PRIMUS
Problems, Resources, and Issues in Mathematics Undergraduate Studies
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t776095548

Teaching Modeling with Partial Differential Equations: Several Successful Approaches

Joseph Myers; David Trubatch; Brian Winkel

Online Publication Date: 01 March 2008
To cite this Article: Myers, Joseph, Trubatch, David and Winkel, Brian (2008) 'Teaching Modeling with Partial Differential Equations: Several Successful Approaches', PRIMUS, 18:2, 161 - 182
To link to this article: DOI: 10.1080/10511970701643889
URL: http://dx.doi.org/10.1080/10511970701643889

PRIMUS, XVIII(2): 161–182, 2008

ISSN: 1051-1970 print / 1935-4053 online
DOI: 10.1080/10511970701643889

Teaching Modeling with Partial Differential Equations: Several Successful Approaches

Joseph Myers, David Trubatch, and Brian Winkel

Abstract: We discuss the introduction and teaching of partial differential equations (heat and wave equations) via modeling physical phenomena, using a new approach that encompasses constructing difference equations and implementing these in a spreadsheet, numerically solving the partial differential equations using the numerical differential equation solver in Mathematica, and analytically constructing solutions from reasoned building blocks. We obtain graphical feedback as soon as possible in each approach and permit "what if" modeling wherever possible. This approach is contrasted with the usual Fourier series development and series solution using boundary value solution strategies.

Keywords: Mathematical modeling, partial differential equation, heat and wave equation, numerical solution, difference equation, spreadsheet, Mathematica, analytic solution, graphical feedback.

PARTIAL DIFFERENTIAL EQUATIONS IN UNDERGRADUATE MATHEMATICS

Partial differential equations (PDEs) appear in a wide variety of course settings and in a number of levels of mathematical sophistication throughout undergraduate mathematics curricula. At a relatively simple level, they can be found immediately after the introduction of partial derivatives in a multivariable calculus course in exercises that require students to confirm that a given function of more than one variable satisfies a given PDE. At the other end, because of their intrinsic mathematical richness, PDEs can be the focus of a high-level course that requires familiarity with at least the basics of analysis. In between, PDEs are often associated with applied mathematics, as in a course in mathematical methods for physicists or engineers or more recently in financial mathematics.

Address correspondence to Brian Winkel, Department of Mathematical Sciences, United States Military Academy, West Point, NY 10996, USA. E-mail: Brian.Winkel@usma.edu

While the most mathematically sophisticated aspects of PDEs are beyond the undergraduate curriculum, the solution of linear, constant-coefficient PDEs requires only the basics of multivariable calculus. Moreover, the classical second-order equations (the heat equation, the wave equation, and Laplace's equation) have both a long mathematical history and accessible physical applications. Thus, not surprisingly, many introductions to PDEs start with these equations (cf. [3–5]). The most common approach to the solution of these linear partial differential equations is the technique of separation of variables. In contrast, in this article, we offer some alternative approaches to the study of evolution equations, the heat and wave equations.

We note that separation of variables leads immediately to boundary-value problems and Fourier series. Therefore, when taking the standard approach to PDEs, the instructor must decide whether to do Fourier series first or to have them motivated at the moment of need in the solution algorithm for the PDEs. When we have taken this route, we have elected to introduce the Fourier series first and to ask students to derive Fourier coefficients from a least squares criterion with graphical feedback (cf. [7,8]). When teaching this course, we pause to play with the power of the Fourier series in sound applications, spectral analysis, and signal processing. Once armed with the notions of Fourier series, the dive into separation of variables offers a wonderful place to see the Fourier coefficients emerge in the development of the infinite series solution in which the coefficients are the Fourier series for the initial conditions. Similarly, as described in this article, we introduce the analytic solution of the heat equation some time after the discussion of Fourier series.

Over the years we have followed the approach described within two settings: (i) a mathematical modeling course for mathematics majors and (ii) a post-calculus engineering mathematics. In both cases, we have emphasized the exploration of the solutions of PDEs and the interplay between physical and mathematical intuition in the study of PDEs as mathematical models of physical phenomena. We usually study two physical phenomena: (i) heat flow in an insulated bar, modeled by the one-dimensional (in space) heat equation and (ii) the transverse oscillations of a string held down at both ends, modeled by the one-dimensional (again in space) wave equation. For the heat flow modeling we usually give our students a guided opportunity to build their own model (as outlined below). For the latter, we typically refer to the course text [5] and/or an exceptionally clear exposition in a well-known text [3] and do not spend significant course time on the derivation. Instead, the class moves directly to an examination of the solutions. As noted, these equations are, of course, two of the three standard second-order equations studied in a first course on PDEs. Our approach is therefore not distinguished by the equations studied but rather by our method of approach.

MATHEMATICAL MODELING

Wherever possible, we incorporate either modeling from first principles (e.g., Newton's laws of mechanics) or from data sets used throughout the mathematics curriculum at the United States Military Academy. In some cases the modeling is used to motivate the study of the mathematics, while in others the modeling is in the application of the mathematics under study. In this article, we demonstrate both cases.

Modeling takes time, both with in-class activities and out-of-class assignments. When in class, faculty must give time for students to discover strategies and solutions on their own, perhaps with some teacher direction or guidance. It is very important to let the students feel the uncertainty of the moment in which small and cautious steps, some good and some bad(!), are taken in the model-building process. This is best done in small groups so that no one student faces the situation alone. The groups usually generate (and reject) many ideas and will move out on their own. However, it is a good idea to interrupt the process in class to ask one group that is making particularly good progress to share their process and conclusions. Other groups might then redirect their efforts because of these presentations, but they bring their own ideas and experience to this new level of modeling based on communal knowledge. Occasionally, faculty lead or summarize the steps for success in modeling, but the students, having invested time and intellectual capital in the process are truly shareholders, not just consumers, of such teacher-led time.

Usually, it is a good idea to ask students to do extensions of the model under consideration for homework or project activities, sometimes with significant change in the model and other times using the model derived for computational efforts along with "what if?" study with various parameters and physical interpretations. In our experience the flow from physical phenomena, to equation formulation, to numerical solutions, and then to analytic solutions, holds students' attention, as it gives them ownership and the ability to practice both equation formulation, solution strategies, and interpretation of solutions. Moreover, a continuing emphasis on the graphical representation of solutions makes the subject concrete.

