Utilization of MATLAB in Structural Analysis

Shahnam Navaee, Nirmal K. Das
Georgia Southern University

Abstract

In this paper an alternate approach to analyzing structures using MATLAB software is discussed. The procedure is to be implemented in teaching a structural analysis course offered in the Civil Engineering Technology Program at Georgia Southern University in the fall semester of 2002. A series of carefully selected set of problems are designed to familiarize the students to the MATLAB programming tools needed to analyze statically determinate as well as indeterminate structures. The problems designed for this project cover a number of major topics typically discussed in an introductory level structural analysis course such as equilibrium, shear and moment diagrams, and deflections. By performing the exercises selected for this course, the students learn how to utilize MATLAB to perform a variety of tasks related to analyzing structures. These tasks can involve activities such as determining the reactions in a simple statically determinate beam using static equilibrium considerations, or analyzing a more complicated indeterminate frame using the method of slope-deflection. The procedure implemented in this project arms the students with a powerful computational tool they could utilize to verify the accuracy of their developed hand-solutions. This project also helps the students to gain a more in-depth understanding of the structural concepts, since this knowledge is needed in writing MATLAB script files. By performing the exercises designed for this course, the students acquire a better appreciation for the power of computers and their application to solve structural analysis problems. Included in this paper are examples to illustrate the procedure described.

I. Introduction

MATLAB is a powerful computing software which is presently utilized in a number of educational institutions around the country to solve mathematics and engineering-related problems. The name of the software MATLAB stands for “Matrix Laboratory” since the built-in capabilities of this package are specifically designed for efficient handling of matrix and array operations. The effective and easy-to-use computing environment of MATLAB along with availability of a large number of helpful MATLAB built-in functions has rendered it the popular tool of choice for many educators in various engineering fields. Using the MATLAB interactive environment, programs placed in script files can easily be created and edited to perform the desired computations and to generate the needed output. The capabilities of MATLAB can further be enhanced by additional “toolbox” modules that can separately be purchased through The MathWorks, Inc., the company that produces the MATLAB software. These modules are designed to perform a variety of specialized tasks. The solutions presented in this paper are obtained using the basic features of MATLAB without utilizing any specialized MATLAB toolboxes.

In the submitted paper the procedure for solving structural analysis problems using MATLAB software is discussed. This procedure is to be implemented in teaching one section of the structural analysis course in the fall semester of 2002 in the School of Technology at Georgia Southern University. The goal of this project is not to replace or alter the traditional techniques and procedures used in teaching the subject, but as a means to complement the course and to make it more meaningful. The procedure is described in the paper through formulating and discussing the MATLAB solutions for three sample problems dealing with various topics and methods traditionally covered in a basic structural analysis course offered in most Civil Engineering and Civil Engineering Technology Programs. These problems are listed below.

- Analysis of statically determinate trusses (Method of Joints and Method of Sections)
- Analysis of statically determinate beams (shear and bending moment diagrams, computation of slopes and deflections).
- Analysis of statically indeterminate frames (Slope-Deflection Method)

The example problems presented in this paper are taken from the text book[1] used for the course. The script files for these problems are formulated in terms of beam and loading parameters to obtain a more useful and powerful program that can be utilized to analyze a wider variety of problems. The program results for the specific data presented in the text are also provided in the paper.

In order to enable the students to develop MATLAB script files for analyzing structures such as the ones listed above, a handout is developed by the authors. This handout is carefully organized to clearly explain the following concepts and topics through a number of easy-to-follow examples.

- Basis MATLAB software features
- Basic mathematical operations in MATLAB
- Creating MATLAB script files
- Array and matrix operations
- Solving a system of linear equations
- Controlling input and output
- Utilizing MATLAB library functions
- Creating user-defined functions
- Creating and utilizing conditional statements ("if" statements)
- Creating and utilizing loops ("for" and "while")
- Plotting with MATLAB

The authors believe that the prepared handout is sufficient in providing the students with the needed tools and skills necessary in developing script files for the assigned problems in the course. The procedure described in the paper encourages the students to pay careful attention to the details involved in the development of the theoretical formulation of the problem, knowing that this formulation is critical if they are to develop a script file that produces the correct results. Once the scripts are developed and the solutions are verified for a number of problems for which the solutions are known, the students can then use these scripts to verify their hand-solutions for other similar problems. The developed MATLAB programs can also serve as an excellent and effective educational tool to provide the students with better understanding of the behavior of structures under different loading conditions. That is, using these scripts, different scenarios can be created and studied by altering the geometry of the structure, the magnitude and type of applied loads, support conditions, and structure's material properties. Developing powerful script files that are capable of analyzing a wide variety of structures is perceived to generate a greater enthusiasm and interest among the students in the course.

II. Analysis of Statically Determinate Trusses

In this section of the paper, the procedure for analyzing a statically determinate truss using the MATLAB software is provided through outlining the method of solution for an example problem. The solution is based on the application of the method of joints and the method of sections. The application of both methods requires solving a system of linear equations.

