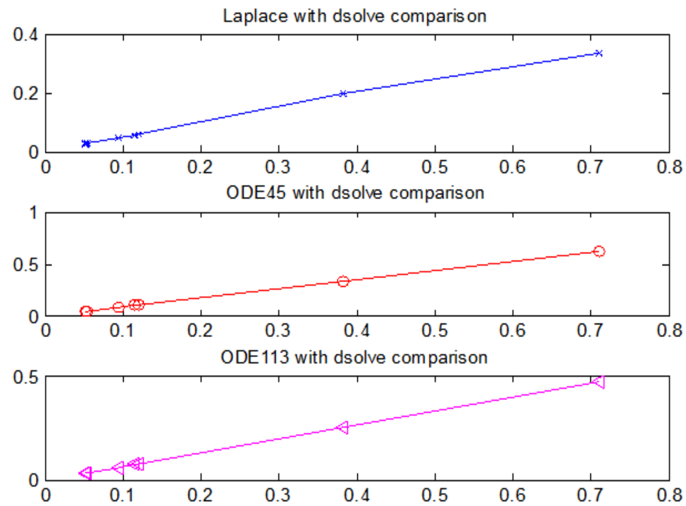

Fig. 5: Comparisons between dsolve and other method

Experimental part: MATLAB provides us with different methods of solving second order linear ODES such as: dsolve, Laplace transforms, ODE 45, ODE 113 and SIMULINK.

But which method to use better? All of them give accurate results as shown in Fig. 2, but which is the fastest method?

Table 2 show the time needed to solve Eq. (2) by each of the above mentioned methods.

The following experiments were performed:

Experiment 1: The following inhomogeneous ODE was solved:

$$y'' + 3y' + 2y = e^{3x} + \cos x, \qquad y(0) = 2, \qquad y'(0) = -$$

And below are the implementation results.

Figure 3 shows the matching between the 2 results.

Time for dsolve =
Elapsed time is 0.047000 sec
Symbolic MATH dsolve function output is:
1/10*cos (t) + 3/10*sin (t) + 1/20*exp (3*t) -2/5exp (-2*t) + 9/4*exp (-t)
Laplace Transforms of the given 2nd order ODE with ICs
(2*S^4 - 11*s^2 - s^3 - 4*s - 14) / (s-3) / (s^2 + 1) / (3*s + 2 + s^2)
Time for Laplace =
Elapsed time is 0.015000 sec
Solution found using Laplace transforms
1/10*cos (t) + 3/10*sin (t) + 1/20*exp (3*t) - 2/5*exp (-2*t) + 9/4*exp (-t)

Experiment 2: The following inhomogeneous ODE was solved:

Table 2: Time to solve Eq. (2)

Method	Execution time in sec
Dsolve	0.0610
Laplace	0.0305
ODE45	0.0530
ODE113	0.0400
SIMULINK	0.0001

$$y'' + 3y' + 2y = \frac{1}{1-x}, \qquad y(0) = 1, \qquad y'(0) = 2$$

And below are the implementation results.

Figure 4 shows the matching between the 2 results.

Time for dsolve = Elapsed time is 0.031000 sec

Time for laplace = Elapsed time is 0.016000 sec

Experiment 3: The previous code was implemented using various ODES such as mentioned in Table 1, 20 examples of each type of ODE were taken and implemented, 20 SIMULINK models were built for each type of ODE and Table 3 summarizes the results of this experiment focusing on the executions time.

RESULTS AND DISCUSSION

The above mentioned methods are very accurate in solving second order linear ODES as was shown in Fig. 2 to 4.

From the results in Table 3 we can say that the best method (with minimum implementation time) is SIMULINK modeling, but it needs more efforts in building the desired model (in experiment 3 it took 900 min to build and run 180 models).

Among the programming method the best one is Laplace transforms and the worst method is dsolve method.

Table 3: Summary results of experiment 3 (# of examples = 20)

ODE with	Avg. implementation time in sec				Avg. SIMULINK models execution time in sec
	Dsolve	Laplace	ODE45	ODE113	
Real, distinct roots homogeneous	0.053	0.0270	0.0461	0.0347	0.00019
Real, distinct roots inhomogeneous	0.095	0.0478	0.0827	0.0625	0.00015
Real, distinct roots inhomogeneous	0.710	0.3370	0.6170	0.4703	0.02304
Complex conjugate, distinct roots homogeneous	0.053	0.0260	0.0462	0.0348	0.00013
Complex conjugate, distinct roots inhomogeneous	0.054	0.0270	0.0464	0.0347	0.00016
Complex conjugate, distinct roots inhomogeneous	0.055	0.0270	0.0465	0.0348	0.00016
Repeated roots homogeneous	0.116	0.0560	0.1008	0.0762	0.00362
Repeated roots inhomogeneous	0.120	0.0620	0.1043	0.0790	0.00428
Repeated roots inhomogeneous	0.382	0.2000	0.3321	0.2540	0.01206

Avg.: Average

Figure 5 shows the comparisons between dsolve method and other methods.

CONCLUSION

From the results obtained previously we can conclude the following:

- The best performance can be achieved using SIMULINK models, but extra efforts are needed to build the model.
- Among the programming methods the best performance can be achieved using Laplace transforms and the worst performance can be achieved by dsolve method.
- Comparing with dsolve method Laplace method has a speedup of 2 times, ODE 113 has a speedup of 1.525 times and ODE 45 has a speedup of 1.15 times.

REFERENCES

Bryant, R., M. Dunajski and M. Eastwood, 2008a. Metris Ability of Two-Dimensional Pro-Ejective Structures. arXiv: math/0801.0300 (2008).

Bryant, R., G. Manno and V.S. Matveev, 2008b. A solution of a problem of Sophism Lie: Normal forms of two-dimensional metrics admitting two projective vectored. Math Ann., 340(2): 437-463.

Fels, M. and P. Olver, 1997. On relative invariants. Math Ann., 308(4): 701-732.

Hsu, L. and N. Kamran, 1989. Classification of second-order ordinary differential equations admitting Lie groups of-bre-preserving point symmetries. Proc. London. Math Soc., 58(2): 387-416.

Ibragimov, N. and F. Magri, 2004. Geometric proof of lie's linearization theorem. Non-linear Dyna., 36: 41-46.

Kruglikov, B., 2008. Invariant characterization of Liouville metrics and polynomial integrals. J. Geomet. Phys., 58(8): 979-995.

Kruglikov, B.S. and V.V. Lychagin, 2008. Geometry of Diferential Equations. In: D. Krupka and D. Saunders, (Eds.), Handbook of Global Analysis, Elsevier, pp: 725-772.

Kruglikov, B.S. and V.V. Lychagin, 2006a. Invariants of pseudo group actions: Homolog-ical methods and Finiteness theorem. Int. J. Geomet. Meth Mod. Phys., 3(5-6): 1131-1165.

Kruglikov, B.S. and V.V. Lychagin, 2006b. Compatibility, multi-brackets and inte-grability of systems of PDEs. Report number: Preprint of Tromso University no. 2006-49 ArXive: math.DG/0610930.

Nurowski, P. and G.A. Sparling, 2003. Three-dimensional Cauchy-Riemann structures and second-order ordinary di®erential equations. Classical Quantum Gravit., 20(23): 4995-5016.

Yumaguzhin, V., 2008. Di®erential invariants of 2{order ODEs, I, arXiv: math.DG/0804.0674 v1.