MODELING HEAT CONDUCTION IN A THIN ROD

Overview

We consider a conducting rod of length L with constant cross-sectional area A. The rod is insulated along its lateral surface so heat is constrained to flow along its length. We make the (reasonable) simplifying assumption that the material is uniform and unchanging throughout the experiment, except for its

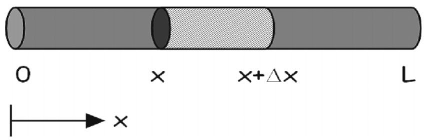


Figure 1. "Plug" of a thin conducting rod.

heat energy content. Moreover, inside the bar, heat energy is neither created nor destroyed (e.g., by a chemical or radiological process). The variable x is the distance measured along the rod from left to right (see Figure 1). The function $u(x, t)$ is the temperature of the rod at position x and time t from the start of the experiment.

We direct the students to focus on a small "plug" of rod between x and $x + \Delta x$. Our strategy is to give the students enough grounding in physics and time for discussion so that they can obtain expressions for the rate of change of heat energy in the plug of the rod in two different ways: (i) the (time) rate of change in the thermal energy in the plug and (ii) the net change in heat energy due to the heat flow at each of the two ends. The equality of these two rates (as required by conservation of heat energy) then leads to the partial differential equation that governs the evolution of the temperature distribution (namely, the heat equation).

Standard Derivation

To compose expressions for the two rates of change we rely on two basic physical laws:

Law 1 The amount of thermal energy in the body is proportional to the temperature of the body times the mass of the body.

Law 2 Thermal energy flows across an uninsulated interface at a rate proportional to the area and the temperature gradient. (The temperature gradient is the rate of change of temperature with respect to *distance* where the distance is taken perpendicular to the area).

We usually get the students to articulate these laws in small groups and then structure them with the above wording. For the students to proceed to build a model, they also need some technical terms. Moreover, they need to be reminded, constantly, to keep track of units!

To determine the expression for the time rate of change in the amount of thermal energy in the plug we first build an expression for the amount of thermal energy in the plug. This calculation follows from Law 1 with careful attention to units:

$$\text{thermal energy} = c\left[\frac{\text{cal}}{\text{g}\cdot{}^{\circ}C}\right]\rho\left[\frac{g}{\text{cm}^3}\right]A\left[\text{cm}^2\right]\Delta x[\text{cm}]\ u(x,t)[{}^{\circ}C],$$

where c is the specific heat and ρ is the mass density. Canceling units, we have the following expression for the amount of thermal energy in the plug:

$$\text{thermal energy} = c\,\rho A\,\Delta x\,u(x,t)\ [\text{cal}].$$

From this, we obtain the (time) rate of change in the thermal energy in the plug as:

$$\text{rateofchange} = \frac{\partial}{\partial t}\text{thermal energy} = c\,\rho A\,\Delta x\,u_t(x,t)\left[\frac{\text{cal}}{\text{min}}\right], \tag{1}$$

where, following our previous assumptions, the dimensions of the rod and its specific heat are constant.

We then compute the time rate of thermal energy going into (out of) the left side of the plug and the same on the right side of the plug. From Law 2 we have:

$$\text{heat flow into rod at } x\left[\frac{\text{cal}}{\text{min}}\right] = -k\,A\,[\text{cm}^2]\ \left.\frac{\partial u}{\partial x}\right|_x\ \left[\frac{{}^{\circ}C}{\text{cm}}\right],$$

$$\text{heat flow into rod at } x+\Delta_x\left(\frac{\text{cal}}{\text{min}}\right) = k\cdot A\,(\text{cm}^2)\cdot\left.\frac{\partial u}{\partial x}\right|_{x+\Delta x}\left(\frac{{}^{\circ}C}{\text{cm}}\right),$$

where the units of k, the *thermal conductivity*, must be $\frac{\text{cal}}{\text{min}\cdot\text{cm}\cdot{}^{\circ}C}$. The net flow of thermal energy into (or out of) the plug is therefore

$$\text{net heat flow} = kA[u_x(x+\Delta x,t) - u_x(x,t)]\ \left[\frac{\text{cal}}{\text{min}}\right] \tag{2}$$

As noted above, invoking the principle of conservation of energy, we equate (1) and (2). Simplifying and omitting the units, we obtain:

$$u_t(x,t) = \frac{k}{c\rho}\frac{u_x(x+\Delta x,t) - u_x(x,t)}{\Delta x}. \tag{3}$$

Then in the limit as $\Delta x \to 0$ we obtain the partial differential equation:

$$u_t = \alpha u_{xx} \tag{4}$$

where the constant $\alpha = \frac{k}{c\rho}$, the thermal diffusivity, has units of cm^2/sec. Equation (4) is the standard one-dimensional heat equation.

In order to have a completely posed initial value problem, we need information about the initial temperature distribution in the rod; i.e., $u(x, 0) = f(x)$ for some given function $f(x)$. Furthermore, we need to specify conditions at the ends of the rod; e.g., $u(0, t) = h_1(t)$ and $u(L, t) = h_2(t)$, where $h_1(t)$ and $h_2(t)$ are known functions of time.

Before designing and implementing a numerical scheme to solve the heat equation, we ask students to predict the behavior of the thermal energy based on their physical intuition. For example, with the conditions $u(x, 0) = f(x) = x(L-x)$ and, at the boundaries, $u(0, t) = h_1(t) = 0$ and $u(L, t) = h_2(t) = 0$ (i.e., frozen), students will, in general, correctly predict that the initial temperature distribution is symmetric about the longitudinal midpoint of the rod and that the heat will escape through the "frozen" ends until the rod has a uniform temperature of $0°C$. In fact, in our experience, they can sketch a solution over the x-t domain that looks qualitatively like the exact solution.

Alternative Derivation

An alternative derivation of the heat equation follows from an alternative form of Law 2. If one thinks of each plug as so small that it has no temperature gradient across itself, then there is a jump in temperature at each of the interfaces. Accordingly, Law 2 is modified to:

Law 2′ Heat flows across an uninsulated interface at a rate: proportional to the area of the interface, proportional to the difference in temperature and inversely proportional to the length of the plug.

