

TEACHING PDE-BASED MODELING TO ChE UNDERGRADUATES

Overcoming Conceptual and Computational Barriers

KARSTEN E. THOMPSON

Louisiana State University • Baton Rouge, LA 70803

Introducing partial differential equations (PDEs) in the undergraduate engineering curriculum can be frustrating for both students and instructors. Many students gain a dislike of differential equations before the core engineering curriculum even begins. As engineering instructors, we then face a number of challenging problems. We must help the students overcome conceptual barriers associated with the math and help them envision the physical phenomena being described. Additionally, we must devise models for which we can obtain solutions in limited timeframes (*i.e.*, class time or homework time). This latter constraint is imposed by computational barriers, which often restrict us to overly simplistic problems that have limited engineering relevance. Although these computational issues have been a stumbling block in the past, modern numerical packages prove to be very accessible, even to the undergraduate student, and have been shown to improve learning in a number of ways.^[e.g.,1]

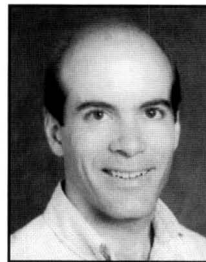
This paper describes an instructional framework that was used for incorporating computational tools into a relatively short section on PDEs during an undergraduate modeling course. The rationale is that by allowing the students to jump headlong into the solution of real engineering problems, the emphasis in the classroom can change. Attention can be diverted away from arduous mathematical details (for the moment) and focused on broader conceptual issues such as the general behavior of classes of equations, or using a model for design considerations. This approach has a number of advantages. The students' concurrent, hands-on solution of problems is a powerful method for illustrating fundamentals (that otherwise seem abstract). More time can be committed to model development and general behavior (which remain with students longer than the details of solution techniques). And, the ability to solve real engineering problems illustrates to the students the true power of mathematical modeling.

A number of computational packages are available and appropriate for undergraduate education. Mathcad, Maple, Mathematica, MATLAB, and Polymath are all common in chemical engineering education.^[1-4] Fluent is especially effective for CFD applications.^[5] We focus on MATLAB in this paper, largely because of its unique PDE Toolbox. At LSU, MATLAB is provided to the students via the department's PC network. (An academic unit can provide MATLAB on a number of PCs without a large investment by purchasing a classroom kit.) Additionally, Mathworks has recently released a student version of MATLAB (which is the professional version plus popular toolkits) that helps students who wish to work at home.

In the remainder of this paper we will discuss the context for this approach and one possible strategy for breaking down conceptual barriers in the classroom. At the end, three example problems will be solved using MATLAB, illustrating the type of modeling exercises that can be assigned to complement conceptual discussions in the classroom.

CONTEXT

The ideas and examples described here were developed as part of a senior-level math-modeling course at LSU, which is described below. When beginning the PDE section of the course, it was helpful to consider the contexts in which our



Karsten E. Thompson is an assistant professor of chemical engineering at Louisiana State University, where he has been since 1996. He received his BS degree from the University of Colorado and his PhD from the University of Michigan. He teaches courses in numerical methods, math modeling, and transport phenomena. His research interests include flow in porous media, numerical methods, and fluid mechanics.

© Copyright ChE Division of ASEE 2000

Chemical Engineering Education

students typically are introduced to PDEs. Unfortunately, these encounters are infrequent and generally presented as asides to the main material.

Consider three of the most likely places that ChE undergraduates run into PDEs. First, they may arise at the end of an introductory numerical methods course. But using Chapra and Canale^[6] as a guide, we see that PDEs are relegated to Chapter 29, suggesting that the topic probably receives a quick overview near the end of this type of class. PDEs are usually presented again in the undergraduate fluids course in the form of the Navier-Stokes equations, but the emphasis here is placed on the flow phenomena rather than the mathematics (rightly so), and the solved examples are nearly always one-dimensional problems that reduce to ODEs. Hence, for chemical engineering students, the lasting impression of PDEs usually comes from dealing with the unsteady heat equation. While the equation and its application are fairly easy to grasp, the mathematics are not; the most common introductory solution is for a semi-infinite domain, which calls for abstract boundary conditions, a similarity transform, and an error-function solution. (Surely, the frequent appearance of *erf* on comical student-organization T-shirts should tell us something!)

