PROJECTS WITH APPLICATIONS OF DIFFERENTIAL EQUATIONS AND MATLAB

David Szurley
Francis Marion University
Department of Mathematics
PO Box 100547
Florence, SC 29502
dszurley@fmarion.edu

I. INTRODUCTION

Differential equations (DEs) play a prominent role in today's industrial setting. Many physical laws describe the rate of change of a quantity with respect to other quantities. Since "rate of change" is simply another phrase for derivative, these physical laws may be written as DEs. For example, Newton's Second Law of Motion states that the rate of change of momentum of an object is equal to the sum of the imposed forces. Normally this is written as

$$F = ma,$$

where F represents the total imposed force. However, if we let x denote the position of the object, then this may be rewritten as

$$F = m\ddot{x}.$$

Hence, Newton's Second Law of Motion is a second-order ordinary differential equation. There are many applications of DEs. Growth of microorganisms and Newton's Law of Cooling are examples of ordinary DEs (ODEs), while conservation of mass and the flow of air over a wing are examples of partial DEs (PDEs). Further, predator-prey models and the Navier-Stokes equations governing fluid flow are examples of systems of DEs. Many introductory ODE courses are devoted to solution techniques to determine the analytic solution of a given, normally linear, ODE. While these techniques are important, many real-life processes may be modeled with systems of DEs. Further, these systems may be nonlinear. Nonlinear systems of DEs may not have exact solutions. However, we still desire some type of "solution".

There are many numerical techniques to obtain an approximation to the solution of a DE or system of DEs. Many scientific packages contain library commands to numerically approximate the solution of these. In particular, MATLAB contains the *ode45* command which will numerically approximate the solution to a system of ODEs using a medium order method. Thus, the scientific package MATLAB may be used to explore DEs modeling real-world applications that are more complicated than the DEs presented in a typical introductory course. Using MATLAB also has the advantage that the internal command *guide* may be used to create graphical user interfaces (GUIs) to help facilitate this exploration. The GUIs ensure that the students are not bogged down with reading and understanding MATLAB code.

In this paper, we will discuss how to use MATLAB to simulate the solution to DEs. Then projects the author has assigned involving real-world applications of DEs will be described. One project models the industrial process of film casting while the other models the conduction of heat within a one-dimensional rod. Similar projects will also be briefly described. Finally, student feedback from the projects will be given.

II. SIMULATING SOLUTIONS TO ORDINARY DIFFERENTIAL EQUATIONS IN MATLAB

MATLAB provides many commands to approximate the solution to DEs: *ode45*, *ode15s*, and *ode23* are three examples. Suppose that the system of ODEs is written in the form

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$$

where $\mathbf{y}$ represents the vector of dependent variables and $\mathbf{f}$ represents the vector of right-hand-side functions. All of the commands (e.g., *ode45*) require three arguments:

- a filename which returns the value of the right-hand-side vector $\mathbf{f}$,
- vector representing the domain of the independent variable t,
- set of initial conditions.

Consider the DE that models driven, damped spring motion.

$$\ddot{x} + 2\lambda\dot{x} + \omega^2 x = F$$

Figure 1 contains a portion of the MATLAB code used to numerically approximate the solution to this DE.

```
TSpan = [ 0 tmax ] ;

X0 = [ x_0 x_pr_0 ] ;

[T, X] = ode45('SpringMass', TSpan, X0) ;

Plot(T, X) ;

%%
function [X_pr] = SpringMass(t, X)

global lambda omega

F = Forcing(t) ;

X_pr = [ X(2) ; -2 .* lambda .* X(2) - omega.^2 .* X(1) + F ] ;
```

Figure 1: MATLAB code.

Some notes are in order here.

- The internal commands *ode45*, *ode15s*, etc. only accept first-order DEs.
- Many higher-order DEs may be transformed into systems of first-order DEs.
- The order of the formal arguments in *SpringMass* is important.
- *T* represents the values of the independent variable *t* generated by *ode45*.
- Each row in *X* represents the value of *X* corresponding to the associated time value in *T*.

