

Visualization of PDE Solutions Using Implicit Methods and MATLAB

Raymond G. Jacquot, Jerry C. Hamann
 Electrical Engineering Department
 University of Wyoming

Abstract

In many engineering curricula the formal techniques of the solution of partial differential equations are not studied, however, all such curricula examine problems described by such equations. An alternative to studying the formal solutions is to examine the solutions of such problems by numerical methods. The authors have found MATLAB particularly attractive for the solution of such problems because of the ease with which linear algebra problems are solved and displayed in that computing environment.

Introduction

Physical systems governed by partial differential equations have become a part of most undergraduate engineering curricula although many of these curricula do not require formal mathematical training in such topics. Typical subject matter with such content includes vibrations, acoustics, transport phenomena and electromagnetic theory. Students without formal mathematical training in partial differential equations are almost helpless to understand the solution to such a problems in their engineering courses. An alternative to teaching analytical techniques of mathematical physics is to explore the solution to such problems in an approximate manner. Two ways to approach these approximate solutions are via numerical analysis or by the weighted residual approaches of Galerkin and Ritz. The former method employing MATLAB is the topic of this paper while the latter will be explored in a subsequent paper.

For three decades the numerical solution to partial differential has been explored by practicing scientists, engineers and graduate students in engineering, and a decade ago it was still a formidable task often employing a library of general purpose subroutines to implement a particular numerical solution technique.

The recent availability of MATLAB for desktop computing has provided a computing environment with robust built-in routines for matrix manipulation and seamless two and three dimensional graphics for presentation of the solutions thus obtained. In short, using MATLAB turns efforts the duration of which was formerly measured in days to durations of a few hours.

In the past, implicit methods were often avoided because of the need to solve a set of algebraic equations at each step in time. In the case of linear problems this is reflected by a need to invert a matrix at each step in time. In the case of linear and time and spatially invariant systems, this solution method requires the inversion of a matrix only once at the beginning of the problem. The difficulties cited above are no longer an impediment to problems of moderate size.

In the past, explicit methods were attractive because they did not require solution of a system of algebraic equations. The disadvantage of explicit methods is that they tend to be numerically unstable, particularly when the system being investigated is lightly damped.

Although MATLAB has become the computational tool of choice for systems and signal processing, the predecessors (LINPACK and EISPACK) were developed with the solution of systems of partial differential equations associated with nuclear engineering in mind. It is also interesting to note that The Math Works has



very recently announced availability of the Partial Differential Equation Toolbox [2]. It seems prescient then to forge on with this strength,

A recently published paper [1] explores the use of MATLAB as a tool for the solution of hyperbolic partial differential equations, and this spirit will be continued herein with the following example.

An Example

Consider the example of a taut vibrating string which for small motions is governed by the wave equation

$$\frac{\partial^2 y}{\partial t^2} = \frac{\partial^2 y}{\partial x^2} \quad (1)$$

where x is the scaled spatial variable such that the spatial interval is of unit length, and t is the scaled temporal variable which is chosen to make the wave velocity unity. The boundary conditions are

$$y(0, t) = y(1, t) = 0 \quad (2)$$

and initial conditions

$$y(x, 0) = f(x) \quad (3)$$

and

$$\frac{\partial y(x, 0)}{\partial t} = 0 \quad (4)$$

Since the system is of second order in time, it is attractive to decompose it into two coupled first order stems or define $v(x, t)$ to be the velocity of the string so

$$\frac{\partial y}{\partial t} = v \quad (5)$$

and

$$\frac{\partial v}{\partial t} = \frac{\partial^2 y}{\partial x^2} \quad (6)$$

If we approximate the temporal derivatives with a backward difference and the second spatial derivative with a central difference, we get at some location x_i

$$\frac{y_i(t) - y_i(t - \Delta t)}{\Delta t} = v_i(t) \quad (7)$$

$$\frac{v_i(t) - v_i(t - \Delta t)}{\Delta t} = \frac{y_{i+1}(t) - 2y_i(t) + y_{i-1}(t)}{\Delta x^2} \quad (8)$$