While experiencing PDEs in a variety of contexts emphasizes their importance, it likewise seems to leave the students somewhat unsettled and without a firm foundation for the subject. These considerations helped formulate an approach that was used when given the luxury of spending three or four weeks on the topic of PDEs with senior-level chemical engineers. At LSU, the class in which this occurs is “Development of Mathematical Models.” It affords a number of unique opportunities for the instructor. Being a math course among other more popular electives, it attracts some of the better undergraduates along with a few Masters students from our graduate program and from industry, and these students have more-or-less completed the ‘principles’ classes. Also, beyond the class’s strong modeling emphasis, the topical coverage is left largely to the instructor’s discretion (see Rice and Do^[7] for a template of the course offered at LSU). Finally, at the point in the semester when PDEs are introduced, the students have themselves derived a number of PDEs governing transport and reaction engineering problems that subsequently were reduced to simpler form. Hence, their foundation for model development is strong, and they can clearly see the need to move beyond algebraic equations and ODEs in order to make full use of their models.

APPROACH

The educational objective is to allow the students to solve complex PDEs of real engineering interest, without sacrificing coverage of fundamental mathematical behavior. To realize this goal, an approach is used whereby classroom coverage includes topics such as the origin of the equations, visualization of the solution space, qualitative behavior, and ties between the physical phenomena and mathematics. These topics are potential conceptual barriers, and overcoming them can make the terminology and the detailed mathematics less intimidating. At the same time, to prevent excessive classroom time being devoted to numerical techniques or specialized software, the students are left mostly on their own to pursue numerical solutions to homework or modeling projects. This approach works well (given good numerical tools) because the solutions that the students have themselves worked out are highly effective illustrations of conceptual topics. In the past, it would have proved less feasible because, until the instructor covered solution techniques, the students lacked the tools to fully explore the mathematical models with which they were working.

The intent is not to de-emphasize the crucial subject of solving PDEs. Rather, these issues are postponed until the foundation is stronger. If implemented properly, this approach shares many positive attributes of ‘just-in-time learning’ employed by Finlayson.^[8] modeling projects evolve so that just as students identify the need for new mathematical tools, the relevant subjects are addressed. The benefit of incorporating software such as MATLAB is that topical coverage in the classroom can remain fundamental without slowing the students’ progress toward quantitative solutions.

The ideas and examples described here were developed as part of a senior-level math-modeling course at LSU, . . . When beginning the PDE section of the course, it was helpful to consider the contexts in which our students typically are introduced to PDEs. Unfortunately, these encounters are infrequent and generally presented as asides to the main material.

CONCEPTUAL BARRIERS

Conceptual barriers are never clear-cut, and they of course depend on background, learning style, and ability. But we will attempt to generalize a set of hurdles that stand between more easily conceptualized physical behavior and the corresponding mathematics. An example is the categorization of PDEs; the mathematical definitions of parabolic versus hyperbolic PDEs probably seem abstract at first, but explaining the analogy to the synonymous heat and wave equations makes this classification more tangible.

This development of strong ties between analogous physical and mathematical behavior provides a basis for breaking down conceptual barriers. Emphasis is placed on the behavior of *general classes* of PDEs, with concrete examples used to illustrate these ties. It is hoped that the foundation and comfort level that can be achieved by this approach offsets the 'risk' of sending the students headlong into numerical solutions (without much knowledge of the associated techniques).

No single list of conceptual barriers is comprehensive. Table 1 shows a list of topics that are addressed sequentially by the author during a three-week section covering PDEs. The first couple of points are general, but the latter part of the list applies to second-order PDEs, for two reasons. First, many second-order equations can be packaged neatly into elliptic, hyperbolic, and parabolic categories, which aids in generalizing behavior. Second, these are the types of equations amenable to solution in MATLAB's PDE Toolbox, which was an essential part of the approach.

The in-class overview of second-order PDEs was taken largely from Crandall,^[9] who does an excellent job of tying qualitative behavior of the equations to quantitative mathematics (numerical and analytic). The approach promotes picturing a PDE as a family of surfaces, the correct surface being pinned down by the appropriate boundary and initial conditions. Crandall explains the difference between equilibrium problems and propagation problems, which ties in nicely to a discussion of the characteristic curves for parabolic and hyperbolic equations.