Once the solution is obtained, the output may be displayed via the internal command *plot*; frills may be added as needed. Figure 2 contains the portion of code used to display the approximation.

```
figure ;
hold on ;
plot(T, X(:, 1), '-') ;
xlabel('t') ;
ylabel('x(t)') ;
title('Displacement of Spring') ;
grid on ;
line([-1000 1000], [0 0]) ;
line([0 0], [-1000 1000]) ;
axis([0 max(T) min(X(:, 1)) max(X(:, 1))]) ;
hold off ;
```

Figure 2: MATLAB code used to produce the display of the approximation.

The natural next step is to provide students with this code and ask questions. For example, one could ask students what is the effect of changing the damping coefficient. In order for the students to answer this question, they are required to determine the line(s) in the code where the damping coefficient is defined, appropriately change those, and rerun the code. This process demonstrates a drawback to using this approach: it is difficult for the students to read and understand code.

One solution to this problem is the use of GUIs. They provide a means of introducing students to scientific packages without entirely involving the students within the code. Instead of expecting students to modify code, they may simply edit text and click a button. The required calculations are then accomplished for the students without being visible. The MATLAB internal command *guide* is the GUI development interface. It enables simplified arrangement and sizing of GUI components as well as auto-generation of code for these components. Examples of GUI components would be push buttons, axes, sliders, pop-up menus, among others. Figure 3 displays an example GUI for the spring-mass system.

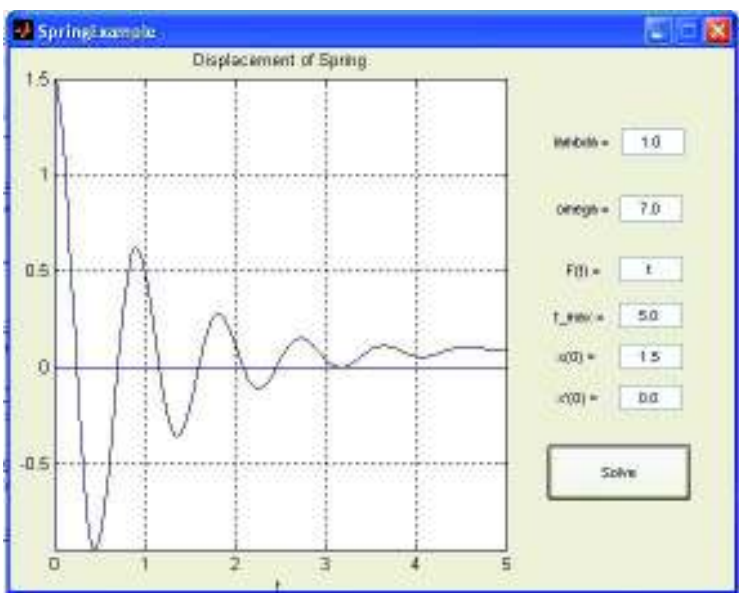

Figure 3: Example GUI for the spring-mass system.

The students may change the quantities by simply typing the new value in the edit text boxes and rerun the code by pressing the push button. With this GUI, it is easier for students to answer questions related to changing system parameters. Now the students may explore the particular model by being provided with the GUI and some auxiliary files.

III. PROJECTS

Students enrolled in an introductory ordinary differential equations course were grouped up and given different projects. Each project involved an industrial process that may be modeled by DEs. The students were asked to understand the process, why it is useful, how the process is modeled, and to present their results at a conference. There were four main thrusts of the projects.

- Expose students to a real-world process
- Expose students to a scientific software package (MATLAB)
- Demonstrate the need for numerical schemes
- Gain experience in public speaking via presentation of their results

IV. FILM CASTING

Today's society has in great abundance products that are made from polymers: clothing made from synthetic fibers, plastic bags, food wrap, and disposable diapers are among the most common examples. It has become imperative for today's manufacturers to understand the processes used to make these products as fully as possible. The processes involve complex fluid flow of a molten polymer, governed by systems of DEs.

Film casting is an industrial process in which fluid (molten polymer) is extruded from a rectangular die of thickness e_0 and length l_0, at a temperature T_0, and a velocity u_0. Please see [1], [3], [6], [9] – [12], and [13] for a detailed description of the process. The fluid is then stretched in the air due to the constant pulling force F of the chill roll, located at a distance of L from the die. The film is then cooled on this chill roll to solidify the fluid. The film casting process is shown in Figure 4.