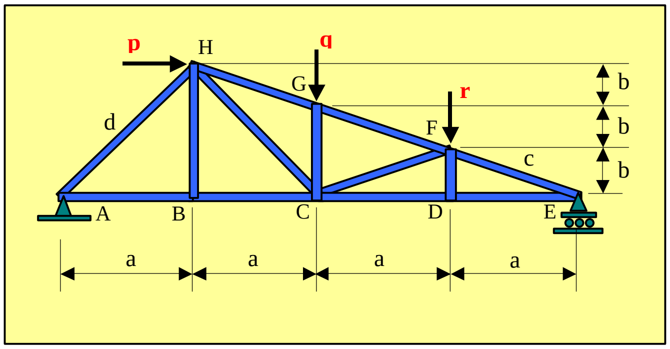


Figure 1. A Statically Determinate Truss

Method of Joints:

The determination of the member forces and support reactions in the truss shown in Figure 1 requires that two force equilibrium equations be written for each of the 8 joints of the truss. This yields a total of 16 equations that can be solved to yield the forces in the 13 members of the truss, and the 3 reactions at the supports at A and E. To obtain the MATLAB solution to this problem, the equilibrium equations for the truss are first formulated in a general format as shown in Eq. (1). Using this equation, the unknown column vector X can be solved for using the MATLAB left-division operation as shown in Eq. 2. For the above truss, the matrices A and B, and the column vector X containing the unknowns in the problem are provided in Eq. 3. Note that in this equation $c = \sqrt{a^2 + b^2}$ and $d = \sqrt{a^2 + 9b^2}$. Note that c and d are the length of the members EF and AH respectively as shown in Figure 1. When formulating the MATLAB program for this type of problem, the "zeros" function can be utilized to generate a series of zeros in the matrix in an easy and convenient way. The script file for this truss and the generated output for the case when a = 10 ft, b = 4 ft, p = 2 kip, q = 3 kip, and r = 3 kip are shown in Figure 2. Note that the script file for the problem is developed in a fashion to allow the user to enter any specific values for the dimensions a and b, and for the applied loads p, q, and r.

$$AX=B \tag{1}$$

$$X=A\backslash B \tag{2}$$

$$\begin{bmatrix}
1 & a/d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 3b/d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 1 & a/c & 0 & -a/d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & b/c & 1 & 3b/d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -a/c & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b/c & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & -a/c & 0 & 0 & 0 & 0 & a/c & -a/c & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -b/c & 0 & 0 & -1 & 0 & -b/c & b/c & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a/c & -a/c & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -b/c & b/c & 0 & 0 & 0 \\
0 & -a/d & 0 & 0 & 0 & 0 & 0 & a/d & 0 & 0 & 0 & 0 & a/c & 0 & 0 & 0 \\
0 & -3b/d & 0 & -1 & 0 & 0 & 0 & -3b/d & 0 & 0 & 0 & 0 & -b/c & 0 & 0 & 0
\end{bmatrix}
\begin{Bmatrix} F_{AB} \\ F_{AH} \\ F_{BC} \\ F_{BH} \\ F_{CD} \\ F_{CF} \\ F_{CG} \\ F_{CH} \\ F_{DF} \\ F_{DE} \\ F_{EF} \\ F_{FG} \\ F_{GH} \\ A_y \\ E_x \\ E_y \end{Bmatrix}
=
\begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ r \\ 0 \\ q \\ -p \\ 0 \end{Bmatrix} \tag{3}$$

Method of Sections:

The method of sections for analyzing a truss is normally utilized in situations where only the forces in specific members are to be computed. In the given truss, to determine the forces in members BC, CH, and GH using this method, these members are cut and two force equilibrium equations and one moment equilibrium equation are written for the section of the truss as shown in Figure 3. The three equations can be arranged in the matrix form as shown in Eq. (4), and the MATLAB solution for the unknowns in the problem can be obtained using the left-division operation as done in the previous method. Note that prior to determining the unknown member forces F_{BC}, F_{CH}, and F_{GH} in Eq. (4), it is necessary to compute the reaction A_y at support A using the equilibrium of the entire truss. The expression for A_y is provided in Eq. 5. The script file for this problem and the corresponding results for the data already presented are provided in Figure 5.

```
% Analysis of a Truss Utilizing the Method of Joints
% ________________________________________________________________________________
% Program objective:
% To compute the member forces and support reactions of the given truss utilizing
% the method of joints.
% ________________________________________________________________________________
% Data acquisition:
p=input('Enter the value of force p(kip):\n');
q=input('Enter the value of force q(kip):\n');
r=input('Enter the value of force r(kip):\n');
a=input('Enter the value for the dimension a(ft):\n');
b=input('Enter the value for the dimension b(ft):\n');
% ________________________________________________________________________________
% Computation of the truss member forces and support reactions:
c=sqrt(a^2+b^2);
d=sqrt(a^2+9*b^2);
A(1,:)=[1,a/d,zeros(1,14)];
A(2,:)=[0,3*b/d,zeros(1,11),1,zeros(1,2)];
A(3,:)=[-1,0,1,zeros(1,13)];
A(4,:)=[zeros(1,3),1,zeros(1,12)];
A(5,:)=[zeros(1,2),-1,0,1,a/c,0,-a/d,zeros(1,8)];
A(6,:)=[zeros(1,5),b/c,1,3*b/d,zeros(1,8)];
A(7,:)=[zeros(1,4),-1,zeros(1,4),1,zeros(1,6)];
A(8,:)=[zeros(1,8),1,zeros(1,7)];
A(9,:)=[zeros(1,9),-1,-a/c,zeros(1,3),-1,0];
A(10,:)=[zeros(1,10),b/c,zeros(1,4),1];
A(11,:)=[zeros(1,5),-a/c,zeros(1,4),a/c,-a/c,zeros(1,4)];
A(12,:)=[zeros(1,5),-b/c,zeros(1,2),-1,0,-b/c,b/c,zeros(1,4)];
A(13,:)=[zeros(1,11),a/c,-a/c,zeros(1,3)];
A(14,:)=[zeros(1,6),-1,zeros(1,4),-b/c,b/c,zeros(1,3)];
A(15,:)=[0,-a/d,zeros(1,5),a/d,zeros(1,4),a/c,zeros(1,3)];
A(16,:)=[0,-3*b/d,0,-1,zeros(1,3),-3*b/d,zeros(1,4),-b/c,zeros(1,3)];
B=[zeros(11,1);r;0;q;-p;0];
X=A\B;
% ________________________________________________________________________________
% Outputing the member forces and support reactions:
fprintf('Member Forces(kip):\n\n')
fprintf('AB = %5.3f AH = %5.3f BC = %5.3f BH = %5.3f CD = %5.3f CF = %5.3f CG = %5.3f\n\n',...
  X(1),X(2),X(3),X(4),X(5),X(6),X(7))
fprintf('CH = %5.3f DF = %5.3f DE = %5.3f EF= %5.3f FG = %5.3f GH = %5.3f\n\n',...
  X(8),X(9),X(10),X(11),X(12),X(13))
fprintf('Support Reactions(kip):\n\n')
fprintf('Ay = %5.3f Ex = %5.3f Ey = %5.3f\n',X(14),X(15),X(16))
```

Output:

```
Member Forces(kip):

AB = 1.375 AH = -2.148 BC = 1.375 BH = 0.000 CD = 8.875 CF = -4.039 CG = -3.000

CH = 5.858 DF = 0.000 DE = 8.875 EF= -11.713 FG = -7.674 GH = -7.674

Support Reactions(kip):

Ay = 1.650 Ex = 2.000 Ey = 4.350
```

Figure 2. The MATLAB Script File and a Sample Output for the Analysis of a Truss Using the Method of Joints

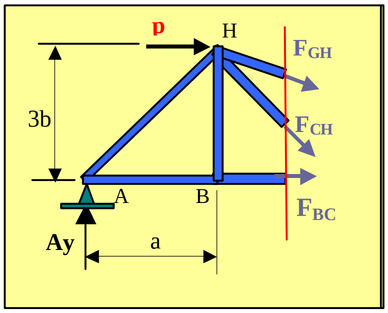


Figure 3. Free Body Diagram of the Left Section of the Truss

$$\begin{bmatrix} 1 & a/d & a/c \\ 0 & -3b/d & -b/c \\ 0 & -3ab/d & -3ab/c \end{bmatrix} \begin{Bmatrix} F_{BC} \\ F_{CH} \\ F_{GH} \end{Bmatrix} = \begin{Bmatrix} -p \\ -Ay \\ A_y\, a + 3pb \end{Bmatrix} \tag{4}$$

$$A_y = \frac{ra + 2qa - 3pb}{4a}. \tag{5}$$

III. Analysis of Statically Determinate Beams

The procedure for obtaining the shear and moment diagrams and plotting the variation of slope and deflection along the length of the beam is described through the solution of a sample beam and loading condition shown in Figure 4. Prior to developing the MATLAB script file for this problem, the following preliminary task involving the determination of the algebraic expressions for the shear, moment, slope, and deflection needs to be performed.