To write Law 2′ in mathematical form, we denote the temperature in the nth plug as u_n. For the net heat flow, one then obtains the expression

$$\text{net heat flow} = k\frac{A}{\Delta x}(u_{n+1} - u_n) + k\frac{A}{\Delta x}(u_{n-1} - u_n),$$

which can be rewritten as:

$$\text{net heat flow} = kA\frac{u_{n+1} - 2u_n + u_{n-1}}{\Delta x}. \tag{5}$$

In this notation, (1) is rewritten as

$$\text{rate of change} = \frac{d}{dt}\text{thermal energy} = c\rho A\Delta x\frac{d}{dt}u_n. \tag{6}$$

Then, to obtain the heat equation, we again invoke the principle of conservation of energy and equate (6) and (5), which yields:

$$\frac{d}{dt}u_n = \frac{k}{c\rho}\frac{u_{n+1} - 2u_n + u_{n-1}}{\Delta x^2}. \tag{7}$$

Now, because the temperature is constant in each plug, we can identify u_n with the temperature on the left-hand side of that plug, $u_n(t) = u(n\Delta x,\ t)$. Thus, (7) is equivalent to

$$\frac{\partial}{\partial t}u(x,t) = \frac{k}{c\rho}\frac{u(x+\Delta x,t) - 2u(x,t) + u(x-\Delta x,t)}{\Delta x^2},$$

where $x = n\Delta x$. In the limit as $\Delta x \to 0$ we again obtain the partial differential Equation (4).

We remark that Law 2´ is consistent with Law 2 in the limit as the width of the plug goes to zero. Thus, in some sense, the standard derivation that makes use of Law 2 requires taking the limit $\Delta x \to 0$ twice, while the alternative derivation requires taking the limit only once.

MOVING TO DIFFERENCE EQUATIONS AND A SPREADSHEET IMPLEMENTATION

Numerical scheme

For the purpose of numerical simulation of the heat equation, we have the students approximate the continuous partial differential Equation (4) with a partial difference equation. To do this, they need to revisit the definition of derivative that they applied in converting Equation (3) to a partial differential equation. The time-evolution can be written as:

$$u_t(x,t) = \lim_{\Delta t \to 0}\frac{u(x,t+\Delta t) - u(x,t)}{\Delta t}$$

Therefore, the left-hand side of the heat equation in Equation (4) can be replaced by the finite difference

$$u_t(x,t) \approx \frac{u(x,t+\Delta t) - u(x,t)}{\Delta t}. \tag{8}$$

Similarly, we have

$$u_{xx}(x,t) = \lim_{\Delta x \to 0}\frac{\frac{u(x+\Delta x,t)-u(x,t)}{\Delta x} - \frac{u(x,t)-u(x-\Delta x,t)}{\Delta x}}{\Delta x}.$$

and, consequently,

$$u_{xx}(x,t) \approx \frac{u(x+\Delta x,t) - 2u(x,t) - u(x-\Delta x,t)}{\Delta x^2}. \tag{9}$$

We note that Equation (9) is equivalent to Equation (5).

Substituting the difference approximations (8)–(9) for the respective partial derivative terms in the heat equation (and simplifying), we obtain a partial difference equation that permits us to predict the spatial temperature distribution at a "future" time (i.e., $u(x, t + \Delta t)$) given the "current" temperature distribution (i.e., $u(x, t)$). Specifically,

$$u(x,t+\Delta t) = u(x,t) + \left[\frac{\alpha\Delta t}{(\Delta x)^2}\right][u(x+\Delta x,t) - 2u(x,t) + u(x-\Delta x,t)] \tag{10}$$

where the equality is, in fact, an approximation whose local error shrinks with Δt and Δx.

We remark that in Equation (9) the central differencing follows naturally from the fact that heat flows to (or from) the point x from both the left, $x - \Delta x$, and the right, $x + \Delta$. In fact, the central difference is a step "back" to the difference Equation (5) in the alternative derivation of the heat equation. In contrast, in Equation (8) we choose forward differencing specifically because we wish to predict the thermal energy distribution of the rod at future times, $t + \Delta t$, based on the current distribution at time t. (For students who are curious, the implementation of the ill-posed *backwards* difference equations provides an interesting area for investigation and further study).

Now, given: (i) a numerical value for the constant α, which depends upon the nature of the material; (ii) an initial condition $u(x, 0) = f(x)$; (iii) two boundary conditions, $u(0, t) = h_1(t)$ and $u(L, t) = h_2(t)$; and (iv) values for Δx and Δt, we can "march" forward in time with Equation (10). In this way, we estimate the diffusion of thermal energy along the length of the rod.

Spreadsheet Implementation

To execute the numerical simulations, we take time in class for the students to implement Equation (10) with a spreadsheet (see Figures 2 and 3.) We start with a blank spreadsheet and build with the students input. If the facilities permit, the students can build their own spreadsheets simultaneously. We are careful to lead the students to good stewardship of their spreadsheet by identifying parameters needed that they could possibly change, specifically a, Δt, and Δx.

From the spreadsheet, we see concretely how to advance forward in time from the given initial condition and inward from the boundary conditions at each end of the rod. We get the students to draw a picture of how Equation (10) demonstrates the stepping, in time, of the thermal energy flow solution. We illustrate this in summary in Figure 3.

From here, students immediately plot their solution to see that it confirms their physical understanding of thermal energy flow in an insulated rod with both ends in ice; i.e., $u(0, t) = 0$ and $u(1, t) = 0$. We illustrate this in Figure 4.