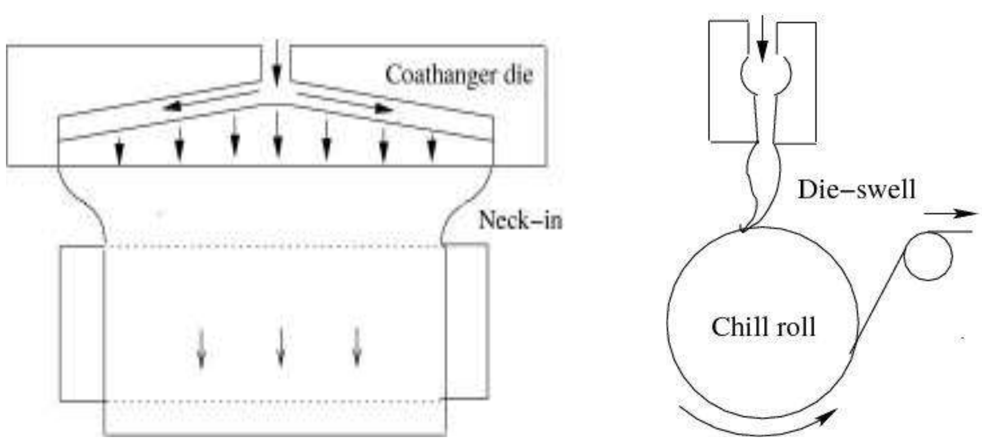

Figure 4: Schematic of the film casting process.

The five dependent variables of the flow are length l, velocity u, pulling force F, temperature T, and thickness e. Under certain assumptions, the equations governing this process may be

$$\frac{dl}{dx} = \frac{12\eta u - 2\sqrt{36\eta^2 u^2 + 2\left(\frac{F}{A}\right)^2 l^2}}{2l\frac{F}{A}}$$

$$\frac{du}{dx} = \frac{1}{24\eta}\frac{F}{A}\left(-l^2\right) \qquad \frac{dF}{dx} = \rho A u \frac{du}{dx} - \rho A g$$

$$\frac{dT}{dx} = \frac{2}{\rho C_p u e}\left[\left(T_\infty - T\right) + \varepsilon\sigma\left(T_\infty^4 - T^4\right)\right] \qquad \frac{de}{dx} = -\left(\frac{d\rho}{dx}\frac{e}{\rho} + \frac{dl}{dx}\frac{e}{l} + \frac{du}{dx}\frac{e}{u}\right)$$

given as follows.

This is a system of five first order nonlinear ODEs. Further, while the initial conditions for length, velocity, temperature, and thickness are simply those at the die exit, the initial pulling force F_0 is not known a priori. The numerical technique of shooting is used to determine the value of F_0. As opposed to attempting to solve this system analytically, it would be better to numerically approximate the solution using a numerical package (e.g., *ode45*).

Code was written that will numerically simulate the solution to these equations given a set of parameters. Moreover, a GUI was designed so that students were required to only edit the parameters. Figure 5 displays the GUI provided to the students as well as a typical solution profile. Students were asked to change the set of parameters and to determine the effect the change had on the solution.

