**Session 2220**

# Solution of Static and Dynamic Beam Bending and Static Buckling Problems Using Finite Differences and MATLAB

**R. G. Jacquot and B. R. Dewey**
**University of Wyoming**

## Abstract

The authors present here a way to utilize MATLAB for the solution of a class of static and dynamic solid mechanics problems. In particular, the authors demonstrate with simple codes that complicated problems of bending, buckling and beam vibration can be solved with high accuracy and, in the case of beam vibration, can also be animated.

## 1. Introduction

The solution of  bending and buckling problems is integral to the study of civil, mechanical and aerospace engineering.  The academic introduction to the bending of beams with constant cross section is usually given to students of these disciplines early in the engineering curriculum in a course in mechanics of materials.  That which is presented here is not meant to supplant the fundamental understanding of the bending phenomenon but rather to provide an introduction to numerical approximations often needed for real-world applications. It is important for students to first gain understanding of the static bending problems from an analytical and physical point of view.  Typical courses where static bending problems are encountered are structural analysis, advanced mechanics of materials, aircraft structures and design of machine elements.  Dynamic bending is a topic commonly introduced in courses in vibrations, structural dynamics and applied mathematics. The topic of buckling is often introduced in an introductory structures course. Advantageous applications would be to static bending and buckling problems where additional stiffnesses are attached at discrete points along the beam or column and dynamic problems where lumped mass or stiffness is added to the beam. The technique is also useful when the geometric or material properties vary along the beam.

The visualization  of solutions to differential equations has received attention previously in the educational literature [1-4].  Moreover computer animation has been employed to clarify solutions to such problems so students can understand the nature of such solutions [1,4].

The use of finite differences in the spatial dimension results in linear algebraic equations for static bending problems and linear ordinary differential equations for dynamic bending problems. MATLAB is particularly attractive for the solution of such problems because of the very robust solvers that are intrinsic to MATLAB for linear algebraic equations and ordinary differential equations. Similarly the column buckling problem can be solved with  MATLAB as an algebraic eigenvalue problem. In addition to ease of computation MATLAB has powerful two- and three-dimensional graphics for presentation of the results of such computations. Much of that material included here is included in the excellent text of James, Smith and Wolford[5] but the solutions are accomplished employing MATLAB as the solution tool instead of FORTRAN.

## **2**. Central Differences

Central differences are commonly employed to approximate derivatives of functions in a wide variety of applications. If the spatial independent variable x is discretized with equal increments Δx, the approximations for the derivatives of a function w(x) at a point $x_i$ are

$$\begin{aligned}
\left.\frac{dw}{dx}\right|_{x_i} &\cong \frac{w_{i-1} - w_{i+1}}{2\Delta x} \\
\left.\frac{d^2w}{dx^2}\right|_{x_i} &\cong \frac{w_{i-1} - 2w_i + w_{i+1}}{(\Delta x)^2} \\
\left.\frac{d^3w}{dx^3}\right|_{x_i} &\cong \frac{-w_{i-2} + 2w_{i-1} - 2w_{i+1} + w_{i+2}}{2(\Delta x)^3} \\
\left.\frac{d^4w}{dx^4}\right|_{x_i} &\cong \frac{w_{i-2} - 4w_{i-1} + 6w_i - 4w_{i+1} + w_{i+2}}{(\Delta x)^4}
\end{aligned} \qquad (1)$$

## 3. Model Development

To develop a solvable model the beam or column is segmented into an integral number of sections of equal length and the Bernoulli-Euler beam equation is approximated at the junctures of those sections with the central difference approximation. The two boundary conditions at each end of the beam are applied and result in a modification of the corner elements of the associated stiffness matrix.

## 4. Static Bending of a Cantilever Beam

Consider the cantilever beam shown in Figure 1 with the linearly varying load shown.