COMPUTATIONAL BARRIERS

Effective use of a math model requires, of course, a solution. In the past, a significant time investment was required to introduce analytic

solution techniques and the problems were more often than not restricted to one spatial dimension. From a numerical perspective, innovative ideas have been presented for programming solutions to PDEs at an introductory level,^[10,11] but these too are dimensionally restrictive. In contrast, modern numerical software gives the student more flexibility with respect to the type of problems that can be solved and the way in which they can be explored.

Software such as MATLAB's PDE Toolbox allows two-dimensional problems of arbitrary geometry to be set up and solved in a matter of minutes. In MATLAB, the system geometry is entered using a graphical user interface (GUI) that strongly resembles familiar drawing programs. Equations and boundary conditions are chosen from radio-button menus, and clicking with the mouse results in mesh discretization, solution by the finite element method, and a wide range of three-dimensional color graphical output. Numerical output is only slightly harder to manipulate, requiring a review of MATLAB's built-in 'pde-tool' scripts. The PDE toolbox can be used at an introductory level without extensive knowledge of the basic MATLAB software. Hence, the time spent in familiarizing students with the software can be kept relatively small. In 1997, the PDE toolbox was introduced to the students during a single-

TABLE 1
Synopsis of Conceptual Topics Discussed During Classtime
During the PDE Section of a Modeling Course.

<i>Topic used to address conceptual barrier</i>	<i>Synopsis of the classroom discussion</i>
Independent and dependent variables	Use a 'familiar' equation to emphasize independent/dependent and how they define a PDE.
Picturing the solution surface	An ODE describes a family of curves; one or more boundary conditions pin down the curve of interest. Similarly, many PDEs can be pictured as a surface, pinned at the edges.
Second-order PDEs	These are categorized as elliptic, parabolic, or hyperbolic. It helps to understand physical behavior associated with each in the form of Poisson's equation, the heat equation, and the wave equation.
Equilibrium versus propagation	Elliptic equations describe equilibrium or 'jury' problems ^[9] —the boundaries (which are fixed on all sides) wholly dictate the shape of the interior. Propagation problems have an open boundary. The surface evolves with time, pinned on the sides by boundary conditions and at the front by initial conditions.
Characteristic lines	Characteristic lines are sudden changes in the slope of the surface (like creases), brought about by changes in boundary conditions. Parabolic equations have one characteristic line; changes at the boundary are propagated instantly but weakly into the interior. Hyperbolic equations have two characteristic lines. Changes at the boundary are propagated into the interior at full strength but at a finite speed.
Finite difference methods	There are analogous characteristic lines for the finite difference method. These partly dictate step size in propagation problems.
Stability and numerical diffusion	When using numerical solutions, we must be wary of errors inherent to the solution technique. Many of these stability issues relate to the form of the governing equation.

class session held in the department computer lab. In 1999, it was introduced via a step-by-step instruction set handed out to the students, thus requiring no class time.

The use of software should not de-emphasize the importance of teaching analytic and numerical techniques for solving PDEs. Instead it should allow us to embrace the introduction of PDEs at the undergraduate level (if their use enhances the fundamentals being taught) and encourage exploration and critical thinking early on.

Three examples are given below. These were chosen to illustrate a number of points. First, they are indicative of the

types of problems easily solved by MATLAB (most importantly, those with arbitrarily complex boundary geometries). Second, one each of an elliptic, parabolic, and hyperbolic equation are shown. Third, and most important, they typify problems where the student clearly understands the engineering relevance and the manner in which their models can be used in design. This last aspect becomes especially effective when a model can be tied to an experiment, as is shown in Example 1, or, for instance, in reference 12. The MATLAB scripts for these examples can be found on the web at <www.che.lsu.edu/faculty/thompson/education.htm>