for $i = 1, \dots, N$. The subscripts denote the location at which the velocity and displacement are evaluated, The two boundary conditions of(2) dictate that $y_0(t) = y_{N+1}(t) = 0$ for all t . Expressions (7) and (8) can be rewritten as backward difference equations or

$$y_i(t - \Delta t) = y_i(t) - \Delta t v_i(t) \quad (9)$$

and

$$v_i(t - \Delta t) = v_i(t) - \frac{\Delta t}{\Delta x^2} (y_{i+1}(t) - 2y_i(t) + y_{i-1}(t)) \quad (10)$$

for $i = 1, \dots, N$. Further these equations can be rewritten in matrix form as

$$\begin{bmatrix} y_1(t - \Delta t) \\ \vdots \\ y_N(t - \Delta t) \\ v_1(t - \Delta t) \\ \vdots \\ v_N(t - \Delta t) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} y_1(t) \\ \vdots \\ y_N(t) \\ v_1(t) \\ \vdots \\ v_N(t) \end{bmatrix} \quad (11)$$

where the A_{ij} are each respectively defined as

$$A_{11} = I_N \quad (12)$$

$$A_{12} = -I_N \Delta t \quad (13)$$

$$A_{22} = I_N \quad (14)$$

and

$$A_{21} = \begin{bmatrix} 2\delta & -\delta & 0 & \dots & 0 \\ -\delta & 2\delta & -\delta & \dots & 0 \\ 0 & & & & \vdots \\ \vdots & & -\delta & 2\delta & -\delta \\ 0 & \dots & 0 & -\delta & 2\delta \end{bmatrix} \quad (15)$$

where $\delta = \Delta t / (\Delta x)^2$.

The solution to the problem posed now involves the specification of an initial condition vector and the solution of the set of linear equations of (11) each step in time. This is most efficiently accomplished by inverting the associated matrix once at the start of the problem. For this example, $N = 21$ and $\Delta t = 0.1$ and the initial plucked position of the string is

$$f(x) = \begin{cases} 22x & 0 \leq x \leq \frac{5}{22} \\ \frac{110}{17}(1-x) & \frac{5}{22} \leq x \leq 1 \end{cases} \quad (16)$$

The method of solution outlined has been used to solve the problem for the first half cycle of string motion and the results are presented in the discussion that follows. The MATLAB code for this solution is included in Appendix A.

The string position for various values of time are shown in Figure 1. These results maybe corroborated by the knowledge of the D'Almbert solution to the wave equation for this problem [3]. The "granularity" of the solution near the extreme points corresponds to the Δx selected for this computation.

Figure 2 illustrates the motion of four points along the string, which in the case of the exact solution are each a trapezoidal wave shape, This illustrates why the tones of stringed instruments are so rich in harmonic overtones.

Figure 3 illustrates one of the very positive aspects of MATLAB because it allows the easy presentation of three dimensional plots to graphically illustrate the solution. In this case, the string motion as a function of both location and time are displayed. With a bit of patience and some careful data storage, the time history of the vibration can be graphically animated to help "tie the knot" of student comprehension.

The authors have used this simple technique to evaluate the solution to conduction heat transfer problems, contaminant transport problems, beam vibration problems with a variety of boundary conditions, and transmission line problems.