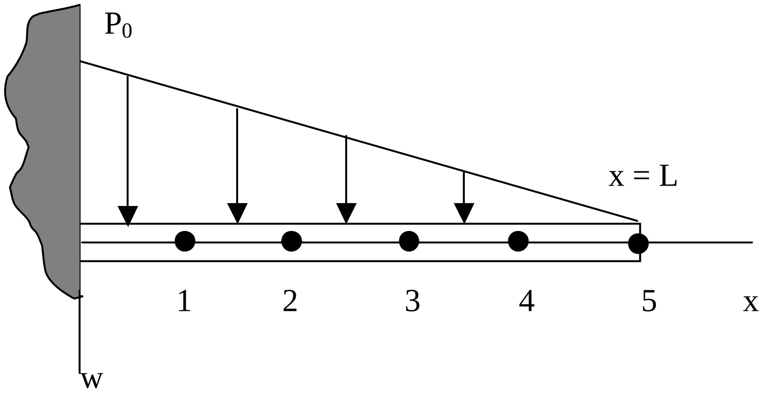

Figure 1. Cantilever Beam with Linearly Varying Load

For small deflections and elastic material the beam is governed by the Bernoulli-Euler equation where the right side represents the applied transverse load

$$EI\frac{d^4w}{dx^4} = P_0(1-\frac{x}{L}) \tag{2}$$

where EI is the bending stiffness. If nodes are established as indicated in Figure 1 then using the finite difference approximation at each node yields

$$\frac{EI}{(\Delta x)^4}[w_{i-2} - 4w_{i-1} + 6w_i - 4w_{i+1} + w_{i+2}] = P_0(1-\frac{i\Delta x}{L}) \tag{3}$$

If five nodes are used then the clamped boundary condition at the left end implies that the deflection and slope are zero so

$$\begin{aligned} w_0 &= 0 \\ w_{-1} &= w_1 \end{aligned} \tag{4}$$

The zero shear and zero moment condition at the free end implies that the second and third derivatives must vanish at node 5 or

$$\begin{aligned} w_6 &= 2w_5 - w_4 \\ w_7 &= w_3 - 4w_4 + 4w_5 \end{aligned} \tag{5}$$

then the resulting algebraic equations are in matrix form

$$\begin{bmatrix} 7 & -4 & 1 & 0 & 0 \\ -4 & 6 & -4 & 1 & 0 \\ 1 & -4 & 6 & -4 & 1 \\ 0 & 1 & -4 & 5 & -2 \\ 0 & 0 & 2 & -4 & 2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{bmatrix} = \frac{P_0(\Delta x)^4}{5EI} \begin{bmatrix} 4 \\ 3 \\ 2 \\ 1 \\ 0 \end{bmatrix} \tag{6}$$

