Session Number _____

Solving Differential Equations Using MATLAB/Simulink

Frank W. Pietryga, P.E.

University of Pittsburgh at Johnstown

Abstract

During the sophomore year, the mechanical and civil engineering technology students are required to complete a course in computer systems, programming and applications. The selected software package for this course was MATLAB designed and developed by the Mathworks [1]. Each student was provided with a student version of the programming software. After the introductory material had been covered, the latter part of the course was used to solve basic differential equations using the MATLAB symbolic equation solver and the built-in plotting capabilities [2]. Also, the students were provided instruction on Simulink. Simulink is a graphical programming language that uses MATLAB as the computational engine. Finally, methods are presented that allow the student to use both programming tools in conjunction with one another. The students were given the opportunity to contrast the techniques of solving differential equations using classical mathematical methods and the computer.

The students experienced actual hands on programming time with the instructor during the laboratory period of this course. The students seemed to really enjoy these methods and were much more comfortable with the graphical results.

Introduction

Many applications in electrical, mechanical and civil engineering technology require solving differential equations. Often, the solutions of such equations require a substantial amount of time and effort by the student. The mathematical solution of these equations does not readily provide the student with a "graphical picture" of the result. Therefore, students are uncomfortable with the entire process of solving these types of systems. Using MATLAB/Simulink to solve differential equations is very quick and easy. It may also provide the student with the symbolic solution and a visual plot of the result.

This paper will examine 3 simple applications in electrical, mechanical, and civil engineering technology requiring the solution of a differential equation. First, the author will present a method using the symbolic processing capabilities of MATLAB to quickly code a differential equation for a graphical solution. Second, the differential equations will be modeled and solved graphically using Simulink. Finally, the author will present methods which use both MATLAB and Simulink together. The MATLAB script files being used to call a Simulink model of a

Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition

system. The Simulink program sending the simulation results back to MATLAB for plotting purposes.

Electrical Engineering Technology Application

The first order ordinary differential equation that describes a simple series electrical circuit with a resistor, inductor, and sinusoidal voltage source is as follows:

$$L \times \frac{di}{dt} = 10 \times \sin(150 \times t) - i \times R$$

For this example, the inductance (L) is 1 henry and the resistance (R) is 10 Ω. The voltage source is sinusoidal with a peak voltage equal to 10 volts and an angular velocity of 150 radians/second. The MATLAB statement written to solve the above ODE is as follows:

```
solution_1 = dsolve('DI=10*sin(150*t)-10*I','I(0)=0','t');
```

The dsolve function symbolically solves the ordinary differential equation specified above. The first argument is the actual ODE to be solved. D denotes differentiation with respect to the independent variable which in this case is time or t. A repeated differentiation process would be required if a digit other than 1 were to follow D. The character following the D (I in this case, which is the circuit current), is the dependent variable. The remaining terms of the equation are coded as usual. The second argument is used to specify the initial conditions of the circuit. The initial current in this circuit at t = 0 is 0. The final argument is used to specify the independent variable. Note that all arguments must be contained within single quotation marks as shown.

The output from the dsolve function is the symbolic solution to the differential equation. To get a quick visual understanding of the solution, the student should use the MATLAB ezplot function in conjunction with the xlabel, ylabel, and grid commands. The result is a simple graph which displays the numerical solution to the differential equation, the symbolic solution to the differential equation, and appropriate units for the x and y axes. The MATLAB statements written to provide the output to the graphical environment are as follows:

```
ezplot(solution_1,[0 .5]),…

xlabel('Time,Seconds'),ylabel('Current,Amperes'),…

grid
```

The electrical circuit defined above may be solved quickly using the symbolic solving capabilities of MATLAB. Only 1 statement was needed to define the engineering application. For plotting purposes and proper documentation of the solution, 3 more statements are required. This short program was run on the Student Version of the MATLAB software distributed to the class. The results of the program are presented in Figure 1.