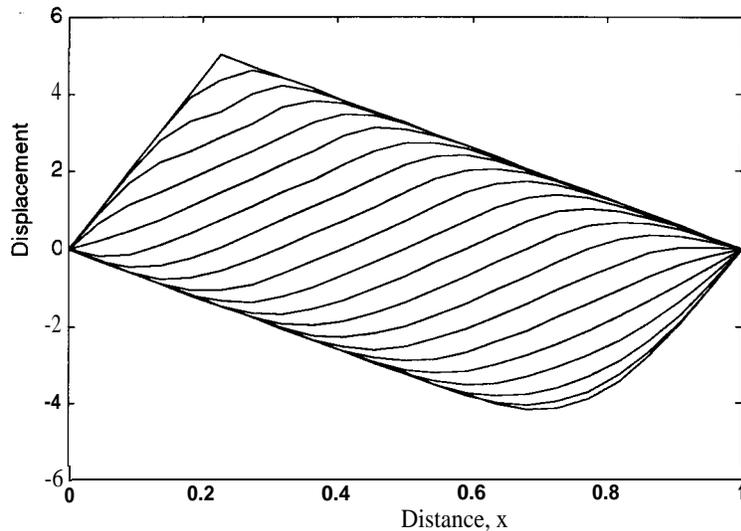


Figure 1. String Position at Various Times for a Half-Cycle of Motion.

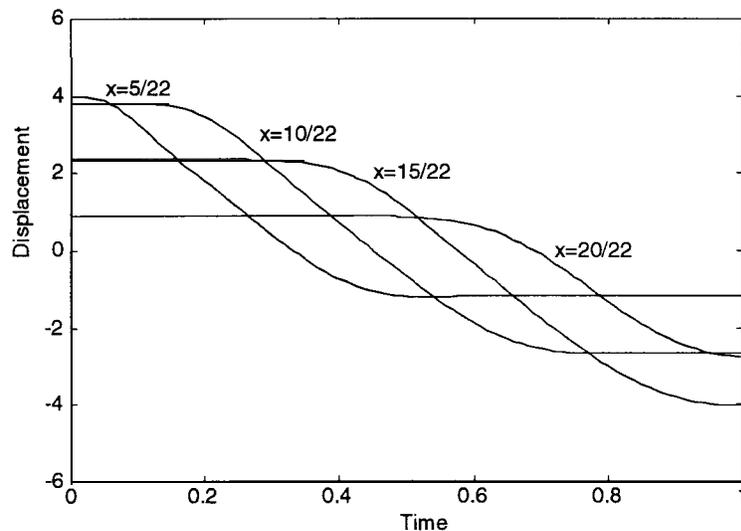


Figure 2. String Displacement as a Function of Time for Four Points on the String.

Conclusion

The authors have solved a number of partial differential equation problems over finite and semi-infinite spatial domains, similar to the flavor of the preceding example. It has been discovered that a MATLAB algorithm from one problem is quickly modified to solve an unrelated problem from an unrelated field of engineering.

The engineering education community has here-to-fore been somewhat reluctant to pursue the solution of partial differential equations at the undergraduate level. The authors argue that the major computational hurdles for numerical solutions no longer exist thanks to contemporary tool such as MATLAB. In the interest of our students' understanding, perhaps we should now press forward with exploiting the capabilities at our desktops.



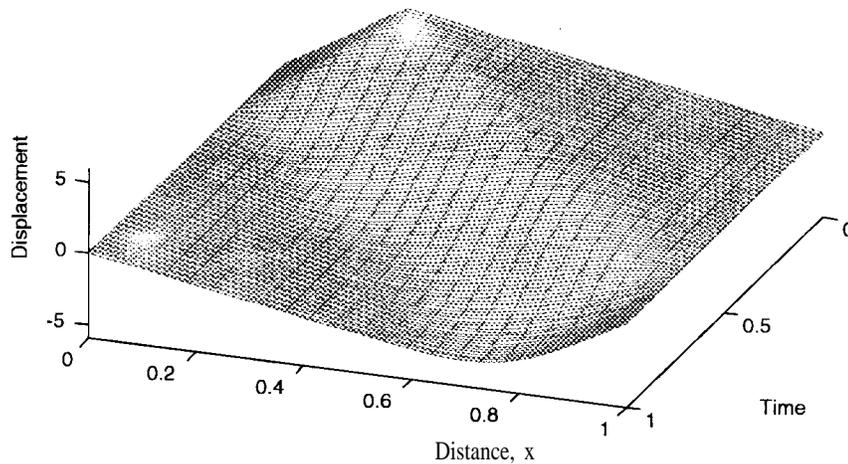


Figure 3. A Three Dimensional Plot of String Displacement as a Function of Location and Time.