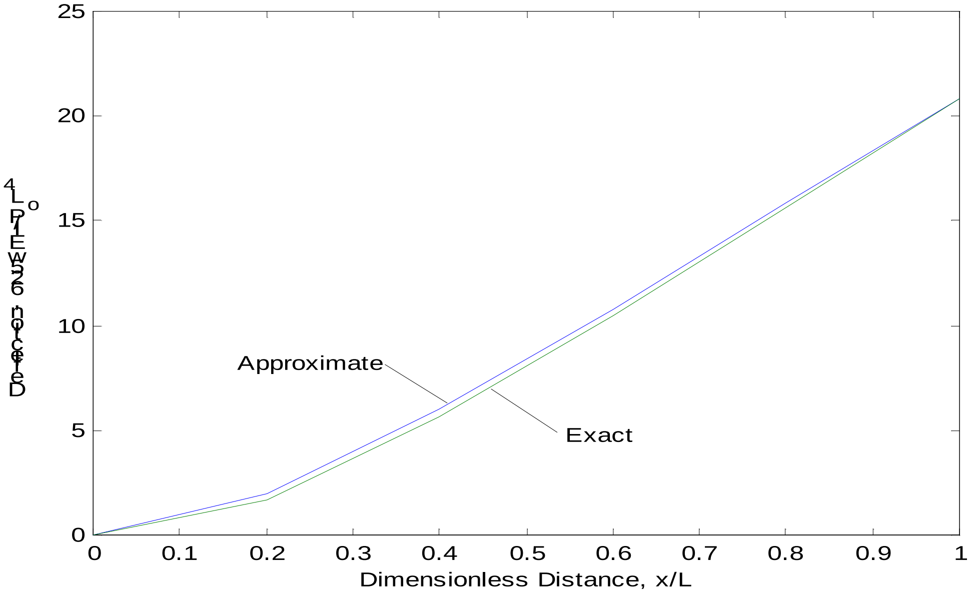

Figure 2. Approximate and Exact Cantilever Beam Deflection

For a beam with variable properties the EI terms would be included in the square matrix. Solution of the linear equations is a simple operation using the left divide operator (\) in MATLAB. A simple script file for the problem is given in Appendix A-1 and the a plot of the five node solution and the exact solution are shown in Figure2

5. A Buckling Problem

Consider the clamped-free column illustrated in Figure 3 with x = 0 at the clamped end and axial load P. The governing differential equation is

$$EI\frac{d^2w}{dx^2} = P(\delta - w) \qquad (7)$$

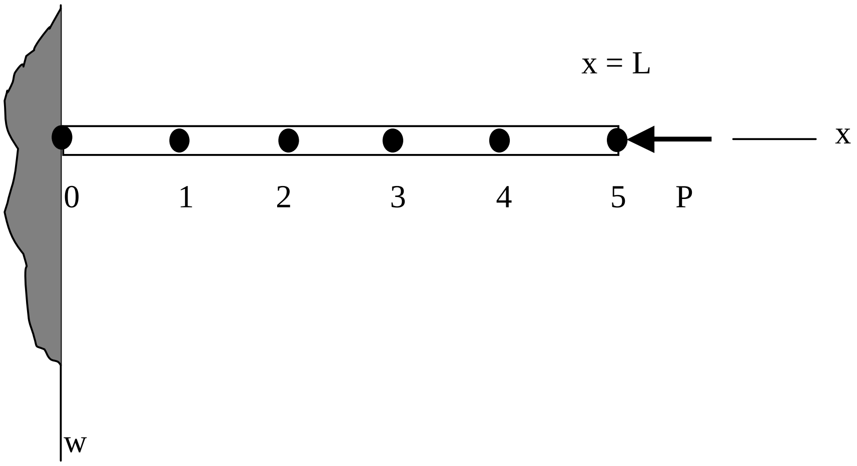

Figure 3. Clamped –Free Column with Axial Load

Approximation of the second derivative with the second of equations (1) the resulting difference equation is

$$\frac{EI}{(\Delta x)^2}[w_{i-1} - 2w_i + w_{i+1}] = P[w_n - w_i] \qquad (8)$$

where $w_n$ is the tip deflection δ. If 5 nodes are employed the boundary conditions are those of equations (5). The resulting equations for nodes zero to four in matrix form are

$$\frac{EI}{(\Delta x)^2}\begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{bmatrix} = P\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{bmatrix} \qquad (9)$$

where the upper left elements of the matrix reflect the clamped boundary condition at the left end.. The goal is to find the value of $P(\Delta x)^2/EI$ which satisfies equation (9). If we premultiply by the inverse of the right hand matrix we have an ordinary eigenvalue problem which is easily solved with MATLAB for the eigenvalues and associated eigenvector. The smallest eigenvalue is most often of most interest because it corresponds to the lowest critical load. The deflection shape is shown in Figure 4 in which is also presented the lowest critical load which is in error by only 0.8% from the analytical solution. The script file for he buckling problem is given in Appendix A-2.