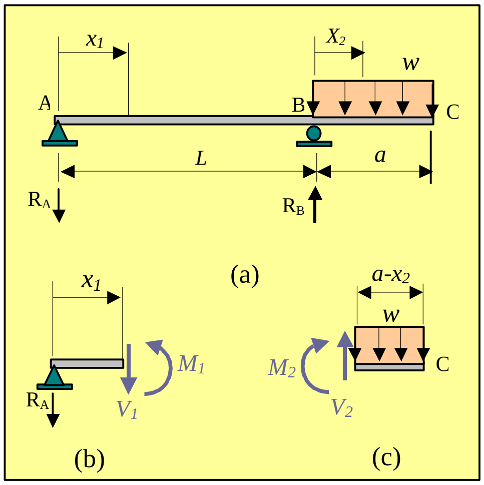


Figure 4. Free Body Diagrams for the Beam

```
% Analysis of a Truss Utilizing the Method of Sections.
% ____________________________________________________________________________
% Program objective:
% To compute the forces in members BC, CH, and GH of the given truss utilizing the method
% of sections.
% ____________________________________________________________________________
% Data acquisition:
p=input('Enter the value of force p(kip):\n');
q=input('Enter the value of force q(kip):\n');
r=input('Enter the value of force r(kip):\n');
a=input('Enter the value for the dimension a(ft):\n');
b=input('Enter the value for the dimension b(ft):\n');
% ____________________________________________________________________________
% Computation of the support reaction at A:
Ay=(r*a+2*q*a-3*p*b)/(4*a)
% ____________________________________________________________________________
% Computation of the truss member forces:
c=sqrt(a^2+b^2);
d=sqrt(a^2+9*b^2);
A(1,:)=[1,a/d,a/c];
A(2,:)=[0,-3*b/d,-b/c];
A(3,:)=[0,-3*a*b/d,-3*a*b/c];
B=[-p;-Ay;Ay*a+3*p*b];
F=A\B;
% ____________________________________________________________________________
% Outputing the member forces:
fprintf('Member Forces(kip):\n\n')
fprintf('BC = %5.3f CH = %5.3f GH = %5.3f\n',F(1),F(2),F(3))
```

Output:

```
Member Forces(kip):

BC = 1.375 CH = 5.858 GH = -7.674
```

Figure 5. The MATLAB Script File and a Sample Output for the Analysis of a Truss Using the Method of Sections

Writing the force and moment equilibrium equations for the free body diagrams of the two sections of the beam shown at the bottom of Figure 4, the following expressions for the shear force *V* and bending moment *M* can be established for each of the beam segments AB and BC.

$$0 \le x_1 \le L \qquad\qquad 0 \le x_2 \le a$$

$$V_1 = \frac{-wa^2}{2L} \qquad\qquad V_2 = w(a - x_2) \tag{6}$$

$$M_1 = \frac{-wa^2}{2L} x_1 \qquad\qquad M_2 = \frac{-w(a - x_2)^2}{2} \tag{7}$$

Upon substituting for the moments M_1 and M_2 in the differential equations of the two regions of the beam shown below:

$$\begin{array}{ll} 0 \le x_1 \le L & 0 \le x_2 \le a \\ EIv_1'' = M_1 & EIv_2'' = M_2 \end{array} \qquad (8)$$

the following two differential equations are obtained for the beam.

$$\begin{array}{ll} 0 \le x_1 \le L & 0 \le x_2 \le a \\ EIv_1'' = \dfrac{-wa^2}{2L} x_1 & EIv_2'' = \dfrac{-w(a - x_2)^2}{2} \end{array} \qquad (9)$$

In these expressions E, and I are respectively the modulus of elasticity and the moment of inertia of the beam. All other parameters are as defined in Figure 4. Upon utilizing the method of successive integration and applying the boundary conditions:

$$v_1(x_1 = 0) = 0,\ \ v_2\ (\mathrm{x}_2 = 0) = 0 \qquad (10)$$

stating that the deflection of the beam at the supports A and B are zero, and the continuity equations:

$$v_1'(x_1 = L) = v_2'(x_2 = 0),\ \ v_1(x_1 = L) = v_2(x_2 = 0) \qquad (11)$$

stating that there should only be a single value for the slope and a single value for the deflection at point B, the following expressions are obtained for the slope *v'* and deflections *v* for the two beam segments AB and BC.

$$\begin{array}{ll} 0 \le x_1 \le L & 0 \le x_2 \le a \\ v_1' = \dfrac{wa^2}{12EIL}(L^2 - 3x_1^2) & v_2' = \dfrac{-w}{6EI}(a^2L + 3a^2x_2 - 3ax_2^2 + x_2^3) \qquad (12) \\ v_1 = \dfrac{wa^2x_1}{12EIL}(L^2 - x_1^2) & v_2 = \dfrac{-wx_2}{24EI}(4a^2L + 6a^2x_2 - 4ax_2^2 + x_2^3) \qquad (13) \end{array}$$

