Advantages and Disadvantages of Using MATLAB/ode45 for Solving Differential Equations in Engineering Applications

Waleed K. Ahmed *w.ahmed@uaeu.ac.ae*
Faculty of Engineering/ Engineering Requirements Unit
UAE University
Abu Dhabi, Al-Ain, 15551, UAE

Abstract

The present paper demonstrates the route used for solving differential equations for the engineering applications at UAEU. Usually students at the Engineering Requirements Unit (ERU) stage of the Faculty of Engineering at the UAEU must enroll in a course of Differential Equations and Engineering Applications (MATH 2210) as a prerequisite for the subsequent stages of their study. Mainly, one of the objectives of this course is that the students practice MATLAB software package during the course. The aim of using this software is to solve differential equations individually and as a system of equation in parallel with analytical mathematics trends. In general, mathematical models of the engineering systems like mechanical, thermal, electrical, chemical and civil are modeled and solve to predict the behavior of the system under different conditions. The paper presents the technique which is used to solve DE using MATLAB. The main code that utilized and presented is MATLAB/ode45 to enable the students solving initial value DE and experience the response of the engineering systems for different applied conditions. Moreover, both advantages and disadvantages are presented especially the student mostly face in solving system of DE using ode45 code

Keywords: Advantages, DE, Disadvantages, Engineering, MATLAB, ode45.

1. INTRODUCTION

Engineering software is one of the most important tools in engineering education and they can be very useful in teaching the working principles of various engineering instruments and devices [1]. Usually differential equations can be used for solving different applications of engineering fields like electrical, mechanical, civil and chemical besides to the petroleum engineering. Normally, at the early stages of their study in the faculty of engineering at UAEU, a considerable amount of time and efforts are needed by the students to solve the differential equations of the mathematical models of different engineering systems especially the dynamic systems [2]. In general, students experience applied differential equations through the engineering applications whereas involved with the classical math solution through the mathematic lecture. One of the objectives of the course MATH 2210 is to let students practice solving applied differential equations of the engineering cases [3]. Generally, the mathematical solution of these equations does not readily which provides the student with a graphical image of the anticipated results, since they deal with different style of the mathematical models depending on the case studied itself. As a result, the students are uncomfortable with the entire process of solving these types of systems [4].

As a matter of fact, writing a program from first principles is not an easy task and needs considerable programming skills [1]. Mainly programs are developed using a high-level programming language such as FORTRAN, C++, PASCAL or BASIC to solve the differential equations. The advantages and disadvantages of writing a simulation program can be summarized as following:

• Requires excellent programming skills
• Takes long time to develop and test
• Very cheap as the existing compilers can be used for development.
• Source code is available, so it can be modified and upgraded at no cost.
• The program is usually written for a dedicated simulation problem.

An alternate option of solving engineering system is to use a commercially available simulation package. Using this option can be classified into two categories: general simulation packages, and dedicated simulation packages. General simulation packages can be used to simulate and solve most of the engineering of different fields, such as Excel [5], MATLAB [6], MatriXx [7], and so on. The advantages and disadvantages of general simulation packages are:

• Affordable cost.
•No Programming skills are required.
• Relatively easy to simulate and generate results.
• It is necessary to get the mathematical model of simulated system and this may be difficult to handle.
• The source code is not available and hence it is not possible to modify/upgrade the simulation package.

MATLAB is a computer program that provides the user with a convenient environment for performing many types of calculations [8]. Besides it is usually used to solve differential equations and it is an effective way and can be considered as quick and easy. Moreover, it may also provide the student with the symbolic solution and a visual plot of the result. One of the most popular codes used to solve differential equation is ode45, which is mainly used for solving engineering applications of the MATH2210. Dealing with diverse engineering applications can result different order of differential equations, besides the engineering system it may result a system of differential equation whether the system has the same style of DE or mixed order DE. This can cause a sort of complication toward writing the MATLAB program which is dedicate to predict the behavior of the engineering case studied.

In this paper, a different system of differentials equations which express the mathematical model of diverse applications [9] is inspected to test the differences in the solution by MATLAB using the ODE45 code technique instead of using numerical analysis technique. This will clarify the advantages and disadvantages of the code used. Mainly three cases are demonstrated, the first order differential equation which represented by the thermal system and the second order differential equation which is practiced through mechanical system and eventually the mixed order of differential equations which is mostly encountered by the electrical applications.