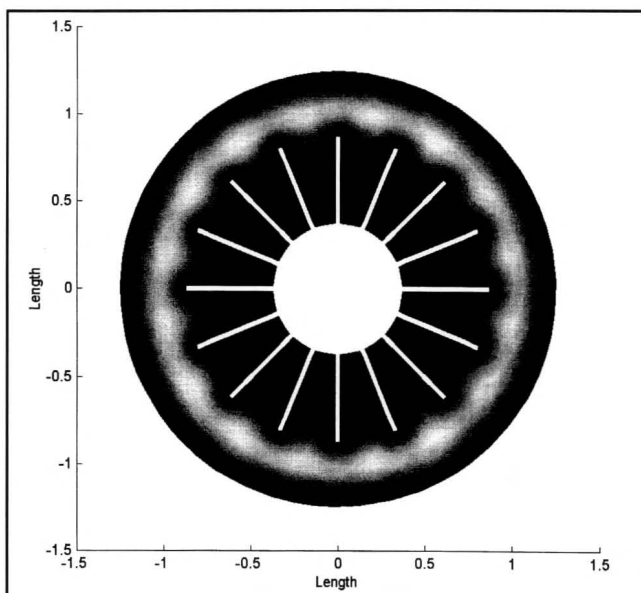


Figure 1. Velocity map in entire heat-exchanger domain taken from undergraduate students' solution.

domain (boundaries along the centerlines of two neighboring fins) or 1/32 of the domain (one boundary along a fin and one along a fluid line of symmetry). We show the former approach below.

For unidirectional flow, the Navier-Stokes equations reduce to^[13]

$$\rho \frac{\partial u}{\partial t} = G(t) + \mu \nabla_2^2 u \quad (1)$$

where u is the velocity in the direction of flow, $G(t)$ is the pressure gradient in this direction, and ∇_2^2 is the Laplacian for the two directions orthogonal to flow. Hence, for a steady flow, G is constant and the equation re-

duces to Poisson's equation

$$\nabla_2^2 u = -G/\mu \quad (2)$$

Zero-velocity boundary conditions are used along all surfaces of the heat exchanger. Depending on how the FEM domain is chosen, lines of symmetry are likely to arise, in which case symmetry boundary conditions ($\mathbf{n} \cdot \nabla u = 0$) are also used. In MATLAB, the elliptic equation is written as

$$-\text{div}[c * \text{grad}(u)] + a * u = f \quad (3)$$

where 'div' and 'grad' are the divergence and gradient operators, respectively. Hence, one would specify $c=1$, $a=0$, and $f=G/\mu$ to perform the calculation. The solution using MATLAB's PDE Toolbox involves four steps:

1. Map the domain using the GUI. (The geometry can be drawn crudely using the mouse, and then refined by double-clicking on the various polygons to type in precise vertex positions.)
2. Select the governing equation and boundary conditions. The GUI contains a radio-button interface that allows the user to specify the type of equations and boundary conditions along with values of parameters.
3. Solve the problem. After step two, the solution consists of clicking two buttons: one to generate and refine the FEM mesh and the second to solve the problem. A wide range of graphical output is available.
4. Quantitative analysis. This last step requires slightly more user experience since the command-line interface must be used. For instance, values of velocity at each node in the mesh are contained in an array that can be exported to the MATLAB workspace. There are a series of 'pde-tool' commands that can then be used to perform interpolation, integration, etc. Integration is used to calculate volumetric flowrate as a function of the pressure gradient that was

Example 1
Steady Laminar Flow in a Finned Heat Exchanger

For parallel laminar flow through a duct, the Navier-Stokes equations reduce to Poisson's equation (a scalar equation since there is only one velocity component). This example is for flow in the annulus of a heat exchanger in LSU's ChE measurements laboratory. This example was of particular interest to students in the math modeling class because many of them had used this apparatus for a pressure-drop-versus-flowrate experiment in which they calculated the friction factor for the annular region. Additionally, the complex geometry in the annulus makes the problem an excellent candidate for solution using MATLAB.