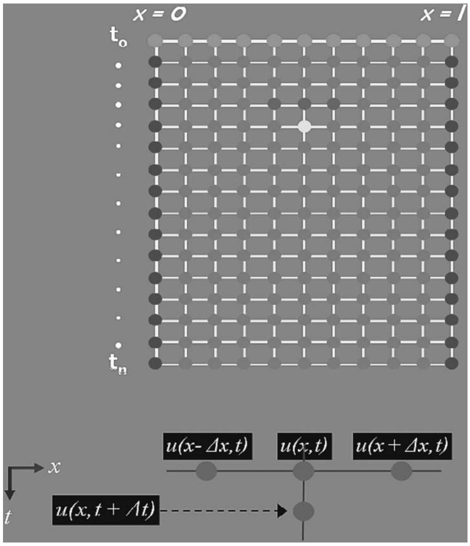


Figure 2. Illustration of how students "see" the building of the spreadsheet, especially the computation of $u(x,\ t + \Delta t)$ from nearby cells, $u(x - \Delta x,\ t)$, $u(x,\ t)$, and $u(x + \Delta x,\ t)$, all of which are at the previous time increment.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1				**Modeling Heat Equation on Spread Sheet**										
2					**u_t(x,t) = a u_xx(x,t).**									
3														
4		**Rod**			**a =**	0.3	**dt =**	0.01	**dx =**	0.1	**L =**	1		
5	**Time**	**Across**												
6	**Down**	**0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**		
7	**0**	0	0.09	0.16	0.21	0.24	0.25	0.24	0.21	0.16	0.09	0	**<--- Initial Condition**	
8	**0.01**	0	0.084	0.154	0.204	0.234	0.244	0.234	0.204	0.154	0.084	0		
9	**0.02**	0	0.08	0.148	0.198	0.228	0.238	0.228	0.198	0.148	0.08	0		
10	**0.03**	0	0.076	0.143	0.192	0.222	0.232	0.222	0.192	0.143	0.076	0		
11	**0.04**	0	0.073	0.138	0.186	0.216	0.226	0.216	0.186	0.138	0.073	0		
12	**0.05**	0	0.071	0.133	0.181	0.21	0.22	0.21	0.181	0.133	0.071	0		
13	**0.06**	0	0.068	0.128	0.175	0.204	0.214	0.204	0.175	0.128	0.068	0		
14	**0.07**	0	0.066	0.124	0.17	0.198	0.208	0.198	0.17	0.124	0.066	0		
15	**0.08**	0	0.064	0.12	0.165	0.193	0.202	0.193	0.165	0.12	0.064	0		
16	**0.09**	0	0.062	0.117	0.16	0.187	0.197	0.187	0.16	0.117	0.062	0	**We hold both ends**	
17	**0.1**	0	0.06	0.113	0.155	0.182	0.191	0.182	0.155	0.113	0.06	0	**of the bar in ice,**	
18	**0.11**	0	0.058	0.11	0.151	0.177	0.185	0.177	0.151	0.11	0.058	0	**u(0,t) = u(L,t) = 0.**	
19	**0.12**	0	0.056	0.106	0.146	0.171	0.18	0.171	0.146	0.106	0.056	0		
20	**0.13**	0	0.054	0.103	0.142	0.166	0.175	0.166	0.142	0.103	0.054	0		
21	**0.14**	0	0.053	0.1	0.138	0.162	0.17	0.162	0.138	0.1	0.053	0		
22	**0.15**	0	0.051	0.097	0.134	0.157	0.165	0.157	0.134	0.097	0.051	0		
23	**0.16**	0	0.05	0.094	0.13	0.152	0.16	0.152	0.13	0.094	0.05	0		
24														
25				**From discretization we have**										
26					**u(x,t + Δt) = u(x,t) + Δt*a/(Δx)^2*(u(x+Δx,t) - 2 u(x,t) + u(x-Δx,t))**									
27														
28				**Typical cell implementation in EXCEL**										
29					**c8 = c7 + h4*e4*(d7-2c7+b7)/(j4)^2**									

Figure 3. Finished spreadsheet for heat equation $u_t = \alpha u_{xx}$ with initial condition $u(x,\ 0) = x(1 - x)$ and boundary conditions $u(0,\ t) = 0$ and $u(1,\ t) = 0$, where $\alpha = 0.3$, $\Delta t = 0.01$, and $\Delta x = 0.1$.

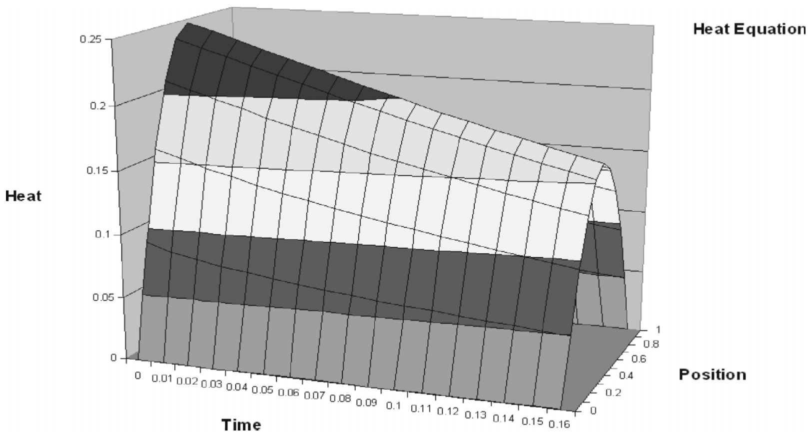


Figure 4. Plot of the numerical solution in Figure 3.

One advantage of the numerical simulation over the analytical solution is that the spreadsheet can be easily modified to investigate inhomogeneous boundary conditions. Specifically, we can modify the implementation to simulate a situation in which the temperature at one end varies periodically in time; e.g., $u(L, t) = \sin(t)$, for $0 \leq t$. (In contrast, the analytical solution of the heat equation with a time-periodic boundary condition involves several additional non-trivial steps).

In our experience, students can develop the heat equation, formulate the finite-difference scheme, and build a spreadsheet implementation in two to three 55-minute class periods. We have the students read the text material [2, 5] before class and come prepared to present one of the two approaches for computing the time rate of change of thermal energy in the plug outlined in the text.

We usually put two students from each computational approach into a four-person group. Here, we have each pair, in turn, convince the other pair of their way of accounting for the time rate of change of thermal energy and then acknowledge the fact that each pair has computed the same thing; hence, they have an equation by setting their computed values equal to each other. We collectively get to the difference equations and have the students build their spreadsheets, all for the same heat problem and conditions. We obtain a plot and play with several initial and boundary conditions to see that the model we have built makes sense.

Numerical Instability

Finally, we conduct a naïve investigation for stability and get the spreadsheet to blow up! We do this by first surreptitiously calculating the stability limit

$\frac{1}{2}\frac{\Delta x^2}{\alpha}$ somewhere on the side of our spreadsheet, where students can't see it. Then we choose a value for the time step Δt that is greater than this value. Executing the spreadsheet immediately leads to disaster: it fills with numbers that oscillate between positive millions and negative millions. A graph of the temperature shows initially small oscillations that grow without bound. We wonder aloud with students: what might have gone wrong? We suggest that maybe we were too greedy in getting to the desired final time in too few steps. We pick a new time step that happens to be less than our calculated stability limit and then recalculate and show how everything is now nicely behaved as we expect. Only then do we draw attenting to our stability limit calculation and the fact that our first computation violated it and our second did not.

We do not derive or try to explain this limit in any detail; that is not our purpose in presenting it.

Instead, we explain that our purpose was to show them that finite difference methods similar to what we just developed cannot be taken for granted; that there are lots of very reasonable ways to discretize partial differential equations that will end up in total failure, and so people attempting numerical solutions should generally find known algorithms with known stability properties and stay within the prescribed bounds. We also emphasize that this is a lesson in using almost any computational package; stability, convergence, and applicability of the algorithm featured in software package A to the physical features of problem B can be real issues, and therefore the user must remain vigilant in inspecting package output to see if it is consistent with various bounds and common-sense checks. It is not necessarily the fault of the package but is rather a manifestation of the limitation of numerical algorithms and the complexity of the real world.