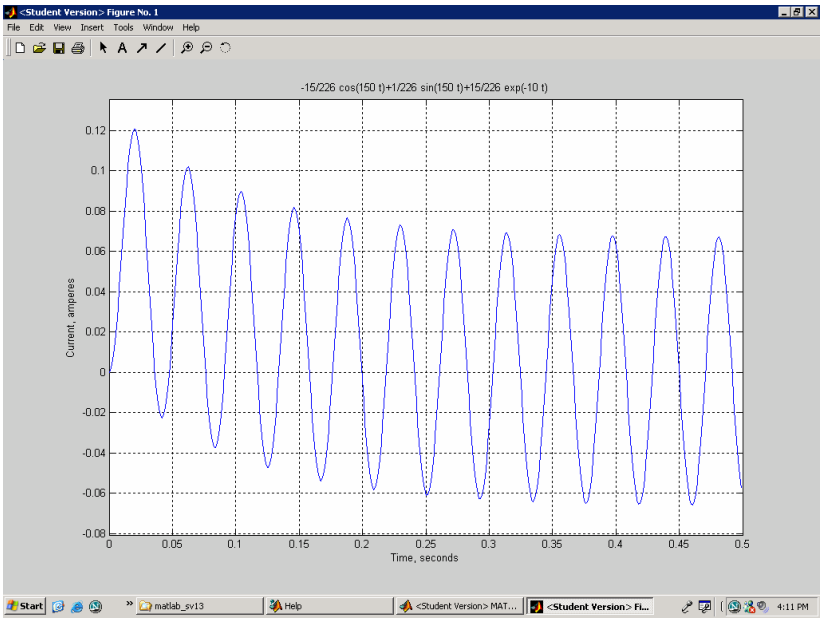

FIGURE 1 Solution to ODE describing an RL Electrical Circuit

Next, the differential equation describing the RL electrical circuit is modeled using Simulink. Simulink is a block diagram programming language that is packaged with the Student Version of MATLAB. With Simulink, the differential equation is described using blocks from the Simulink library. The blocks include an integrator, gain, summer, sine wave source, and scope. The blocks are then "wired" together to generate the differential equation. The complete Simulink model for the electrical circuit is depicted in Figure 2

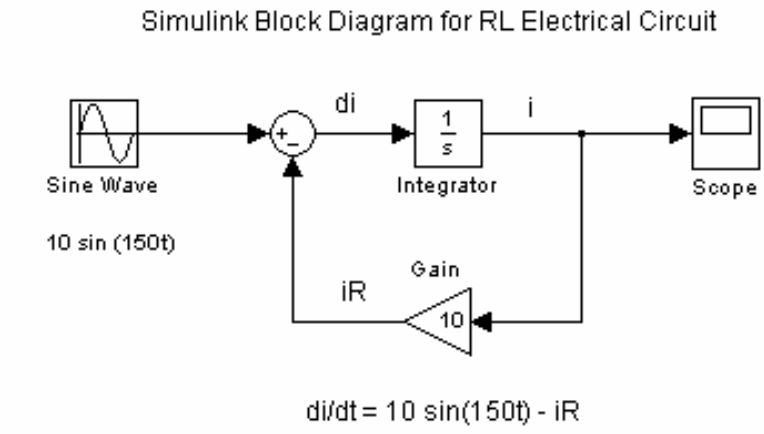

FIGURE 2 Simulink model for an RL Electrical Circuit

The scope block in Simulink allows the student to generate a quick "picture" of the current flowing in the electrical circuit. For this example, the x and y axes were scaled to resemble the MATLAB graph produced for Figure 1. The results of the solution to the equation using Simulink are shown in Figure 3.