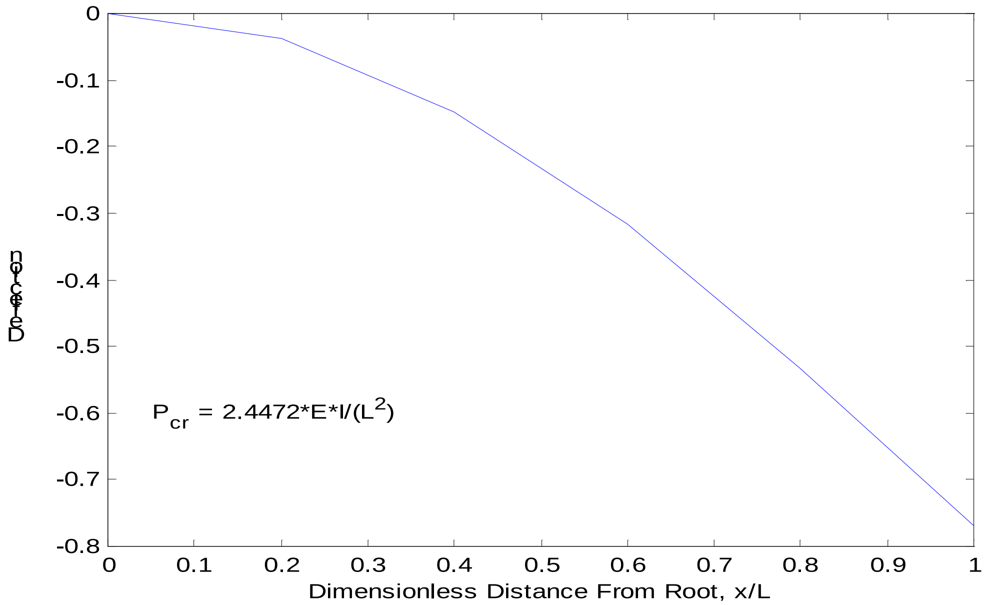

Figure 4. Buckling Deflection Curve and Critical Load

The techniques just demonstrated are also applicable to beam-columns which involve combinations of axial and lateral loads. Such problems are not eigenvalue problems but rather spatial differential equation problems wherein both the second and fourth derivative of the deflection are approximated with appropriate differences.

6. Cantilever Beam Vibration

The free and forced lateral vibration of a uniform, slender beam is governed by the Bernoulli-Euler equation

$$\mu\frac{\partial^2 w}{\partial t^2} + EI\frac{\partial^4 w}{\partial x^4} = f(x,t) \qquad (10)$$

where EI is the bending stiffness, $\mu$ is the mass per unit length and f(x, t) is the lateral forcing function. If the derivative with respect to x is approximated with the fourth of relations (1) the result is

$$\mu\frac{d^2 w_i}{dt^2} + \frac{EI}{(\Delta x)^4}[w_{i-2} - 4w_{i-1} + 6w_i - 4w_{i+1} + w_{i+2}] = f(x_i, t) \qquad (11)$$

Now scale time to a new temporal variable $\tau$ such that

$$\tau = \frac{t}{(\Delta x)^2}\sqrt{\frac{EI}{\mu}} \qquad (12)$$

*Proceedings of the 2001 American Society for Engineering Education Annual Conference & Exposition*
*Copyright ©, American Society for Engineering Education*

then the equations of motion become

$$\frac{d^2 w_i}{d\tau^2} = [w_{i-2} - 4w_{i-1} + 6w_i - 4w_{i+1} + w_{i+2}] + \frac{(\Delta x)^4}{EI} f(x_i, \tau) \tag{13}$$

Now define velocities as $v_i = dw_i/d\tau$ and the column vectors $\mathbf{w} = [w_1\ w_2\ \ldots w_8]^T$ and $\mathbf{v} = [v_1\ v_2\ \ldots\ v_8]^T$ so the resulting state variable matrix form is

$$\frac{d}{d\tau}\begin{bmatrix}\mathbf{w}\\ \mathbf{v}\end{bmatrix} = \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ \mathbf{-S} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{w}\\ \mathbf{v}\end{bmatrix} + \frac{(\Delta x)^2}{EI}\begin{bmatrix}\mathbf{0}\\ \mathbf{f}\end{bmatrix} \tag{14}$$

where **S** is a stiffness matrix similar to that of equation (6) except it is 8x8 and **I** is the 8x8 identity matrix while $\mathbf{f} = [f(x_1,\tau)\ f(x_2,\tau)\ \ldots\ f(x_8,\tau)]^T$. This is ideally suited to the solution of the governing differential equations in MATLAB using the very robust solver ode45. Here the authors chose to solve an unforced problem wherein the beam is given an initial deflection and zero initial velocity. It was also decided to solve the equations of motion for a time equal to the first mode period. Solutions of PDEs have been previously presented graphically [2,3] and such a plot of the beam motion as a function of location and time is presented in Figure 5.