As an aside, we note that the conditional stability of our explicit method can be explained and analyzed by a von Neumann stability analysis, which makes use of the fact that the spatial Fourier modes of the equation evolve independently of one another (cf. [6]). In fact, one can construct an implicit scheme (Crank-Nicolson) that is unconditionally stable (also shown by von Neumann stability analysis). However, this material is beyond what we have discussed with our students.

FURTHER EXPLORATIONS WITH A "BLACK BOX" NUMERICAL SOLVER

Solution to the Original Problem

Our students have seen Euler's method in our core mathematics courses for ordinary first-order differential equations and they have used Mathematica's NDSolve function to solve ordinary differential equations that cannot be easily solved analytically, either by hand or with Mathematica's DSolve function. Therefore, they have some idea of what a numerical solution really means. With that background, we coach them on the use of NDSolve to solve the heat equation. The code in Listing 1 generates a numerical solution of the heat equation. We note that, in Mathematica, the parentheses-asterisk combination denotes a comment.

The result of the numerical calculation in Listing 1 is assigned to the variable sol. Mathematica returns the result of the numerical simulation as a Mathematica Replacement Rule whose object is a Mathematica Interpolating function. (Note that a Mathematica installation contains complete documentation, including a discussion of these structures). However, fortunately, it is not necessary to master these aspects of the Mathematica language to generate a numerical solution and the corresponding plot. The code in Listing 2 generates a three-dimensional plot of the solution found in Listing 1 that is stored in sol. The plot options have been specified so as to improve the clarity and appearance of the plot. The resulting plot is in Figure 5.

```
a=.3; (* Coefficient in Heat Equation *)
L=1; (* Length of the Rod *)
time=1; (* Final time *)
sol=
   {D[u[x,t],t] == a*D[u[x,t],{x,2}], (* Heat Equation *)
    u[x,0] == x*(L-x), (* Initial Condition *)
    u[0,t] == 0,   (* Boundary Condition at 'x=0' *)
    u[L,t] == 0},(* Boundary Condition at 'x=L' *)
    u[x,t], (* The function to be solved *)
   {x, 0, L}, (* Solve from 'x=0' to 'x=L' *)
   {t, 0, time} (* Solve from 't=0' to 't=time' *)
  [[1]];
```

Listing 1: Numerical solution of the heat equation.

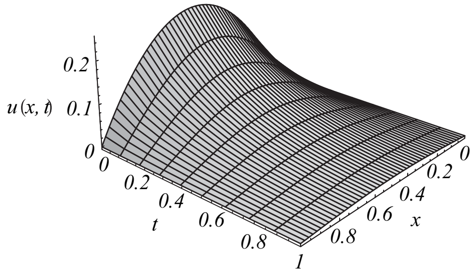


Figure 5. Plot of numerical solution generated with Mathematica code in Listing 2.

```
Plot3D[
  u[x,t]/.sol //Evaluate, (* Use the solution 'sol' *)
  {x,0,L}, (* Plot from 'x=0' to 'x=L' *)
  {t,0,time},(* Plot from 't=0' to 't=time' *)
  PlotPoints -> {50,10},(* Set the plot resolution. *)
  ViewPoint -> {2.4, 2.1, 1.2},
  AxesEdge -> {{+1,-1}, Automatic, Automatic},
  DefaultFont -> {"Times-Italic",12},
  AxesLabel -> {"x","t","u(x,t)"}
];
```

Listing 2: Generate a plot of the solution sol.

Animation

In addition to a three-dimensional plot, one can visualize the solution by generating an animation of the time evolution of the temperature distribution in the rod. In a Mathematica notebook, one first generates the frames and then animates them. The code in Listing 3 generates 41 frames from $t = 0$ to $t = 2$ in increments of 0.05. We illustrate the results in Figure 6 with four

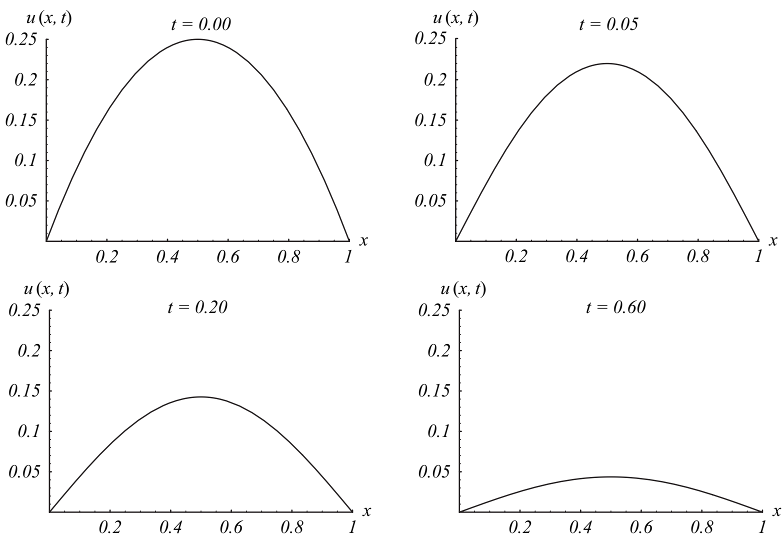


Figure 6. Four snapshots generated by the code in Listing 3.

"snapshots" of the temperature $u(x, t)$ in the rod over $0 \leq x \leq 1$ and at times $t = 0.00$, 0.05, 0.20, and 0.60. The animation is played in the Mathematica notebook by double-clicking on any one of the many frames generated for the animation. Moreover, the animation can be stepped, slowed down, reversed, and looped. Animations can be saved as animated gif files and work independent of Mathematica. We have found that animations are very effective in helping students (and faculty!) "see" their solutions, the success of their modeling, and the ramifications of parameter changes.

```
Table[
   Plot[u[x,t]/.sol//Evaluate, (* Use the solution 'sol'. *)
      {x, 0, 1},  (* Plot u(x,t) from 'x = 0' to 'x = 1'. *)
      PlotRange -> {{0, 1}, {0, .25}}, (* Maintain a consistent
                                          plot window. *)
      DefaultFont -> {"Times-Italic", 12},
      AxesLabel -> {"x", "u(x,t)"},
      PlotLabel -> StringForm["t = `1`", PaddedForm[t,{3,2}]
   ],
   {t, 0, 1, .05} (* Step 't' from 0 to 1 in
                     increments of .05. *)
  ];
```

Listing 3: Code that generates the animation frames of the numerical solution.