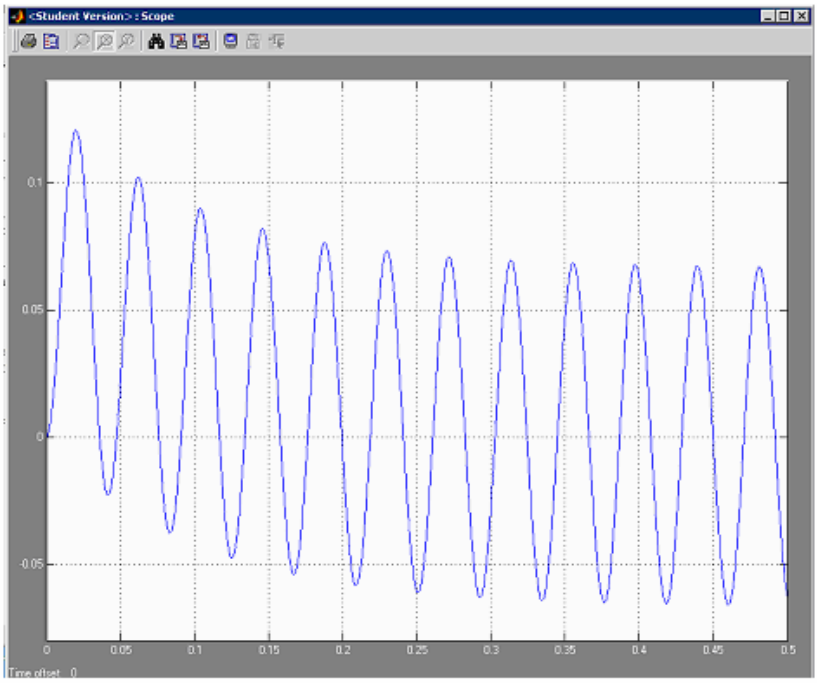

FIGURE 3 Simulink Results for an RL Electrical Circuit

Next, the Simulink model is modified to provide simulation data (time, current) to the MATLAB workspace using the To Workspace block. All Simulink variables are now defined in the MATLAB script file and the Simulink model remains unedited. The script file calls the Simulink model and the resulting simulation data is presented using the MATLAB plot command. The script file allows the student to quickly change the parameters of the model to simulate multiple cases. The modified Simulink model is shown in Figure 4.

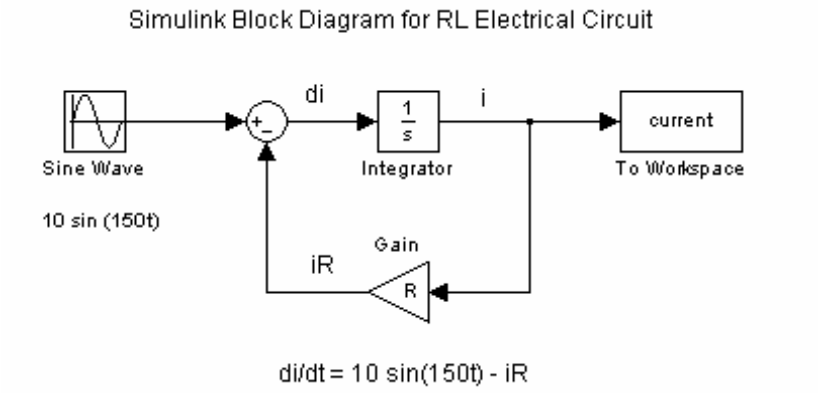

FIGURE 4 Simulink model to send results to MATLAB Workspace

The actual MATLAB m file used for this example is shown in Figure 5 and the results of the simulation are shown in Figure 6.

```
%  Differential Equation Model for RL Electrical Circuit
%
%  Define Constants for Sinusoidal Source
%
Amplitude = 10;
Frequency = 150;
%
%  Define Ohmic Value of Resistor
%
R = 10;
%
%  Call Simulink Model eet_matlab
%  Uses Parameters Defined Above
%  Output tout, current To Workspace
%
sim('eet_matlab')
%
%  Setup Plot
%
plot(tout,current),...
    title('Circuit Current vs. Time'),...
    ylabel('Current, Amperes'),...
    xlabel('Time, Seconds'),...
    grid
```