The annular space in the heat exchanger is contained between $r = 1.05$ inches (OD of inner pipe) and $r = 2.469$ (ID of outer pipe). It contains 16 symmetrically placed fins of width 1/32 inch and length 1/2 inch, as shown in Figure 1. Symmetry allows the solution to be performed in either 1/16 of the

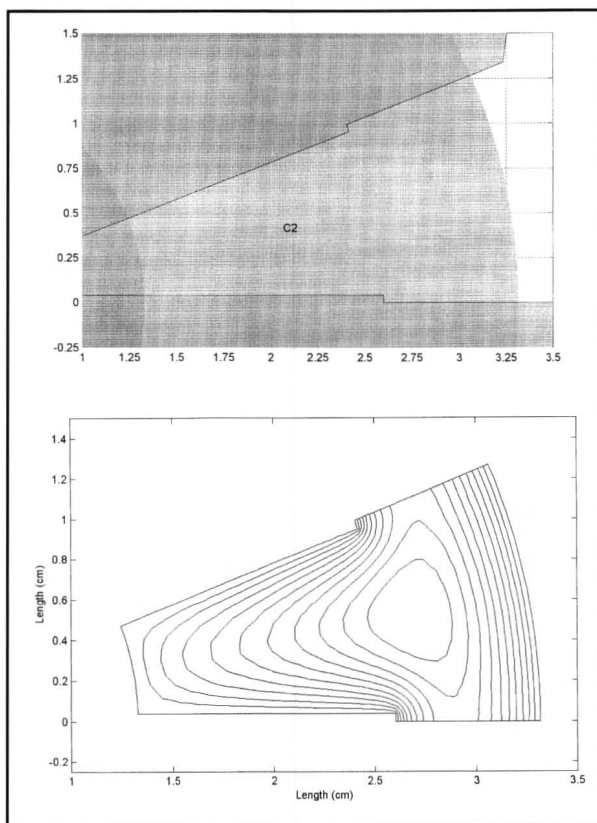


Figure 2. Geometry of 1/16 of flow region in heat exchanger (top); z-direction (perpendicular to page) velocity contours (bottom).

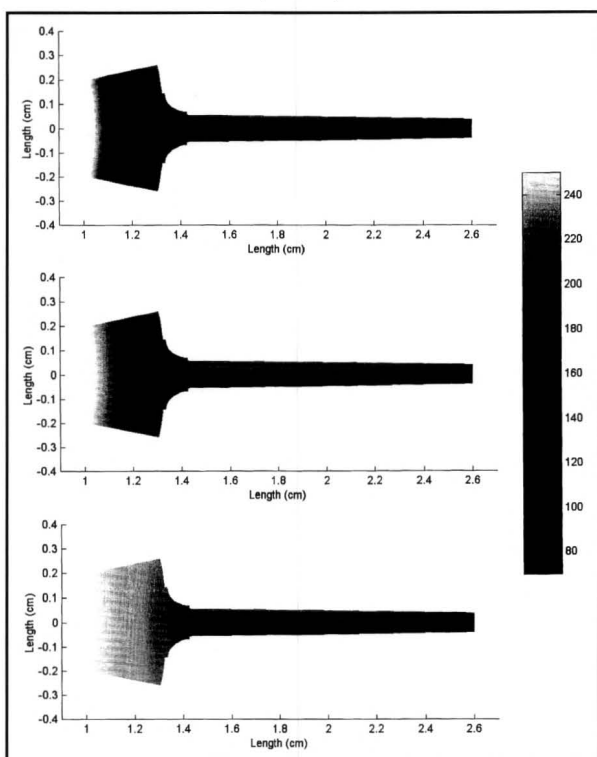


Figure 3. Snapshots of temperature profiles during transient heating ($t=7$ sec, $t=2.4$ min, and $t=12$ min).

imposed during specification of the PDE.

Figure 2 shows how the annulus geometry (1/16 of the entire domain) is defined using various simple geometric shapes and shows contours of the z-direction velocity (*i.e.*, velocity is perpendicular to the page). The contours have not been assigned numerical values in this figure because the velocity depends on one's specific choice of viscosity and pressure gradient. Their shape remains fixed, however.

Integration of the velocity profile to determine flowrate gives a fanning friction factor $f=19.5/Re$, whereas students typically obtain values between $f=20/Re$ and $f=25/Re$ in the experiment (using an effective area approach). When quantitative analysis is used, one can introduce the students to issues of numerical accuracy. Grid refinement is trivial in MATLAB, requiring only the click of a button for successive refinements of the FEM mesh. Using various levels of grid refinement in this example (the standard mesh followed by two successive refinements), one obtains the following values for the friction factor: $20.12/Re$, $19.65/Re$, and $19.52/Re$.