Different Boundary Conditions

The numerical solver is quite flexible and allows us to easily consider a number of scenarios. In particular, we can investigate the affect of changing the boundary conditions. For example, we ask: "How would we model insulating one end of the rod?" This takes some time to get the students to produce. We have told them modeling is not easy and here is an instance.

We give the students hints to help them determine the appropriate mathematical boundary condition: (1) If one end is sealed then no thermal energy crosses that boundary. (2) No thermal energy will flow if the temperature gradient at the boundary is zero. Consequently, the insulating boundary condition at $x = L$ is $u_x(L, t) = 0$, where the rod is insulated at $x = L$. They "get" it, but then ask how to implement this in Mathematica's NDSolve function. The Mathematica code for an insulating boundary condition at $x = 0$ is

```
Derivative[0,1][u][L,t]==0
```

where L corresponds to $x = L$. This code can be substituted into the code in Listing 1 as a replacement boundary condition at $x = L$.

For the numerical simulation, we select a slightly different initial condition to avoid a discontinuity at the corner $(x, t) = (0, 0)$. That is, if $u(x, 0) = f(x) = x(1 - x)$, then $u_x(1, 0) = f'(1) \neq 0$ in contradiction to the insulating boundary condition. To avoid the discontinuity, we choose $f(x) = x(1 - x)^2$, in which case $f'(1) = 0$. Thus, the corner discontinuity/inconsistency is not a theoretical problem. The numerical methods used by Mathematica 5.2, however, do not handle the discontinuity well, instead generating a solution with spurious oscillations. This is another illustration of the fact that numerical simulations are nontrivial and their results cannot be accepted unquestioningly.

A plot of the solution with the insulating boundary condition is given in Figure 7. Notice, as do the students, that, at the insulated end ($x = 1$) the temperature rises as the heat in the center of the rod flows toward both ends (which are cooler than the middle). Unlike the "frozen" end ($x = 0$), in this case the heat accumulates at the insulated end and the temperature rises. Then, when temperature in the center falls below that on the insulated end, the heat flows toward the frozen end. The visual feedback both confirms and helps students further develop their intuition about the solution.

More Variations

In addition to modifications to the end of the rod, e.g., holding the end(s) in ice, sealing off the ends, and varying the temperature on end(s) over time, we did such things as "detect" a leak in the insulation surrounding the rod. In particular, we modeled the insulation so that a small portion of it along the rod was exposed to room temperature. Using Newton's Law of Cooling for

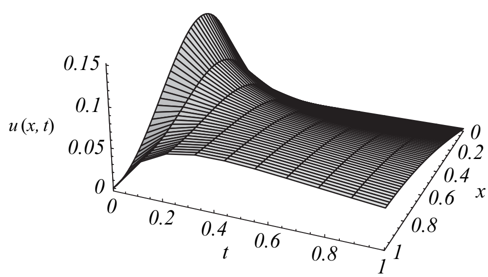


Figure 7. Plot of Mathematica numerical solution using for the heat equation generated by the code in Listing 1 with the boundary condition at $x = 1$ replaced by Derivative [0, 1][u][L, t] = = 0, which corresponds to the insulating boundary condition $u_x(1, t) = 0$.

that section of the rod we solved the nonhomogeneous heat equation for this rod and gave the students "observed" data from one temperature probe placed inside the rod but not in the uninsulated section. From these data we gave them a number of possibilities for what sections of the rod could have leaked and asked them to "sleuth it" to determine just which region was uninsulated. We also opened up the Pandora's box of inverse problems to show them how hard problems can really get, by asking them how hard it might be to determine the nature and place of insulation "failure" if we were to give them several data sets of heat in the rod over time; i.e., could they determine the insulation condition of the rod given some time, position, and temperature data.

Another application we offered students was to determine just how deep to place a root cellar so that it was coldest in the summer and warmest in the winter. This idea came from a COMAP Module [1]. The conceptual idea is that the temperature at one end of a rod of soil—the surface of the earth—varies seasonally and there is a lag as the heat moves through the soil to where the root cellar might be. With given soil parameters we asked them to locate the depth in the soil for the root cellar where that lag is exactly 6 months so that at the root cellar depth it is coldest in the summer and warmest in the winter.

CONSTRUCTION OF AN ANALYTIC SOLUTION

Here we discuss a method for deriving the standard formula for the analytic solution without an (explicit) separation of variables strategy. Instead, we build a solution out of special *elementary* solutions and work back to the Fourier series solution of the initial value problem.

Students who have had a first course in ordinary differential equations (ODE) or have some exposure to ordinary differential equations in their calculus courses are familiar with exponentials and trigonometric functions as solutions of linear, constant-coefficient differential equations. In our experience, such students can be led to conjecture a solution of the form

$$u(x,t) = e^{-\alpha t}\sin(x)$$

for the heat equation (4) with the boundary conditions $u(0, t) = u(\pi, t) = 0$. The solution can then be easily verified by substitution back into the original equation. This may seem cavalier, but students with a basic knowledge of ODEs have mostly seen solutions in terms of exponentials and trigonometric functions and, therefore, these are a natural first guess. Moreover, the physical context predisposes students to the decaying exponential in time. The choice of an oscillatory (sine) function for the spatial dependence then follows naturally from the boundary conditions.

Changing the initial condition to $u(x, 0) = \sin(2x)$, we require the students to modify their solution. The conjectured solution is naturally $u(x, t) = e^{-4t} \sin(2x)$, which can, as before, be checked by substitution back into the heat equation. The modification follows by an invocation of the chain rule and does not require physical motivation. With these two examples in mind, changing the initial condition to $u(x, 0) = \sin(nx)$, we lead them to conjecture a family of solutions of the form

$$u(x,t) = e^{-n^2 t} \sin(nx) \tag{11}$$

which, as verified by direct substitution, satisfy both the partial differential equation and the homogeneous boundary conditions. Having the constructed solutions (11), which we refer to as *elementary* solutions, we point out that, due to the linearity of the partial differential equation, we can construct a more general solution of the form

$$u(x,t) = b_1 u_1(x,t) + b_2 u_2(x,t) + \cdots + b_n u_n(x,t)$$

which has n free parameters and, for any choice of parameters, satisfies the homogeneous boundary conditions, $u(0, t) = u(\pi, t) = 0$. We note that each particular choice of parameters (i.e., numerical values for $\{b_1, \ldots, b_n\}$) generates a solution that satisfies a particular initial condition, namely,