2. FIRST ORDER DIFFERENTIAL EQUATIONS OF THERMAL SYSTEM

One of the first engineering applications that always students practice in the MATH 2210 course is the thermal system [10]. The aim of this application is to start with a single first order differential equation and as well as a system of equations. Unusually this system generates a mathematical model in term of first order and this is because the rate of change of temperature. As an example Consider the closed vessel of thermal resistance R_{La} which is filled with liquid that contains an electrical heater immersed in the liquid. The heater is contained within a metal jacket of thermal resistance R_{HL}. The heater has a thermal capacitance of C_H, and the liquid has a thermal capacitance of C_L. The heater temperature is Θ_H, and the liquid temperature is Θ_L. The heater is supplying energy at a rate of qi(t), and the exterior temperature is Θ_a. The objective is to find the system mathematical model in terms of $\Theta_H(t)$, $\Theta_L(t)$ which represents the temperature variation with respect to time. The present case is modeled by a couple of DE derived from the heat transfer principles [10]. The mathematical model is shown below:

$$\dot{\theta}_H(t) = \frac{1}{C_H}\left[q_i(t) - \frac{1}{R_{HL}}\left(\dot{\theta}_L(t) - \theta_L(t)\right)\right] \tag{1}$$

$$\dot{\theta}_L(t) = \frac{1}{C_L}\left[\frac{1}{R_{HL}}\left(\theta_H(t) - \theta_L(t)\right) - \frac{1}{R_{La}}\left(\theta_L(t) - \theta_a\right)\right] \tag{2}$$

This can be written in matrix form:

$$\begin{bmatrix} \dot{\theta}_H(t) \\ \\ \dot{\theta}_L(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{C_H R_{HL}} & \frac{-1}{C_H R_{HL}} \\ \frac{1}{C_L R_{HL}} & \frac{-1}{C_L R_{HL}} - \frac{1}{C_L R_{La}} \end{bmatrix} \begin{bmatrix} \theta_H(t) \\ \\ \theta_L(t) \end{bmatrix} + \begin{bmatrix} q_i(t) \\ \\ \frac{\theta_a}{R_{La}} \end{bmatrix} \tag{3}$$

In compact form this is represented by [3]:

[illegible] = [illegible] + [illegible]([illegible]) (4)

$$x^{\bullet} = \begin{bmatrix} \dot{\theta}_H(t) \\ \\ \dot{\theta}_L(t) \end{bmatrix}, \mathrm{A} = \begin{bmatrix} \frac{1}{C_H R_{HL}} & \frac{-1}{C_H R_{HL}} \\ \frac{1}{C_L R_{HL}} & \frac{-1}{C_L R_{HL}} - \frac{1}{C_L R_{La}} \end{bmatrix} \tag{5}$$

$$\mathrm{x} = \begin{bmatrix} \theta_H(t) \\ \\ \theta_L(t) \end{bmatrix}, B = \begin{bmatrix} q_i(t) \\ \\ \frac{\theta_a}{R_{La}} \end{bmatrix} \tag{6}$$

One of the features that ode45 solver requires is that the system of equations must be in first order differential equations [8], and this is already generated by the thermal system mathematical model. Therefore it will be straight forward to program the code necessary for the solution as shown below without any complication, only needed is to form the mathematical model to be accepted by the code. The program for the mathematical model is shown below.

```
Clear;clc
[t,theta] = ode45('ThermalEx3',[0 1000],[300 300]);
plot(t,theta(:,1),t,theta(:,2))

function dthetadt = ThermalEx3(t,theta)
RHL=1e-3; CH=20e3;
RLa=5e-3; CL=1e6;
thetaa=300;qi=2500;
eq1=[-(1/(RHL*CH))*theta(1)+(1/(RHL*CH))*theta(2)+(1/CH)*qi];
eq2=[-((1/(RHL*CL))+(1/(RLa*CL)))*theta(2)+(1/(RHL*CL))*theta(1)+(1/(RLa*CL))*thetaa];
dthetadt =[eq1;eq2];
```