Returning to Figure 1, one can see the velocity profile for the entire annular region, where the lighter shading indicates higher velocities. While numerical solution over the entire annular region is less efficient than breaking it along lines of symmetry, the resulting graphic is more appealing than Figure 2. Figure 1 was taken directly from two students' homework;^[14] they were able to work the problem after only a short tour of MATLAB during one of the class periods. (The students used inches rather than cm, causing the discrepancy of scale with Figure 2.)

Example 2 Transient and Steady Heat Transfer in a Finned Heat Exchanger

While the heat exchanger described in Example 1 is not used for heat-transfer experiments at LSU, the concept of heat transfer from a fin is both important and readily amenable to visualization.

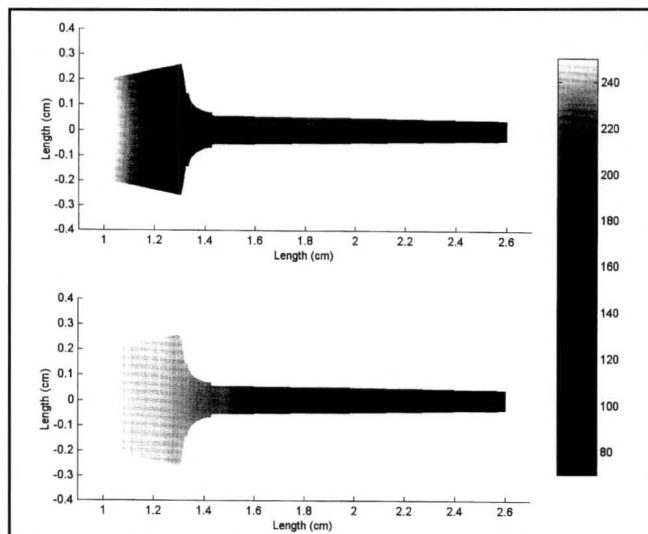


Figure 4. Steady temperature profiles for high (top) and low (bottom) fin-side heat transfer coefficients.

The example shown here can illustrate the unsteady evolution of a temperature profile as well as the steady-state temperature profile for various heat-transfer coefficients. The transient problem underscores the behavior of parabolic PDEs, while the steady problem helps to illustrate engineering fundamentals: the conditions under which fins are useful for increasing heat-transfer efficiency.

Parameters are taken from Example 27-3 in Bennett and Myers,^[15] and the fin dimensions are based roughly on the elliptic example above. The geometry is made somewhat more complex to illustrate the versatility of MATLAB's GUI.

Figure 3 shows the fin at $t=7$ sec, $t=2.4$ min, and $t=12$ min. The shading shows the evolution of the temperature profile in time (an effect that is much more dramatic in color). Figure 4 shows two steady-state profiles, for a high external heat-transfer coefficient [1500 Btu/(hr·ft·°F)] and a low external heat-transfer coefficient [1 Btu/(hr·ft·°F)]. The graphics shown here help students conceptualize the conditions under which fins can significantly increase heat transfer.

We make two final points. First, the evolution of the temperature profile (especially in color) is helpful for understanding the behavior of a parabolic PDE: a change in the boundary condition has immediate but weak influence throughout the fin, and the temperature evolution is smooth. Second, one can easily envision numerous exercises that could be performed to illustrate important behavior. For instance, a spatially varying heat-transfer coefficient (discussed in Bennett and Myers) is impossible to impose for even simple analytic solutions, but can easily be incorporated into the MATLAB solution.

Example 3
Wave Propagation in a Heterogeneous Material

Applications of the basic wave equations are less frequent in chemical engineering. While strong convection effects

can produce wave-like behavior, MATLAB (to the author's knowledge) is not equipped to handle convective transport. One can, of course, simulate vibration problems or certain problems involving sound waves. Instead, the example shown here is a highly simplified illustration of seismic exploration. It was chosen because of its intuitive appeal to an engineer of any discipline.

Artificial seismic waves are used in oil exploration or geophysical analysis to map subsurface structure. The wave source may be either on the surface or lowered into a well, and the responses to the wave at various detector locations are interpreted to give the mapping. On a simplistic level, the propagation of pressure waves in the ground is described in the wave equation^[16]

$$\nabla^2 p - \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} = 0$$

where c is the wave velocity, dependent primarily on the material properties. A time-dependent pressure must be defined at the wave source. Along reflective boundaries of the domain, one specifies $\mathbf{n} \cdot \nabla p = 0$.