Now that the theoretical formulation of the problem is complete, a MATLAB script file can be created. The script file can be developed in a form which prompts the user to input the values for the parameters *w*, *E*, *I*, *L*, and *a*. Then, a MATLAB loop can be employed to compute the values for the shear, moment, slope, and deflection along the length of the beam for a series of values of x, measured from the left support at A, starting from x = 0 and ending at x = L. Note that it is necessary to include a conditional statement within the loop, so that the proper expressions for the determination of unknowns is selected and used in the computations. The MATLAB script file for the given beam is provided in Figure 6, along with the generated plots in Figure 7. These plots are for the case when w = 0.2 kip/ft, E = 29000 ksi, I = 100 in^4, L = 20 ft, and a = 10 ft. Using the powerful MATLAB plotting commands and tools, the users can control and create the plots in any format they desire.

```
function case_1
% Uniformly distributed load acting on the overhang (created by team 1)
% ________________________________________________________________________________
% Function objective:
% To compute and plot the distribution of the shear force, bending moment, slope, and
% deflection along the length of a given overhanging beam subjected to a Uniformly distributed
% load acting on the overhang.
% ________________________________________________________________________________
% Data acquisition:
w=input('Enter the value of the distributed load(kip/ft):\n');
E=input('Enter the value of the modulus of elasticity(ksi):\n');
I=input('Enter the value of moment of inertia(in^4):\n');
L=input('Enter the value L(ft):\n');
a=input('Enter the value a(ft):\n');
w=w/12;L=L*12;a=a*12;
fprintf(' x(in.)   Shear(kip)   Moment(kip.in)      Slope(rad)       Deflection(in)\n')
% ________________________________________________________________________________
% Computing the shear, moment, slope, and deflection.
x=linspace(0,L+a,(L+a)/6+1);
for k=1:1:(L+a)/6+1
  if x(k)<=L
     x1(k)=x(k);
     V(k)=-w*a^2/(2*L);
     M(k)=-w*a^2*x1(k)/(2*L);
     theta(k)=w*a^2*(L^2-3*x1(k)^2)/(12*E*I*L);
     delta(k)=w*a^2*x1(k)*(L^2-x1(k)^2)/(12*E*I*L);
     fprintf('%4.0f %12.3f %14.3f %19.2e %17.2e\n',x(k),V(k),M(k),theta(k),delta(k))
  else
     x2(k)=x(k)-L;
     V(k)=w*(a-x2(k));
     M(k)=-w*(a-x2(k))^2/2;
     theta(k)=-w*(a^2*L+3*a^2*x2(k)-3*a*x2(k)^2+x2(k)^3)/(6*E*I);
     delta(k)=-w*x2(k)*(4*a^2*L+6*a^2*x2(k)-4*a*x2(k)^2+x2(k)^3)/(24*E*I);
     fprintf('%4.0f %12.3f %14.3f %19.2e %17.2e\n',x(k),V(k),M(k),theta(k),delta(k))
  end
end
% ________________________________________________________________________________
% plotting the shear, moment, slope, and deflection.
subplot(2,2,1),plot(x,V),title('Shear'),xlabel('x (in)'),ylabel('Shear Force (kip)'),...
  axis([0 360 -1 3]),set(gca,'XTick',[0:60:L+a]),text(60,0,'V_1 =-wa^2/(2L)'),...
  text(250,2.2,'V_2 =w(a-x_2)')
subplot(2,2,2),plot(x,theta),title('Slope'),xlabel('x (in)'),ylabel('Slope (rad)'),...
  axis([0 360 -0.006 .004]),set(gca,'XTick',[0:60:L+a]),text(10,-0.003,...
  '\theta_1 =wa^2(L^2-3x_1^2)/(12EIL)'),text(40,0.002,'\theta_2 =-w(a^2L+3a^2x_2-3ax_2^2+x_2^3)/(6EI)')
subplot(2,2,3),plot(x,M),title('Moment'),xlabel('x (in)'),ylabel('Moment (kip.in)'),...
  axis([0 360 -150 50]),set(gca,'XTick',[0:60:L+a]),text(30,-90,'M_1 =-wa^2x_1/(2L)'),...
  text(210,20,'M_2 =-w(a-x_2)^2/2')
subplot(2,2,4),plot(x,delta),title('Deflection'),xlabel('x (in)'),ylabel('Deflection (in)'),...
  axis([0 360 -0.6 0.4]),set(gca,'XTick',[0:60:L+a]),,text(10,-0.1,'v_1 =wa^2x_1(L^2-x_1^2)/(12EIL)'),...
  text(5,0.25,'v_2 =-wx_2(4a^2L+6a^2x_2-4ax_2^2+x_2^3)/(24EI)')
```