3. SECOND ORDER DIFFERENTIAL EQUATIONS OF MECHANICAL SYSTEM

Usually the mathematical model of mechanical engineering systems is in the form of second order differential equation, and that because of the Newton's second law. In this example, a mass-spring-damper is modeled and then its behavior is simulated [11]. The system consists of a mass (m), a spring (k), and a damper (C). An external force (f) is applied on the mass. The system can easily be simulated and its response can be plotted as a function of time. Before the system can be simulated it is necessary to derive its mathematical model. The mathematical model of mechanical systems usually has the following form:

$$Mx''+Cx'+kx=f(t) \tag{7}$$

For the above system, simple there are two approaches can be easily used to solve the mathematical model using MATLAB. In this paper both ODE45 code as the technique that the students learn through the lectures. Unfortunately using ODE45 is not straight forward way to solve a second order differential equation, so a modification must be done by the students to be suitable for the code utilization and solution. By reorganizing a single second order differential to a couple of first order differential equation by defining a new system of variables [12], the new system will be quite fit for the ode45 code for the solution. In particular, if a new variables z_1 and z_2 are introduced such that:

$$z_1=x \quad \text{and} \quad z_2=x' \tag{8}$$

These implies that

$$z'_1=z_2\,,\ \ z'_2= x'' \tag{9}$$

Then the model can be written as:

$$m.z_2'+C.z_2+k.z_1=f(t) \tag{10}$$

Next solve for z'_2 to get:

$$z_2'=1/m\ \ (f(t)-C.z_2-k.z_1) \tag{11}$$

Equations (3) and (4) constitute a state variable model corresponding to the reduced model. The variables z_1 and z_2 are the state variables. The choice of the state variable is not unique, but the choice must result in a set of first order differential equations, and a consequence the situation will be more complicated when dealing with more than single equation, and this is one of the most problems that the students face in this stage. The state variable equation can be written in matrix form as follows:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} f(t) \tag{12}$$

In compact form this can be represented by:

$$\mathrm{A} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \quad B= \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \quad z= \begin{bmatrix} \dot{z}_1 \\ z_2 \end{bmatrix} \tag{13}$$

The mechanical system can now be solved the using the MATLAB package. The MATLAB function developed to simulate the system and plot the displacement and velocity of the system. The component values can easily be changed and the system can be re-simulated. In this example, the following component values were chosen for the simulation: m=1 kg, C=2 N.s/m, k=1000 N/m, initial displacement (Xo)=0.5 m, initial velocity(Vo)=0.2 m/s, force(f)=50*sin(2t).

```
function dzdt = Mech(t,z)
m=1;
k=100;
C=2;
dzdt = [ z(2); (1/m)*(50*sin(2*t)-C*z(2)-k*z(1)) ];

[t, z] = ode45('Mech',[0,20],[0.5; 0.2]);
plot(t, z)
```

4. MIXED ORDER DIFFERENTIAL EQUATIONS

The code for a first-order ODE is very straightforward. However, a second or third order ODE cannot be directly used. You must first rewrite the higher order ODE as a system of first-order ODEs that can be solved with the MATLAB ODE solvers [13].

This is an example of how to reduce a second-order differential equation into two first order equations for use with MATLAB ODE solvers such as ODE45. The following system of equations consists of one first and one second-order differential equations:

$$x' = -y * \exp(-t/5) + y' * \exp(-t/5) + 1 \tag{14}$$
$$y'' = -2*\sin(t) \tag{15}$$

By assuming z_1=x, the first step is to introduce a new variable that equals the first derivative of the free variable in the second order equation z_2=y and z_3=y'. Taking the derivative of each side yields the following:

$$z_2'=y'=z3 \tag{16}$$
$$z_3' = y'' \tag{17}$$

Substituting (17) into (15) produces the following:

$$z_3' = -2*\sin(t) \tag{18}$$

Combining (14), (16), and (18) yields three first order differential equations.

$$z_1' = -y * \exp(-t/5) + y' * \exp(-t/5) + 1; \tag{19}$$
$$z_2' = y' \tag{20}$$
$$z_3' = -2*\sin(t) \tag{21}$$

Since z_3 = y', substitute z_3 for y' in equation (19). Also, since MATLAB requires that all derivatives are on the left hand side, rewrite equation (20). This produces the following set of equations:

$$z_1' = -z_2 * \exp(-t/5) + z3 * \exp(-t/5) + 1 \tag{22}$$
$$z_2' = z_3 \tag{23}$$
$$z_3' = -2*\sin(t) \tag{24}$$