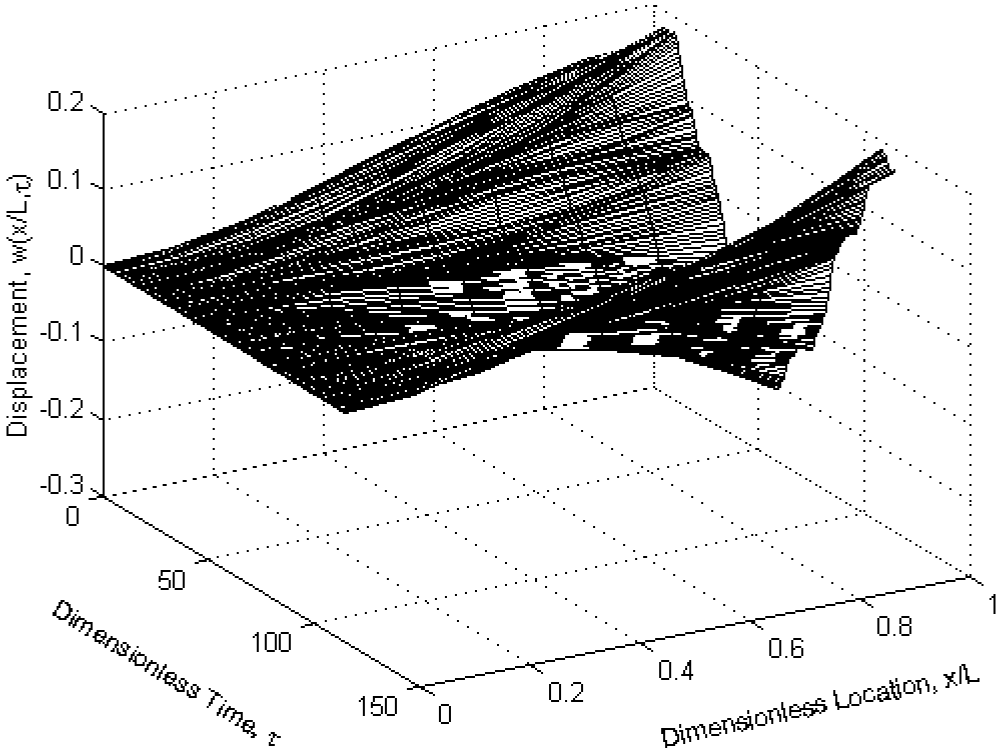

Figure 5. Solution to the Beam Equation as a Function of Location and Time.

More recently, the animation of differential equation solutions has been demonstrated in MATLAB by employing the handle graphics in MATLAB[4] and that has been accomplished here to animate the solution to the problem. Every fifteenth animation frame for one half the first mode period is illustrated in Figure 6. The script file is presented in Appendix A-3.

## 7. Conclusion

The solution of a series of bending problems in MATLAB has been presented and the use of graphics for illustration has been presented. This problem solving methodology provides an intuitive introduction to the use of finite difference techniques in the solution of mechanics problems.