Figure 6. The MATLAB Script for the Analysis of a Statically Determinate Beam

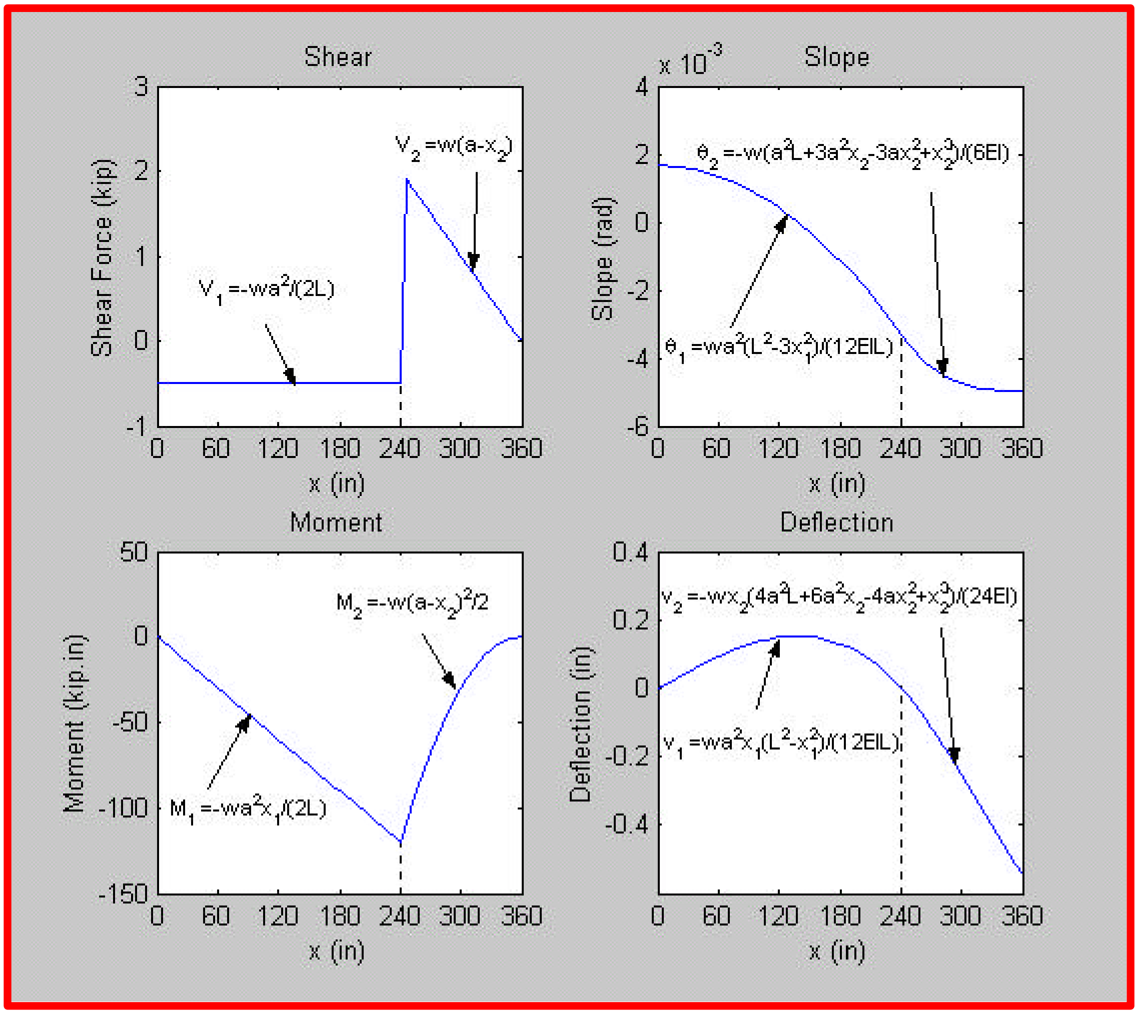
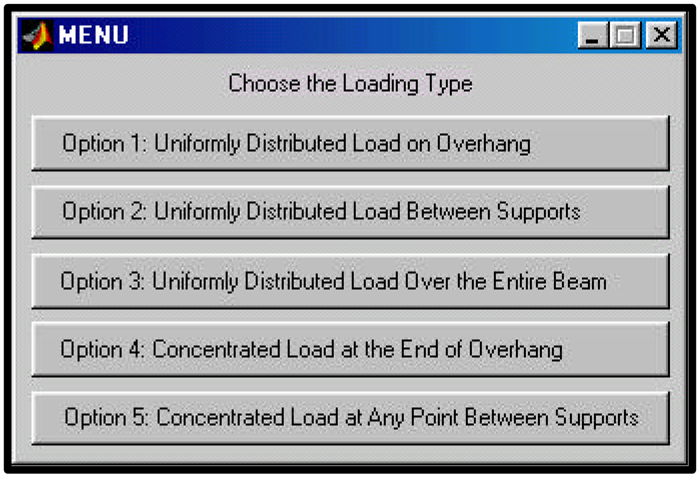


Figure 7. Plot of Shear Force, Bending Moment, Slope, and Deflection

In preparation for teaching the course in the fall semester of 2002, the authors have decided to divide the class into several teams and require each team to provide the solution for a different loading condition acting on the beam. For example, distributed loading over the overhang can be assigned to team 1, distributed loading acting between the supports to team 2, distributed loading over the entire beam to team 3, and so on. When preparing the script files for these problems, the teams are asked to place their scripts in separate “function M-files”, so that these functions can be utilized later by other script files. The reason for this request is that the authors plan to combine the work of all teams into one comprehensive main script, which is capable of making calls to each of the function M-files prepared by the teams. This main script is shown in Figure 8. Note that the script file is prepared in a fashion that when executed it provides the user with a menu to indicate the type of the loading for which the computation is to be performed. This menu is shown in Figure 9. Obviously more loading cases can be included in the formulation of this problem to produce a more powerful script file. It should also be stated that the solution of the problems developed by the teams can as well be combined using the superposition method to generate the results for other combined loading cases.

```
% Analysis of a Statically Determinate Beam.
% ________________________________________________________________________________
% Program objective:

% To compute and plot the distribution of the shear force, bending moment, slope, and
% deflection along the length of a given overhanging beam.
% ________________________________________________________________________________
% Program interactivity:

% Providing the user with a menu so that the desired option can be selected.

k=menu('Choose the Loading Type',...
  'Option 1: Uniformly Distributed Load on Overhang                ',...
  'Option 2: Uniformly Distributed Load Between Supports           ',...
  'Option 3: Uniformly Distributed Load Over the Entire Beam       ',...
  'Option 4: Concentrated Load at the End of Overhang              ',...
  'Option 5: Concentrated Load at Any Point Between Supports');
% ________________________________________________________________________________
% Depending upon the option selected, a call is made to one of the four functions entitled as:
% case_1, case_2, case_3, or case_4.

 if(k==1)
    case_1
    % case_1: Uniformly distributed load acting on the overhang.'
    % (Assigned to team 1)
 elseif(k==2)
    case_2
    % (case_2: Uniformly distributed Load acting between the supports.'
    % (Assigned to team 2)
 elseif(k==3)
    case_3
    % case_3: Uniformly distributed load acting over the entire Beam.'
    % (Assigned to team 3)
 elseif(k==4)
    case_4
    % case_4: Concentrated load acting at the end of the overhang.'
    % (Assigned to team 4)
 else
    case_5
    % case_5: Concentrated load acting at any point between the supports.'
    % (Assigned to team 5)
 end
```