$$u(x,0) = f(x) = b_1 \sin x + b_2 \sin 2x + \cdots + b_n \sin nx. \tag{12}$$

The question then naturally arises as to the solution of the initial value problem when the initial data is not of the form (12). For example, the initial data $f(x) = x(\pi - x)$, which is similar to that used in numerical simulations, is a polynomial with no explicit connection to sine. Because, typically, we have introduced Fourier series in a previous section of the course, it is our expectation (our wish?) that the students recognize the constants b_k in (12) as the Fourier coefficients of the initial condition, $u(x, 0) = f(x)$ on the interval $[0, \pi]$. Once this identification has been made, we can apply the usual formulae for a Fourier sine series

$$b_k = 2 \int_0^{\pi} f(x) \sin(kx)\; dx \tag{13}$$

to obtain the coefficients. For a smooth function, a plot reminds the students that a truncated series converges quickly and, thus, we can use it to obtain a good approximation of the solution for more general initial data.

The analytical solution formula provides more than a means of computing a particular solution for given data. One can use the general solution to obtain information about the behavior of the system. In the case of the heat equation with homogenous boundary conditions, the general solution is

$$u(x,t) = b_1 e^{-1^2 t}\sin(1x) + b_2 e^{-2^2 t}\sin(2x) + \ldots + b_n e^{-n^2 t}\sin(nx) + \cdots$$

where the coefficients are given by (13). From this formula, one can see that all solutions decay to zero as $t \to +\infty$. Moreover, the higher terms decay more rapidly than the lower terms so the solution becomes smoother over time. In fact, solutions with a wide range of initial data come to resemble each other more closely as the higher terms in the sum decay rapidly and the solution becomes dominated by the few lowest terms. As is our general approach, these mathematical observations can be quickly confirmed visually with plots and animations of example solutions. As we have observed, in our experience, such concrete visual feedback makes a strong impact on the students.

We note that one can take the discussion further and construct general solutions for different boundary conditions. To do so, one starts with an elementary solution of the form

$$u(x,t) = e^{-n^2 t}(a\cos nx + b\sin nx)$$

and then finds a, b, and n that satisfy the boundary conditions. In fact, the admissible values of n (and in particular the lowest value) give information about the rate of loss of thermal energy associated with a given set of boundary conditions.

THE WAVE EQUATION AND MAKING SOUNDS

Using elementary physical principles and a free-body diagram of the forces on a plug of mass on a string, we can obtain the wave equation as a model of its transverse oscillations [5], namely,

$$u_{tt}(x,t) = c^2 u_{xx}(x,t)$$

where $u(x, t)$ is the transverse displacement at a distance x from one end of the string at time t, $c = \sqrt{\frac{T}{\rho}}$, where T is the tension in the string, and ρ is the (linear) density of the string material.

Initial (IC) and boundary conditions (BC) are necessary.

$$\text{IC}: \quad u(x,0) = f(x) \text{ and } u_t(x,0) = 0 \quad \text{BC}: \quad u(0,t) = 0 \text{ and } u(L,t) = 0$$

where $f(x)$ is the initial displacement of the string. When we teach the wave equation it is usually after the heat equation and we usually forego the student's developing the physical derivation. Instead, we sketch it in class and move directly to solutions.

Mathematica numerically solves this partial differential equation with various initial conditions, usually keeping the ends fixed as with a musical instrument string that is bowed, pulled and released, or struck. We discuss various initial conditions, e.g., triangular shape, which is tantamount to pulling the string up and releasing it, and the theoretical initial conditions referred to as *modes*, i.e., going from half sine cycle over the length of the string, $u(x,0) = f(x) = \sin\left(\frac{\pi x}{L}\right)$ (first mode), to full cycle over the length of the string, $u(x,0) = f(x) = \sin\left(\frac{2\pi x}{L}\right)$ (second mode). We give examples of solutions for these two modes (other modes are easily replicated) in Figure 8.

Here we tell the students that the first mode frequency of the wave or oscillation is

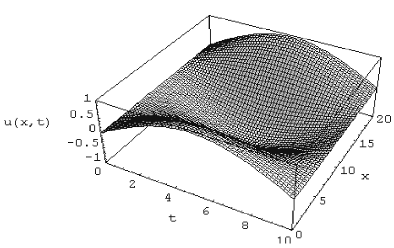
$$f_1 = \frac{1}{2L}\sqrt{\frac{T}{\rho}}. \tag{14}$$

At this point we have not done the building of the analytic solutions involving the eigenfunctions and necessary Fourier series preliminaries; rather, we just tell them about the modes and have Mathematica show them the wave motion in three dimensions and in two-dimensional animations.

We now attempt to tune the string to get a first mode frequency for the string. We consider the following string with ρ = 8 g/cm and L = 10 cm. T, in dynes, is unknown and we wish to determine T so that f_1 is 440-Hz frequency for our first mode of the string's vibration. 440 Hz is the frequency for the A above middle C on the piano and we often digress to talk about the tuning of piano strings by changing the tension with a tuning wrench.

We use Mathematica's Solve routine to solve for and grab T_n, the tension we need to apply to obtain a vibrating frequency of 440 Hz.

```
Tn = T/.Solve[440 == 1/(2*10) Sqrt[T/8], T][[1]]
```



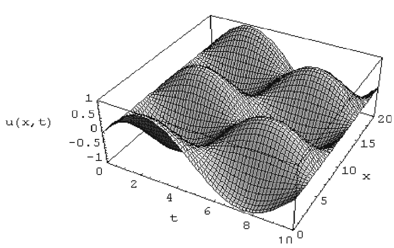


Figure 8. First two modes for vibrating string: first mode (a) where $u(x,0) = f(x) = \sin\left(\frac{\pi x}{L}\right)$ and second mode (b) where $u(x,0) = f(x) = \sin\left(\frac{2\pi x}{L}\right)$.