Figure 5 shows the geometry used for this simple example. The interesting features are the slope of the lower boundary (which could be interpreted as a geologic bedding plane) and the inclusion of a material heterogeneity at the lower right. These two features make the response more interesting, but preclude solution by analytical means.

To solve the problem, a pressure spike was induced (via a rapidly decaying exponential function) along the top boundary at $t=0$. Figure 6 is a qualitative illustration of the resulting wave's behavior. The lighter shading (which represents the traveling high-pressure front) propagates downward, reflects off of the bottom boundary, and then returns to the surface (where it would be detected). Although length and time scales are not included since the ex-

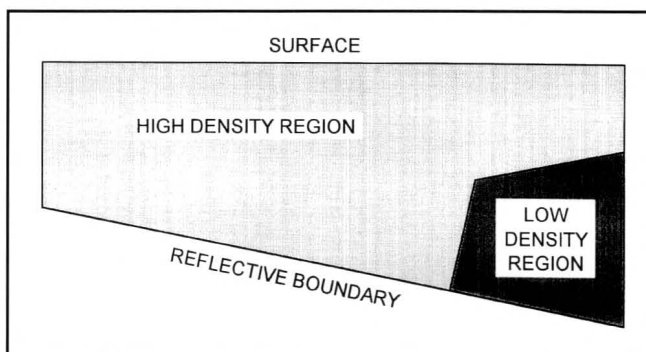


Figure 5. Geometry used for wave propagation example.

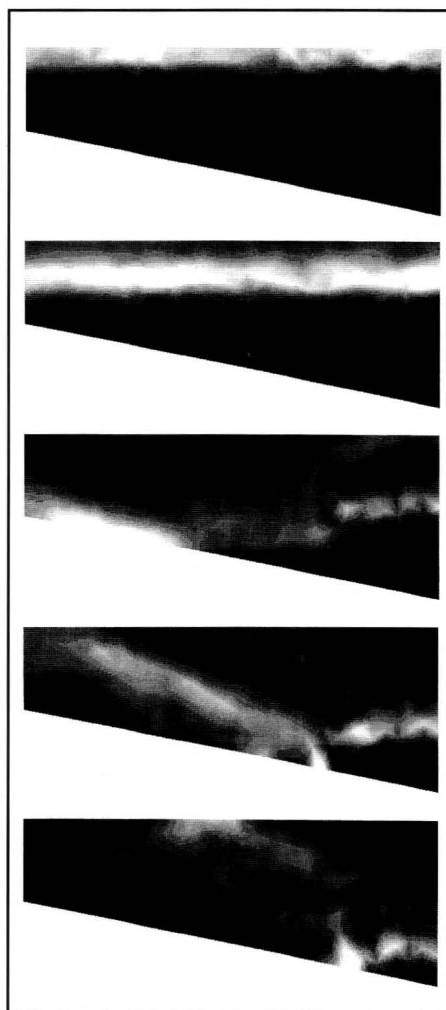


Figure 6. Wave propagation and reflection in a heterogeneous domain. White corresponds to higher-pressure transients.

ample is qualitative, one can see the influence of the lower-density inclusion where the wave gets held up, and one can envision how the surface response allows a map of the subsurface to be generated. From a mathematical viewpoint, this example helps illustrate typical hyperbolic behavior: the response of the interior to a change at the boundary is delayed, but then felt at full strength once the wave reaches a given point. Using this example in the classroom, an otherwise dry discussion of characteristic lines for a hyperbolic equation can become more captivating.

CONCLUSIONS

The use of PDEs in the undergraduate curriculum often has mixed results: Important topics cannot be modeled without PDEs. On the other hand, the simplicity of solution domains for analytic problems often makes for abstract relationships to real engineering problems, and the mathematical details of an analytic solution can distance students from the original objectives.

This paper presents effective uses of modern numerical software for solving real engineering problems at the undergraduate level, which is an increasingly popular approach among chemical engineering educators. The quick learning curve for certain numerical software allows students to begin exploring a model's behavior almost immediately. Classroom time can then be used to break down conceptual barriers associated with PDEs. It is hoped that this approach lays a better foundation and better prepares students for later material on solution techniques, either analytical or numerical.