Figure 8. The Main MATLAB Script File Developed for Utilizing the Function M-Files Created by the Teams



MENU

Choose the Loading Type

Option 1: Uniformly Distributed Load on Overhang

Option 2: Uniformly Distributed Load Between Supports

Option 3: Uniformly Distributed Load Over the Entire Beam

Option 4: Concentrated Load at the End of Overhang

Option 5: Concentrated Load at Any Point Between Supports

Figure 9. Snapshot of the Menu Created by the Developed MATLAB Script File

IV. Analysis of Statically Indeterminate Frames (Slope-Deflection Method)

The method of solution for analyzing a statically indeterminate frame using the method of slope-deflection is described in this section of the paper through the following example problem. Suppose that for the frame shown in Figure 10, the following moments are to be computed at the ends of the members AB, BC, and BD (i.e., M_{BA}, M_{BD}, M_{DB}, M_{BC}, and M_{CB}).

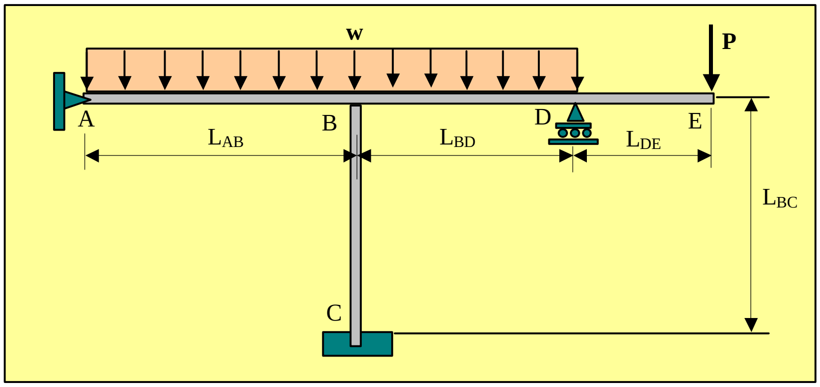


Figure 10. A Statically Indeterminate Frame

Applying the well-known slope-deflection method to members AB, BC, and BD of the frame, the following expressions for the end-moments of the members are obtained.

$$M_{BA} = 3Ek_{AB}q_B + (FEM)_{BA} \qquad (14)$$

$$M_{BD} = 4Ek_{BD}q_B + 2Ek_{BD}q_D + (FEM)_{BD} \tag{15}$$

$$M_{DB} = 4Ek_{BD}q_D + 2Ek_{BD}q_B + (FEM)_{DB} \tag{16}$$

$$M_{BC} = 4Ek_{BC}q_B + (FEM)_{BC} \tag{17}$$

$$M_{CB} = 2Ek_{BC}q_B + (FEM)_{CB} \tag{18}$$

In these expressions, *M*, *E*, *k*, *q* , and *FEM* are respectively, the moment at the end of members, modulus of elasticity, the relative-stiffness factor of members, angular displacements at the end of members, and member fixed-end moments. The fixed-end moments used in the above expressions for the given beam and loading conditions are given by the following equations.

$$(FEM)_{BA} = wL_{AB}^{2}/8 \tag{19}$$

$$(FEM)_{BD} = -wL_{BD}^{2}/12 \tag{20}$$

$$(FEM)_{DB} = wL_{BD}^{2}/12 \tag{21}$$

$$(FEM)_{BC} = 0 \tag{22}$$

$$(FEM)_{CB} = 0 \tag{23}$$

Note that the relative-stiffness of the members in Eqs. (14) through (18) can be computed by dividing the moments of inertia of each member by the corresponding length of the member, i.e.,

$$k_{AB} = I_{AB}/L_{AB}\,,\;\; k_{BD} = I_{BD}/L_{BD}\,,\;\; k_{BC} = I_{BC}/L_{BC} \tag{24}$$

Also, considering the moment equilibrium of joints B and D, the following two equations can be added to the list of the equations already presented.