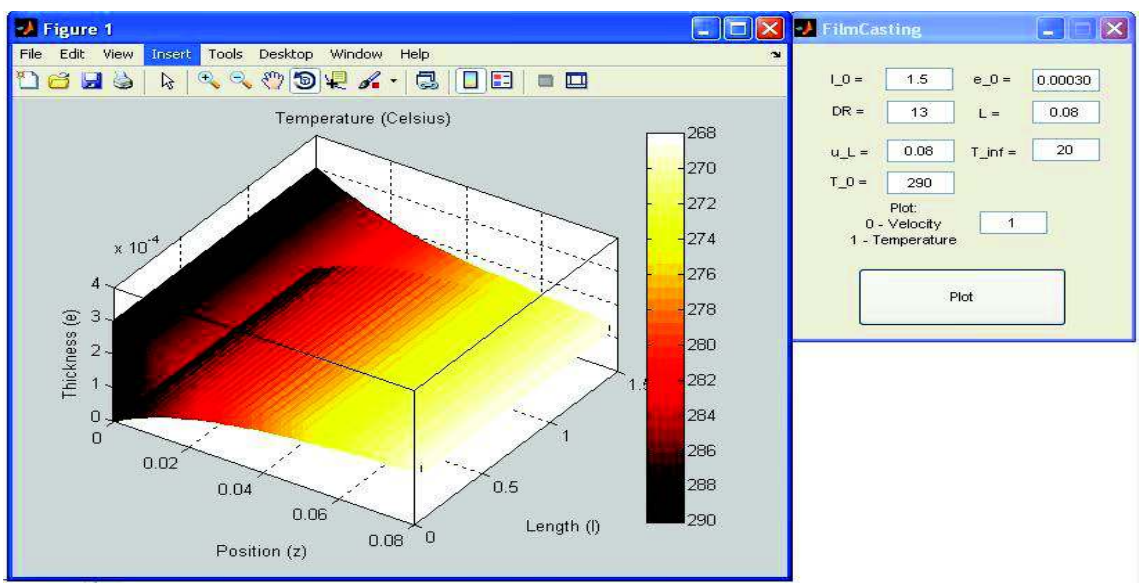

Figure 5: GUI and typical solution profile for the film casting problem.

V. HEAT CONDUCTION IN A ONE-DIMENSIONAL ROD

The manner in which heat is transferred within a one-dimensional rod may be modeled with a PDE. We assume an initial temperature distribution and desire to know how heat is conducted within the rod as time evolves. The PDE that models heat conduction may be given by

$$\frac{\partial u}{\partial t} = k\frac{\partial^2 u}{\partial x^2},$$

where k is the thermal diffusivity (see [8]). We assume that the initial temperature distribution is prescribed and that the temperature at the ends of the rod ($x = 0$ and $x = L$) are given. Hence the boundary and initial conditions are

$$\begin{aligned} u(0, t) &= 0, \\ u(L, t) &= 0, \\ u(x, 0) &= f(x). \end{aligned}$$

Since this is a PDE, the suite of ODE solvers in MATLAB are inappropriate. Hence, we choose to numerically approximate the solution to this PDE via the finite difference method (FDM). See [8] for a rough description of the FDM. The FDM first takes the continuous domain in the xt-plane and replaces it with a discrete mesh, as shown in Figure 6.

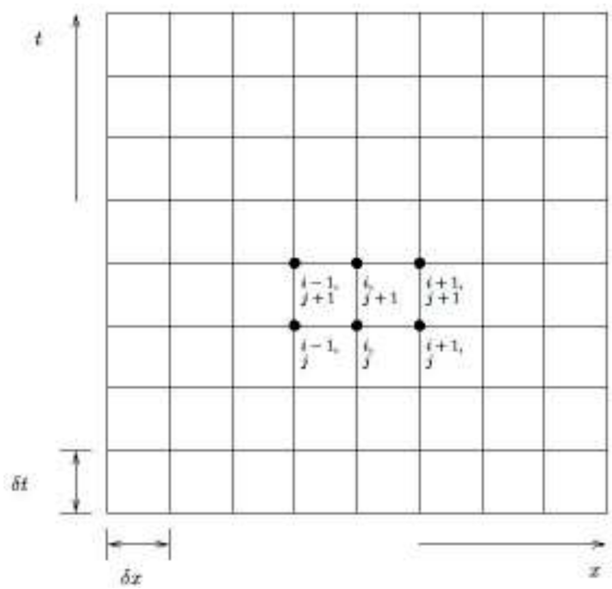

Figure 6: Example of a FDM mesh.

Next, the partial derivatives in the PDE itself are replaced with approximately equivalent difference quotients. If we let $u_j^{(m)} \approx u(x_j, t_m)$, choose to approximate $\frac{\partial u}{\partial t}$ with a forward difference and $\frac{\partial^2 u}{\partial t^2}$ with a centered difference, we obtain the following finite difference equation.

$$u_j^{(m+1)} = u_j^{(m)} + \frac{k\Delta t}{(\Delta x)^2}\left(u_{j+1}^{(m)} - 2u_j^{(m)} + u_{j-1}^{(m)}\right)$$

Code was written to solve the finite difference equation and display the results. Once the code was working properly, a GUI was designed to allow students to numerically approximate the solution for a given parameter set. Figure 7 shows the GUI as well as a solution profile for a parameter set.