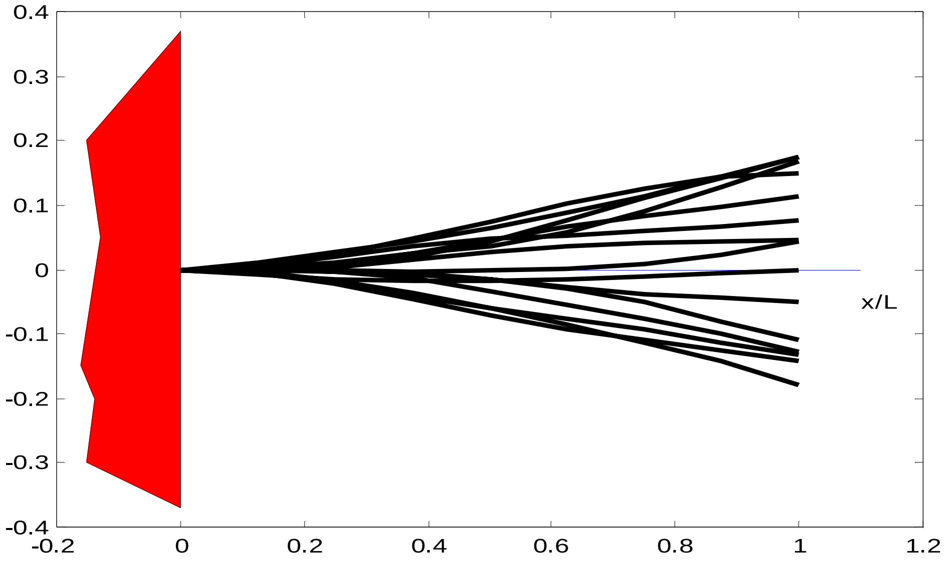

Figure 6. Half Cycle of Animation Frames for Beam Vibration.

Bibliography

1. Jacquot, R.G., D.A. Smith and D.L. Whitman, Animation of Multiple Degree of Freedom Mechanical Systems, *Proc. 1990 ASEE Annual Conference*, June 24-28, Toronto, Ontario, Canada, pp.1302 –1305.

2. Matthews, J.H., Using MATLAB to Obtain Both Numerical and Graphical Solutions to Hyperbolic PDEs, *Computers in Education Journal*, vol. 4, no. 1, Jan./Mar., 1994, pp. 58-60.

3. Jacquot, R.G. and J.C. Hamann, Visualization of PDE Solutions Using Implicit Methods and MATLAB, Computers In Education Journal, vol. 7, no. 3, July/Sept., 1997, pp. 2-5.

4. Watkins, J., G. Piper, K. Wedeward and E.E. Mitchell, Computer Animation: A Visualization Tool for Dynamic Systems Simulations, *Proc. 1997 ASEE Annual Conference*, June 15-18, 1997, Milwaukee, WI, Paper 1620-4.

5. James, M.L., G.M. Smith and J.C. Wolford, *Applied Numerical Methods for Digital Computation; 3rd Ed.*, Harper and Row, New York, 1985.

6 Logan, D.L., *A First Course in the Finite Element Method*, PWS Publishing, Boston, 1993.

## Appendix A

### A-1 Beam Deflection Problem

```
% Cantilever Beam with Linearly decreasing load
n=5; % Number of nodes excluding root
j=[0:1:n]; %nodal indices
% Build matrix
S=[7 -4 1 zeros(1,n-3);-4 6 -4 1 zeros(1,n-4)];
for i=3:n-2
   S=[S;zeros(1,i-3)1 -4 6 -4 1 zeros(1,n-(i+2))];
end
S=[S;zeros(1,n-4)1 -4 5 -2];
S=[S; zeros(1,n-3) 2 -4 2];
load=[];
%Build load vector
for i=1:n
   load=[load;(n-i)/n];
end
w=[ 0 (S\load)’]; %Solve linear equations
w1=(625*(j.^2)/((n^2)*120)).*(10-(10*j/n)+(5*(j.*j)/(n^2))-
((j.*j.*j)/(n^3)));
xoverl=j/n; % Dimensionless Distance
plot(xoverl,w,xoverl,w1)
xlabel(’Dimensionless Distance, x/L’)
ylabel(’Deflection, 625wEI/P_oL^4’)
```