From this we obtain $T_n = 619{,}520{,}000$ dynes and thus $c = \sqrt{\frac{T}{\rho}} = \sqrt{\frac{619520000}{8}} = 8800$ m/s for our wave equation model. Now we offer up the initial conditions, first the initial position of the string – the first mode:

```
ic[x_] = Sin[Pi x/L]
```

and then an initial velocity at $t = 0$. In the NDSolve command we set the initial velocity to zero:

```
iv[x_] = D[y[x,t],t]/.{t->0}
```

Finally, we use NDSolve to generate a numerical solution of the wave equation and grab the solution, $h_1(x, t)$. Notice the command MaxSteps->100000, which allows a tenfold increase in the maximum number of steps Mathematica uses to get us to the value of $t = 1$.

```
h1[x_,t_]=
  y[x,t]/.NDSolve[
 {D[y[x,t],t, t] == c^2*D[y[x,t],x,x],
  y[x,0]==ic[x],iv[x] == 0,
  y[0,t]==0, y[L,t] == 0},
  y,{x,0,L},{t,0,1},
 MaxSteps->100000
][[1]];
```

If we "watch" the spot on the middle of the string ($x = L/2$) as it has greatest amplitude we can study the following function:

```
p[t_] = h1[L/2,t].
```

Our theory says that $p(t)$ oscillates with a frequency of 440 Hz. This is the frequency of the A above middle C on the piano. We can plot (Figure 9) the oscillating function $p(t)$ and the standard 440 Hz signal $\sin(440\ 2\pi\ t)$ and we obtain the following. We see that they are slightly out of phase but clearly of the same frequency.

With Mathematica we can play the audible two frequency sounds and *hear* if our solution has 440-Hz frequency. (Simply replace the function Plot with the function Play). When we play the two signals they sound the same. Indeed, when we add the two signals, if they were a tad off we would get beats, and no beats appear in our case. Thus, our numerical solution from Mathematica gets it right and we have determined the tension (or tuning) necessary to tune a vibrating string to exactly a frequency of 440 Hz. We have used this sound feature of Mathematica on a number of occasions in order to illustrate such notions as resonance and beats in the study of

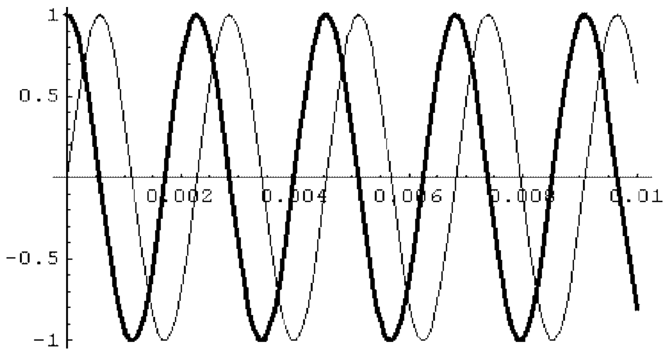


Figure 9. Plot of (1-thin) solution to the wave equation $u_{tt}(x, t) = c^2 u_{xx}(x, t)$ for a vibrating string with initial conditions (first mode); position $-u(x,0) = \sin\left(\frac{\pi x}{L}\right)$ and velocity $- u_t(x, 0) = 0$; and boundary conditions (fixed end points) $u(0, t) = 0$ and $u(L, t) = 0$ and (2 – thick) the pure 440 Hz signal, $g(t) = \sin(440\ 2\pi t)$, shown out of phase to emphasize the common frequency.

second-order, nonhomogeneous, linear, constant coefficient ordinary differential equations and Fourier analysis of sounds.

CONCLUSION

We have offered ways to incorporate mathematical modeling in the study of partial differential equations. The solution of the resulting partial differential equation is accomplished in three different ways (i) discretization using a spreadsheet, (ii) numerical solver of Mathematica, and (iii) equation building from basic components. In all cases there is attention to the phenomenon being modeled and students can address "what if?" issues with the solutions.

Specifically, we have presented an approach for studying two partial differential equations in one space and one time dimension, (i) the heat equation and (ii) the wave equation. In both cases we start with a physical situation and model by using elementary physical principles. In the case of the heat equation we build a forward difference model to numerically render a solution in a spreadsheet. In both cases we turn to Mathematica's NDSolve command to obtain a numerical solution and study our solutions through graphical representation, animation, and, in the case of the wave equation, sound. Moreover, we offer an analytic approach different from the separation of variable/boundary value approach often taken. We have found that students enjoy seeing a mathematical model that demonstrates physical notions. The graphical and audio feedback makes both numerical and analytical solution more concrete for the students and gives them a basis to test and develop their intuition.

ACKNOWLEDGMENTS

The authors acknowledge improvements through discussions and coteaching with Mark Gorak, Bart Stewart, Robbie Williams, Sam Wright, Keith Landry, James Hoskin, Chris Conley, and Mark Evans. Figure 2 was created by Bart Stewart for a summary of the modeling effort for our students. We appreciate the careful reads of our colleagues Leigh Noble and Brian Souhan.

REFERENCES

1. Berresford, G. C. 1981. *Differential Equations and Root Cellars*. Bedford MA: COMAP, Inc.
2. Finizio, N. and C. Ladas. 1982. *An Introduction to Differential Equation with Difference Equations, Fourier Series, and Partial Differential Equations*. Belmont CA: Wadsworth Publishing Co.
3. Guenther, R. B. and J. W. Lee. 1996. *Partial Differential Equations of Mathematical Physics and Integral Equations*. Mineola NY: Dover Publications.
4. Haberman, R. 1987. *Elementary Applied Partial Differential Equations with Fourier Series and Boundary Value Problems*. Englewood Cliffs, NJ: Prentice-Hall.
5. Kreyszig, E. 1999. *Advanced Engineering Mathematics, 8th ed*. New York: John Wiley & Sons.
6. Petter Langtangen, H. 2003. *Computational Partial Differential Equations: Numerical Methods and Diffpack Programming. 2nd ed*. New York: Springer.
7. Winkel, B. 1991. Graphical feedback for teaching Fourier series. *Collegiate Microcomputer*. 9(1): 55–58.
8. Winkel, B. Accepted. Fourier series: Optimization opportunity. *International Journal of Mathematical Education in Science and Technology*.

BIOGRAPHICAL SKETCHES

Joe Myers enjoys teaching and doing applied mathematics at the United States Military Academy. He has taught and directed almost everything during his two decades there, including the freshman calculus program, the sophomore multivariable calculus program, the electives program, the research program, and 23 different courses.

David Trubatch taught with Joe Myers and Brian Winkel at the United States Military Academy before joining the Department of Mathematical Sciences at Montclair State University. His research efforts are focused on differential equations and nonlinear dynamics, including especially nonlinear waves and solitons.

Brian Winkel teaches mathematics and mentors faculty at the United States Military Academy. He is the founder and editor of the journal *PRIMUS Problems, Resources, and Issues in Mathematics Undergraduate Studies* and also founded and is editor emeritus of the journal *Cryptologia*.