The matrix form of the final system of equations is shown by:

$$\begin{bmatrix} z1' \\ z2' \\ z3' \end{bmatrix} = \begin{bmatrix} 0 & -\exp(-t/5) & \exp(-t/5) \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z1 \\ z2 \\ z3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ -2 * \sin(t) \end{bmatrix} \tag{23}$$

To evaluate this system of equations using ODE45 or another MATLAB ODE solver, create a function that contains these differential equations. The function requires two inputs, the states and time, and returns the state derivatives.

Following is the MATLAB code needed for the solution to evaluate the system of equations using ODE45 or another MATLAB ODE solver. Define the start and stop times and the initial conditions of the state vector:

```
function xprime = odetest(t,z)
eq1 = -z(2) * exp(-t/5) + z(3) * exp(-t/5) + 1;
eq2 = z(3);
eq3 = -2*sin(t);
xprime = [eq1;eq2;eq3];

clc
clear
t0 = 5;
tf = 20; x0 = [1 -1 3] % Initial conditions
[t,z] = ode45('odetest',[t0,tf],x0);
plot(t,z)
```

Unfortunately this case mostly can be faced in the electrical applications and this cause complication especially if the mathematical model contain a system of equations and hence the probability of the mistakes through the transformations stages is very high and will be reflected on the final results and not forgetting the time consumed. Instead SIMULINK [14] can be used as an alternate option for the engineering applications.

5. CONCLUSIONS

The present paper shows that dealing with differential equations of the mathematical models of engineering systems mostly encounter difficulties in term of the MATLAB programming. As a matter of fact this complication is due to restrictions of the ode45 which is mostly used to solve such kind of differential equations. Few systems like thermal system are in the form of first order differential equations which are quite fit with requirements of the ode45. The major restriction of the MATLAB solve code is that the system of differential equations should be organized in the form of the first order differential equations, and this frequently is a rare case, whereas the core engineering application either in the form of second order of even mixed order. Therefore, transformation of the system of differential equations is mandatory, and this can make mistakes beside to the time spent.

6. REFERENCES

[1] D. Ibrahim. “Engineering simulation with MATLAB: improving teaching and learning effectiveness”. Procedia Computer Science 3, 2011, pp. 853–858.

[2] G.F. Simmons and S.G. Krantz. “Differential Equations: Theory, Techniques, and Practice”, International Edition, New York: McGraw Hill, 2007.

[3] P.V. O'Neil. “Advanced Engineering Mathematics”, International Student Edition, Ontario: Thomson, 1995.

[4] F. Pietryga. “Solving Differential Equations Using MATLAB/Simulink”, Proceedings of the American Society for Engineering Education Annual Conference & Exposition, 2005.

[5] J. Walkenbach. “Excel 2007 Bible”, Indianapolis: John Wiley & Sons, 2007.

[6] B. Hahn, D. Valentine. "Essential Matlab for Engineers and Scientists", 4th edition, MA: Academic Press, 2007.

[7] MatriXx Software Suite, National Semiconductors, web site: http://www.ni.com .

[8] S.C. Chapra. "Applied Numerical Methods with MATLAB", 2nd ed., New York: McGraw Hill, ,2008.

[9] D. Zill. "A First Course in Differential Equations with Modeling and Applications", Brooks/Cole Publishing Company, 6th ed, 1997.

[10] C. Close, D. Frederick, J. Newell. "Modeling and Analysis of Dynamic Systems", John Wiley and Sons, Inc., 3rd ed, 2002.

[11] K. Ogata. "System Dynamics", 4th ed., New Jersy: Pearson Prentice Hall, 2004.

[12] W.J. Palm. "System Dynamics", International 2nd Edition, New York: McGraw Hill, 2010.

[13] http://www.mathworks.com/support/tech-notes/1500/1510.html.

[14] W.Ahmed, K.Harib, "MATLAB/SIMULINK to Solve Mathematical Models of Engineering Systems: Class Activity", ICERI2011, Proceedings of the 4th International Conference of Education, Research and Innovation, Madrid, Spain, 14-16 November, 2011. ISBN: 978-84-615-3324-4.