FIGURE 5 MATLAB script file used to call Simulink model and plot results

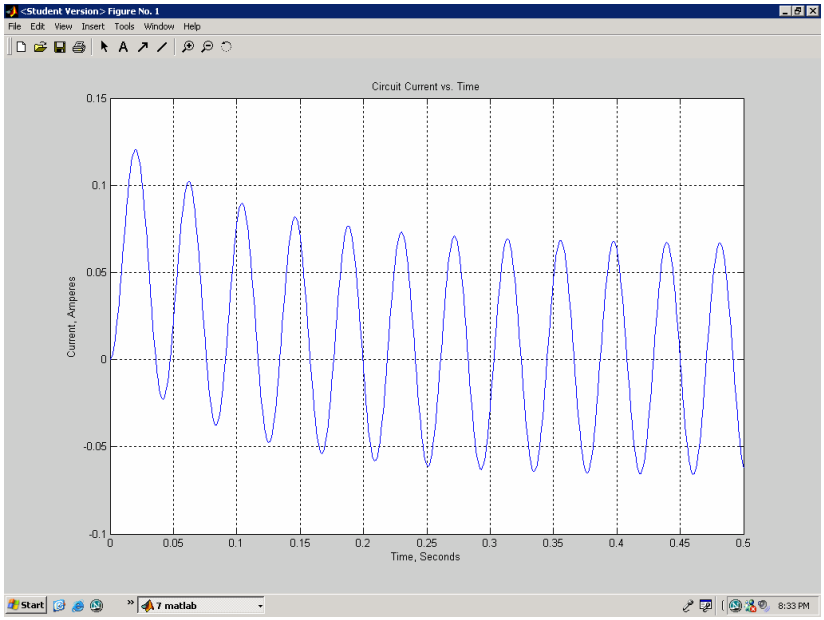

FIGURE 6 Results from MATLAB script file

Mechanical Engineering Technology Application

An object moving though or across a resisting medium experiences a retarding force proportional to the velocity. Applying Newton's Law of Motion produces the following first order differential equation:

$$m \times \frac{dv}{dt} = F - k \times v$$

For this example, the mass (m) is 1 kg., the acting force (F) is 49 N, and the viscous frictional force is equal to (k) .2 times the velocity (v). The MATLAB statement written to solve the above ODE is as follows:

```
solution_2 = dsolve('DV=49-.2*V','V(0)= 0','t');
```

The MATLAB statements written to provide the output to the graphical environment are as follows:

```
ezplot (solution_2,[0 50]),…

xlabel('Time,Seconds'),ylabel('Velocity,Meters/Second'),…

grid
```

The results of the simulation are shown in Figure 7.

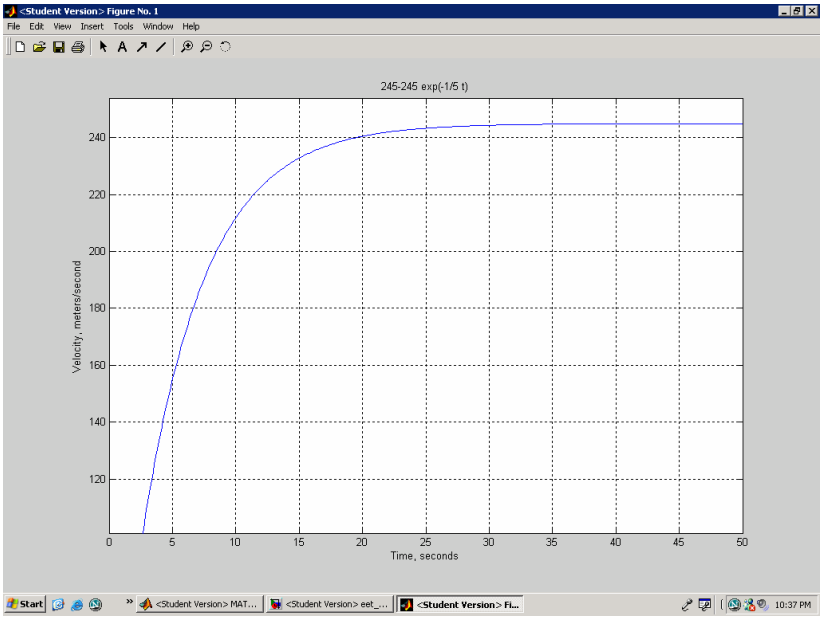

FIGURE 7 Solution to ODE describing a Mechanical system

Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition

The Simulink model for this application is shown in Figure 8.