References

1. Mathews, J. H., "Using MATLAB to Obtain Both Numerical and Graphical Solutions to Hyperbolic PDE's," *Computers in Education Journal*, vol. 4, no. 1, Jan/Mar, 1994, pp.58-60.
2. _____ "Partial Differential Equation Toolbox," *MATLAB News and Notes*, The Math Works, Natick, MA, Fall, 1995.
3. Wylie, C. R. and L. C. Barrett, *Advanced Engineering Mathematics: 5th Ed.*, McGraw-Hill, New York, NY, 1982.

Appendix A

This m-file solves the wave equation
 % for a % half cycle of motion of a
 % plucked string.

```
dt=.01;
dx=(1/22);
delta=dt/(dx^2);
a11=eye(21,21);a12=zeros(21,21);
a21=zeros(21,21);a22=eye(21,21);
for i=1:21
    a12(i,i)=-dt;
    a21(i,i)=2*delta;
end
for i=1:20
    a21(i,i+1)=-delta;
    a21(i+1,i)=-delta;
end
a=[a11 a12;a21 a22];
ainv=inv(a);
% set up initial conditions
```

```
for i=1:5
    uold(i,1)=i;
end
for i=6:21
    uold(i,1)=(110-5*i)/17;
end
vold=zeros(21,1);
xold=[uold ;vold];
usave=[0;uold;0];

% solve problem forward in time
for k=1:100
    xnew=ainv*xold;
    usave=[usave [0;xnew(1:21);0]];
    xold=xnew;
end
% end problem solution
% The remainder of this code is for
% visualization of the solution
figure(2),hold off, clf
```

```

xind=linspace(0,1,23);
plot(xind,usave(:,1))
axis([0 1 -6 6])
hold on
for i=6:5:101
    plot(xind,usave(:,i))
    axis([0 1 -6 6])
end
xlabel('Distance, x')
ylabel('Displacement')
pause
figure(1);hold off;clf
tind=[0:100]*dt;
plot(tind,usave(20,:))
hold on
for i=15:-5:5
    plot(tind,usave(i,:))

```

```

axis([0 1 -6 6])
end
xlabel('Time');
ylabel('Displacement');
text(.05,4.2,'x=5/22')
text(.28,3,'x=10/22')
text(.45,2,'x=15/22')
text(.72,0,'x=20/22')
pause
figure(3);clf
mesh(tind,xind,usave)
axis([0 1 0 1 -6 6])
view(110,50)
xlabel('Time')
ylabel('Distance, x')
zlabel('Displacement')

```

Raymond G. Jacquot

Ray Jacquot received his BSME and MSME degrees at the University of Wyoming in 1960 and 1962 respectively. He was an NSF Science Faculty Fellow at Purdue University where he received the Ph.D. in Mechanical Engineering in 1969. He is currently Professor of Electrical Engineering at the University of Wyoming. He has authored three textbooks in the systems area and his recent interest in education has been in the development of interactive software for systems engineering. Of particular interest is software for the design of digital control systems. He is a member of IEEE, ASME, Sigma Xi and ASEE. He has been active in the Computers in Education Division (COED) of ASEE.

Jerry C. Hamann

Jerry Hamann received the BS degree in Electrical Engineering/Bioengineering option from the University of Wyoming in 1984. After working as a product support engineer with Hewlett-Packard he returned to the University of Wyoming as a National Science Foundation Graduate Fellow to complete an MS in Electrical Engineering in 1988. In 1993 he completed the Ph.D. in Electrical Engineering at the University of Wisconsin-Madison where his research was focused in automatic control systems. He is currently Assistant Professor of Electrical Engineering at the University of Wyoming. His professional interests include the modeling and analysis of uncertain dynamic systems, algorithms, and software for simulation of dynamic systems and applications of digital signal processing in real-time. He is a member of IEEE, ASEE, Tau Beta Pi, Phi Kappa Phi and Golden Key Honorary.