### A-2 Column Buckling Problem

```
% Cantilevre Buckling Problem
n=8;% Number of nodes away from origin
A=[];
B=[];
d=[1 -2 1];
for i=3:n
   A=[A;zeros(1,i-3) d zeros(1,n-i)];
end
for i=2:n
   B=[B;zeros(1,i-2) -1 zeros(1,n-i) 1];
end
A=[2 zeros(1,n-1);-2 1 zeros(1,n-2);A];
B=[zeros(1,n-1) 1;B];
M=inv(B)*A;
[vec,val]=eig(M);
v1=[];
for i=1:n
   v1=[v1 val(i,i)];
end
[lam,ind]=min(v1);
xoverl=[0:1:n]/(n);
plot(xoverl,[0 vec(1:n,ind)’])
xlabel(’Dimensionless Distance From Root, x/L’)
ylabel(’Deflection’)
pf=lam*(n^2);
text(.05,-.6,[’P_cr = ’ num2str(pf) ’*E*I/(L^2)’])
```

### Appendix A-3 Cantilever Beam Vibration Problem

```
% Cantilever vibration
```

```
clf
n=8; % number of nodes
S=[7 -4 1 zeros(1,n-3)];
S=[S;-4 6 -4 1 zeros(1,n-4)];
for i=3:n-2
   S=[S;zeros(1,i-3)1 -4 6 -4 1 zeros(1,n-i-2)];
end
S=[S;zeros(1,n-4) 1 -4 5 -2];
S=[S;zeros(1,n-3) 2 -4 2];
global A
xoverl=[0:1/n:1];
xoverl1=xoverl(:,2:n+1);
w0=.15*(3*(xoverl1.^2)-2*(xoverl1.^3));
x0=[w0 zeros(1,n)];
A=[zeros(n,n) eye(n,n);-S zeros(n,n)];
tspan=linspace(0,1.787*(n^2),300);
[t,x]=ode45(’beamrhs’,tspan,x0);
w=x(:,1:n);
sz=size(t);
w=[zeros(sz(1,1),1)w];
[T,X]=meshgrid(t,xoverl);
figure(1)
clf
set(gca,’Box’,’on’)
xp=[ 0  0 -.15 -.13 -.16 -.14 -.15 0];
yp=[-.37 .37 .2 .05 -.15 -.2 -.3 -.37];
patch(xp,yp,’r’);
hold on
plot([0 1.1],[0 0])
L=plot(xoverl,zeros(1,n+1),’k’,’EraseMode’,’xor’,’LineWidth’,[2.5]);
hold on
text(1.1,-.05,’x/L’)
hold on
axis([-.2 1.2 -.4 .4])
thandl=text(0.2,-0.35,’Press Enter to Set Initial Condition’);
pause
for i=1:sz(1,1)
   set(L,’YData’,w(i,:));
   if i==1
      set(thandl,’String’,’Press Enter to Animate’);
      pause
   end
   drawnow;
end
set(thandl,’String’,’Press Enter to Continue’);
pause
figure(3)
mesh(T,X,w’)
colormap([0 0 0]);
view(60,30)
%axis([0 45 0 1 -.3 .3])
xlabel(’Dimensionless Time, \tau’,’rotation’,-31)
ylabel(’Dimensionless Location, x/L’,’rotation’,12)
zlabel(’Displacement, w(x/L,\tau)’)


function xdot=beamrhs(t,x)
```

```
global A
xdot=A*x;
```

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 a member of ASEE, IEEE and ASME and has been active in ASEE for the past two decades serving as Rocky Mountain Section Chair and PIC IV Chair. His professional interests are in modeling, control, simulation and animation of dynamic systems. He currently serves as Professor of Electrical Engineering.

BRUCE R. DEWEY
Dr. Dewey is a graduate of Iowa State University (B.S. in Mechanical Engineering, 1959, and M.S. in Nuclear Engineering,1964) and the University of Illinois (Ph.D. in Theoretical and Applied Mechanics, 1967). In 1982, he moved to UW. He has worked as a consultant and  research collaborator for Union Carbide, Oak Ridge National Laboratory, General Electric, Laawrence Livermore Laboratory and Pafec, Ltd. His research interests are in finite element analysis, ultrasonic wave phenomena, solid modeling and computer graphics. He is author of the text Computer Graphics for Engineers and is a member of ASME, ASEE, Sigma Xi, and the IEEE Computer Society. He currently serves as Professor of Mechanical Engineering.