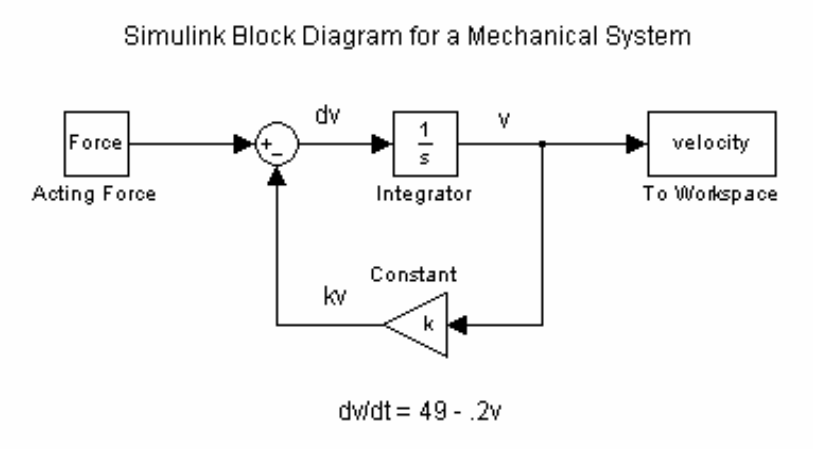

FIGURE 8 Simulink model for a Mechanical system

The actual MATLAB m file used for this example is shown in Figure 9 and the results of the simulation are shown in Figure 10.

```
%  Differential Equation Model for Mechanical System
%
%  Define Constants for Mechanical System
%
Force = 49;
k = .2;
%
%  Call Simulink Model met_matlab
%  Uses Parameters Defined Above
%  Output tout, velocity To Workspace
%
sim('met_matlab')
%
%  Setup Plot
%
plot(tout,velocity),...
    title ('Velocity vs. Time'),...
    ylabel('Velocity, Meters/Second'),...
    xlabel('Time, Seconds'),...
    grid
```

FIGURE 9 MATLAB script file used to call Simulink model

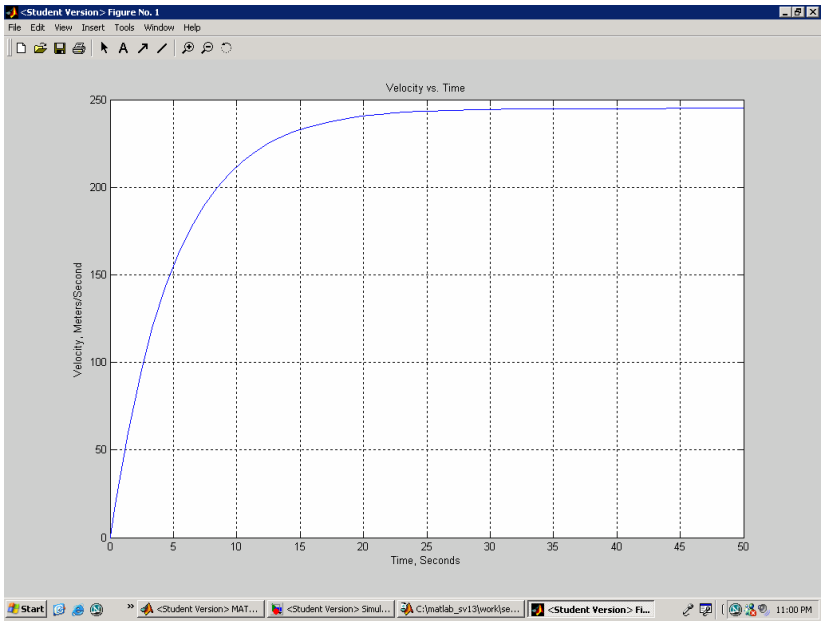

FIGURE 10 Results from MATLAB script file

Civil Engineering Technology Application

The methods presented for solving the previous engineering problems are provided in abbreviated format for a Civil Engineering Technology application. One application encountered in this discipline is the solution of a tank mixture problem. For example, 40 gallons of a solution originally containing 20 pounds of salt are in a tank. Fresh water is flowing into the tank at a rate of 2 gallons/minute. The same amount of mixture is running out of the tank each minute. The first order differential equation that describes this application is as follows:

$$\frac{ds}{dt} = -\frac{1}{20} \times s$$

For this example, (s) is the number of pounds of salt in the tank at time t. The initial condition for this problem is at t = 0 minutes, there is 20 pounds of salt in the tank. The MATLAB statement written to solve the above ODE is as follows:

```
solution_3 = dsolve('DS=-(1/20)* S','S(0)= 20','t');
```

The MATLAB statements written to provide the output to the graphical environment are as follows:

```
ezplot(solution_1,[0 100]),…

xlabel('Time,Minutes'),ylabel('Salt in Tank,Pounds'),…

grid
```

The results of the simulation are shown in Figure 11.

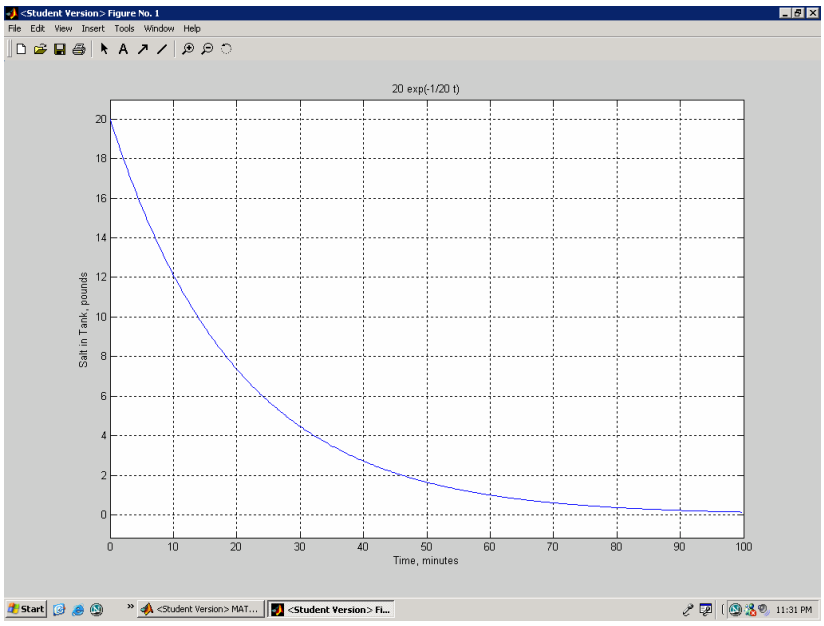

FIGURE 11 Solution to ODE describing a tank mixture problem

The Simulink model for this application is shown in Figure 12.