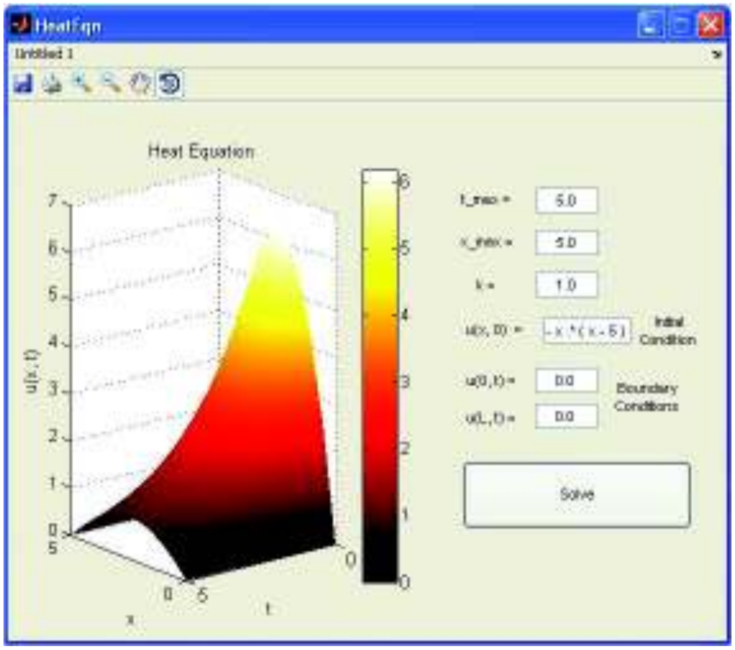

Figure 7: GUI and solution profile for heat conduction in a one-dimensional rod.

Students were asked to change the initial condition and the thermal diffusivity to determine the effect on the solution profile.

VI. FIBER SPINNING PROJECTS

Two projects involved the industrial process of fiber spinning (see [2], [4], [5], [7], and [14]).. In this process, an axisymmetric stream of polymer melt is extruded from a spinneret containing thousands of capillaries. The spinneret may be thought of roughly as a shower head. The polymer melt is then drawn continuously by the take-up rolls. Along the length of the stream, there is a region where cooling air is applied. This process is illustrated in Figure 8.

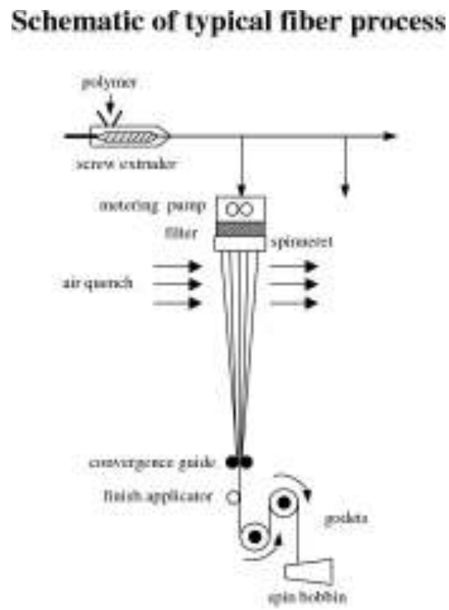

Figure 8: Schematic of the fiber spinning process.

Depending upon the processing parameters, the polymer may enter a semi-crystalline phase where the structure of the molecules of the polymer cannot change. One project explored fiber spinning excluding the semi-crystalline phase, while the other project considered both phases. Both processes are governed by systems of nonlinear ordinary differential equations.

VII. WAVE PROPAGATION PROJECT

The final project explored the propagation of waves on a taut string (see [8]). This process is governed by the following PDE.