ACKNOWLEDGEMENTS

The author would like to acknowledge Charles Varnado for providing the numerical code used to generate Figure 1, and to acknowledge the anonymous reviewers for very insightful comments on the original manuscript.

REFERENCES

1. Harb, J.N., A. Jones, R.L. Rowley, and W.V. Wilding, "Use of Computational Tools in Engineering Education," *Chem. Eng. Ed.*, **30**(3), 145 (1997)
2. Mackenzie, J.G., and M. Allen, "Mathematical Power Tools: Maple, Mathematica, MATLAB, and Excel," *Chem. Eng. Ed.*, **32**(2), 156 (1998)
3. Taylor, R., and K. Atherley, "Chemical Engineering with Maple," *Chem. Eng. Ed.*, **29**(1), 56 (1995)
4. Cutlip, M.B., and M. Shacham, *Problem Solving in Chemical Engineering with Numerical Methods*, Prentice Hall (1999)
5. Sinclair, J.L., "CFD Case Studies in Fluid-Particle Flow," *Chem. Eng. Ed.*, **32**(2), 108 (1998)
6. Chapra, S.C., and R.P. Canale, *Numerical Methods for Engineers*, 3rd ed., McGraw-Hill (1998)
7. Rice, R.G., and D.D. Do, *Applied Mathematics and Modeling for Chemical Engineers*, Wiley (1995)
8. Finlayson, B.A., "Problem-Centered Course in Using Multimedia," paper 3513-02 presented at the ASEE Annual Conference and Exposition, Washington, DC (1996)

9. Crandall, S.H., *Engineering Analysis: A Survey of Numerical Procedures*, McGraw-Hill (1956)
10. Wiggins, E.G., "Computational Fluid Dynamics on a Spreadsheet," *Comp. in Ed. J.*, **7**(2), 7 (1997)
11. Cutlip, M.B., and M. Shacham, "The Numerical Method of Lines for Partial Differential Equations," *CACHE News*, **47**, 18, Fall (1998)
12. Anklam, M.R., R.K. Prud'homme, B.A. Finlayson, "Ion Exchange Chromatography Laboratory: Experimentation and Numerical Modeling," *Chem. Eng. Ed.*, **31**(1), 26 (1997)
13. Leal, L.G., *Laminar Flow and Convective Transport Processes: Scaling Principles and Asymptotic Analysis*, Butterworth-Heinemann (1992)
14. Varnado, C., and K. Loo, Math Modeling Project for ChE 4296, LSU (1997)
15. Bennett, C.O., and J.E. Myers, *Momentum, Heat, and Mass Transfer*, 3rd ed., McGraw-Hill (1982)
16. Berkhout, A.J., *Applied Seismic Wave Theory*, Elsevier (1987)

□

RANDOM THOUGHTS

Continued from page 145.

The more types of assessment data collected for a specific component (column of the matrix), the more reliable, valid, and fair the evaluation of that component. For explanatory notes and literature citations on the different assessment tools, see Reference 3.

How might the scholarship of teaching be included in tenure and promotion decisions?

Many academic institutions have begun to acknowledge the scholarship of teaching as a valid component of tenure and promotion (T/P) applications. An approach being taken by several of these institutions is to allow faculty members to allocate variable percentages of their total effort to teaching, research, and service, with minimum percentages being specified for each area. If more than a certain percentage is allocated to teaching, educational scholarship must be included in the faculty member's activities and a teaching portfolio containing a subset of the items in Table 1 must be included in the T/P dossier. A review committee assigns separate numerical performance ratings to each of the three areas and weights the ratings by the specified percentages to calculate a composite rating, which provides the basis for the decision on tenure or promotion.

For ratings of the scholarship of teaching to be reliable and valid, the evaluating department should take the following steps:

- *Formulate and announce an assessment and evaluation plan.* Decide which items listed in Table 1 will be collected in the teaching portfolio, taking into account both institutional guidelines and considerations specific to the department. Choose a system to rate each of the items in the portfolio (e.g., rate each item on a scale from 0 to 10), weighting factors for each item, and weighted scores that serve as criteria for ad-