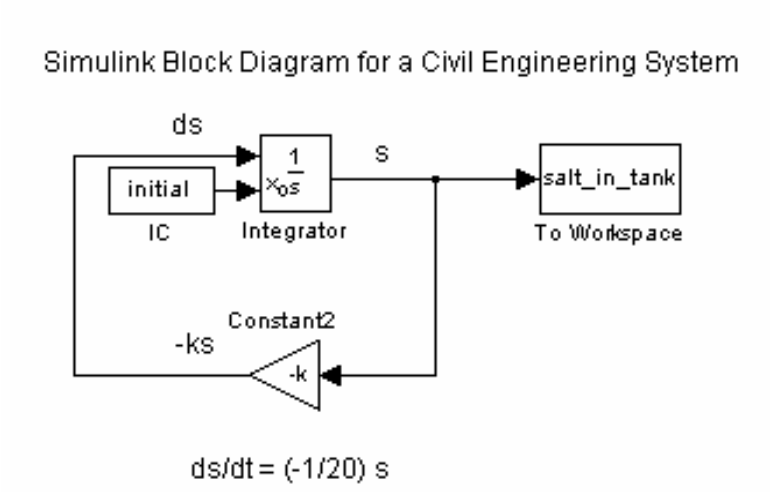

FIGURE 12 Simulink model for a tank mixture problem

The actual MATLAB m file used for this example is shown in Figure 13 and the results of the simulation are shown in Figure 14.

```
%  Differential Equation Model for Civil Engineering Tank Problem
%
%  Define Constants for System
%
k = .05;
initial = 20;
%
%  Call Simulink Model cet_matlab
%  Uses Parameters Defined Above
%  Output tout, salt_in_tank To Workspace
%
sim('cet_matlab')
%
%  Setup Plot
%
plot(tout,salt_in_tank),...
    title ('Salt In Tank vs. Time'),...
    ylabel('Salt In Tank, Pounds'),...
    xlabel('Time, Minutes'),...
    grid
```

FIGURE 13 MATLAB script file used to call Simulink model

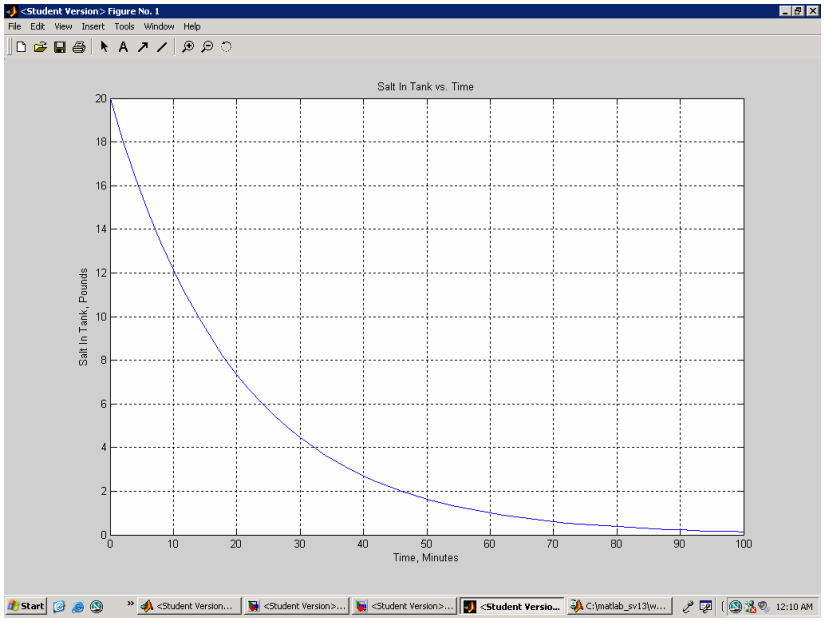

FIGURE 14 Results from MATLAB script file

Conclusions

In this paper, the author has presented a variety of methods using MATLAB/Simulink to solve first order differential equations. An example problem for each of three engineering disciplines has been provided. There was excellent correlation in the results for the three techniques described: using the MATLAB symbolic solver (dsolve), using the block diagram programming

language, Simulink, and using a script file to integrate features from both MATLAB and Simulink. The techniques presented could easily be expanded to provide solutions for higher order systems. With a minimal amount of effort, these techniques allow students to solve complicated engineering applications with a "graphical picture" of the result.

Although no formal data was collected regarding the students' responses to the methods presented, the course evaluations were very favorable. The students were very enthusiastic about applying these solution methods in their other engineering courses.

Bibliography

1. The Mathworks, Inc., Natick, MA. MATLAB/Simulink Student Version 13.0 http://www.mathworks.com.

2. "Engineering Problem Solving with MATLAB", Second Edition, by Delores M. Etter.

Biography

FRANK W. PIETRYGA is an Assistant Professor at the University of Pittsburgh at Johnstown. He graduated from UPJ in 1983 with a BSEET degree and completed his MSEE degree in 1993 at the University of Pittsburgh, main campus. His interests include power system engineering, AC/DC machinery, power electronics, and motor drive systems. Mr. Pietryga is also a registered professional engineer in the Commonwealth of Pennsylvania