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

Here c represents the propagation speed of the wave. The following boundary and initial conditions are enforced.

$$u(0, t) = 0$$
$$u(L, t) = 0$$
$$u(x, 0) = f(x)$$
$$\frac{\partial u}{\partial t}(x, 0) = g(x)$$

Figure 9 displays the GUI provided to the students as well as an example solution profile.
\

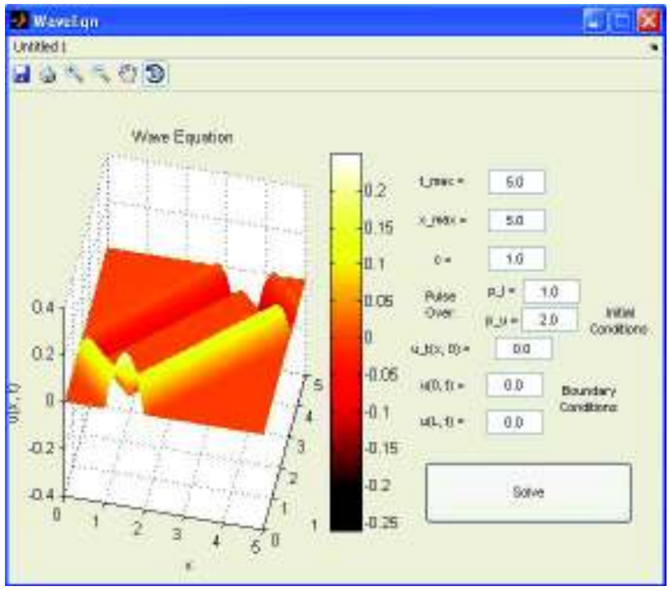

Figure 9: GUI and solution profile for wave propagation on a taut string.

Students were asked to change the initial displacement of the string and the propagation speed to determine the effect on the solution profile.

VIII. STUDENT FEEDBACK

Student response from these projects differed between the objectives of the projects. Student responses were generally positive about exploring real-world applications of DEs and the experience gained in public speaking. However, the responses were generally mediocre for the numerical schemes and the exposure to MATLAB. Many students were surprised that DEs modeling real-world applications could not be solved analytically. Overall, the students gave the projects positive reviews and considered them a success.

REFERENCES

[1] Acierno, D., L. Di Maio, C. Ammirati, "Film Casting of Polyethylene Terephthalate: Experiments and Model Comparisons," *Polymer Engineering and Science*, Vol. 40, pp. 108-117, 2000.
[2] Advani, S., C. Tucker III, "The Use of Tensors to Describe and Predict Fiber Orientation In Short Fiber Composites," *Journal of Rheology*, Vol. 31, pp. 751-784, 1987.
[3] Baird, D., D. Collais, *Polymer Processing*, John Wiley and Sons, New York, 1998.
[4] Denn, M., *Process Modeling*, Longman Inc., New York, 1986.
[5] Doufas, A., A, McHugh, C. Miller, "Simulation of Melt Spinning Including Flow-Induced Crystallization Part I: Model Development and Predictions," *Journal of Non-Newtonian Fluid Mechanics*, Vol. 92, pp. 27-66, 2000.
[6] Eisele, P., R. Killpack, "Propene," *Ullman's Encyclopedia of Industrial Chemistry*, 1993 ed.
[7] Fisher, R., M. Denn, "Mechanics of Nonisothermal Polymer Melt Spinning," *American Institute of Chemical Engineers Journal*, Vol. 23, No. 1, pp. 23-28.
[8] Haberman, R., *Elementary Applied Partial Differential Equations*, Prentice Hall, New Jersey, 1987.
[9] Lide, D., ed. *CRC Handbook of Chemistry and Physics*, CRC Press, New York, 2001.
[10] Mark, J., ed. *Physical Properties of Polymers Handbook*, AIP Press New York 1996.

[11] Smith, S., D. Stolle, “Nonisothermal Two-Dimensional Film Casting of a Viscous Polymer,” *Polymer Engineering and Science*, Vol. 40, pp. 1870-1877, 2000.

[12] Vargaftik, N., *Tables on the Thermophysical Properties of Liquids and Gases*, John Wiley and Sons, New York, 1975.

[13] Ziabicki, A., *Fundamentals of Fibre Formation*, John Wiley and Sons, New York, 1976.

[14] Ziabicki, A., L. Jarecki, A. Waiak, “Dynamic Modelling of Melt Spinning,” *Computational and Theoretical Polymer Science*, Vol. 8, pp. 143-157, 1998.