$$M_{BA} + M_{BC} + M_{BD} = 0 \tag{25}$$

$$M_{DB} + pL_{DE} = 0 \tag{26}$$

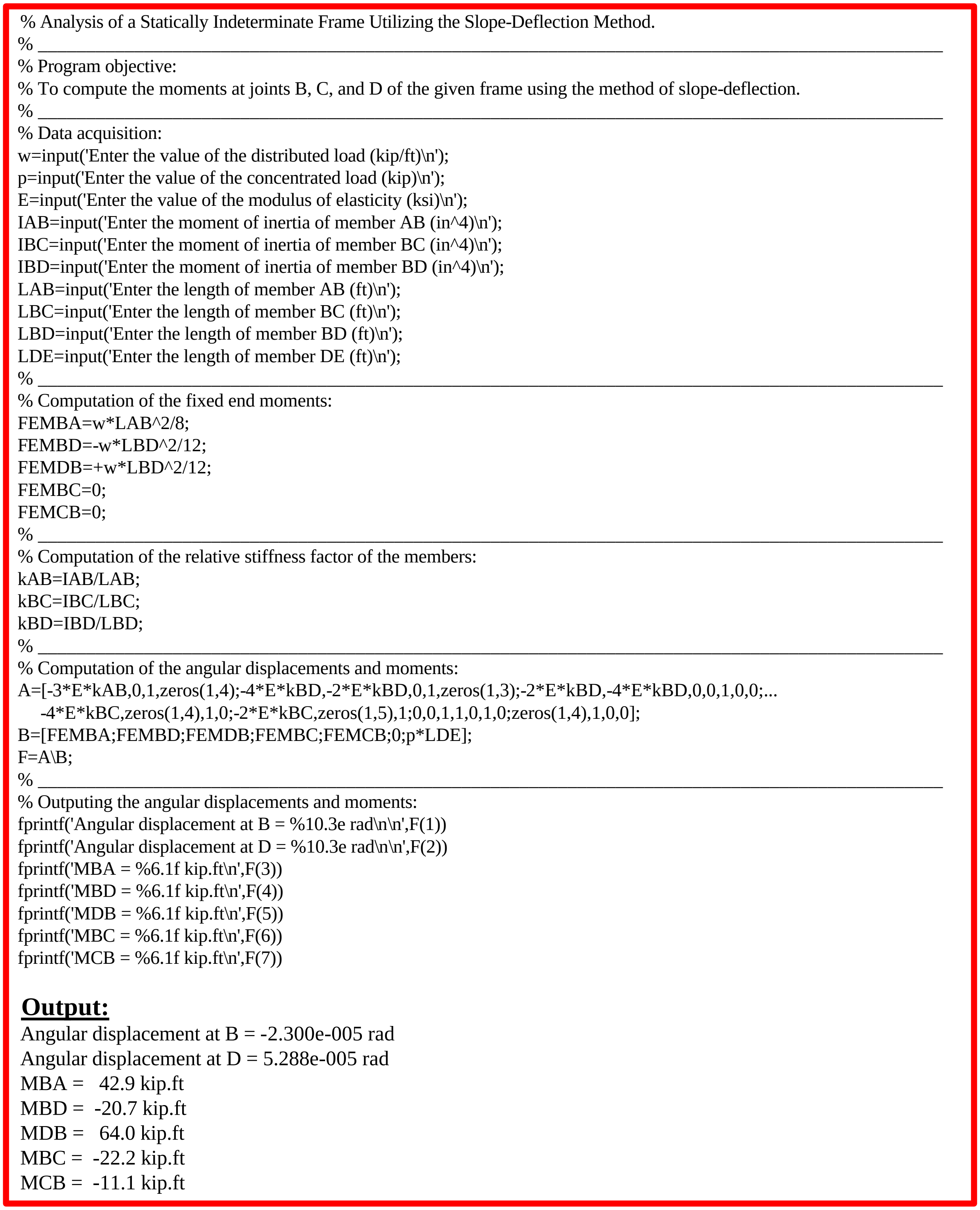
The Eqs. (14) through (18) together with the Eqs. (25) and (26), can be expressed in the matrix format as the following system of linear equations.

$$\begin{bmatrix} -3Ek_{AB} & 0 & 1 & 0 & 0 & 0 & 0 \\ -4Ek_{BD} & -2Ek_{BD} & 0 & 1 & 0 & 0 & 0 \\ -2Ek_{BD} & -4Ek_{BD} & 0 & 0 & 1 & 0 & 0 \\ -4Ek_{BC} & 0 & 0 & 0 & 0 & 1 & 0 \\ -2Ek_{BC} & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{Bmatrix} q_B \\ q_D \\ M_{BA.} \\ M_{BD} \\ M_{DB} \\ M_{BC.} \\ M_{CB} \end{Bmatrix} = \begin{Bmatrix} (FEM)_{BA} \\ (FEM)_{BD} \\ (FEM)_{DB} \\ (FEM)_{BC} \\ (FEM)_{CB} \\ 0 \\ pL_{DE} \end{Bmatrix} \tag{27}$$

The presented solution has the general format of AX=B. Using this expression, the elements of the column vector X, which contains the unknowns in the problem, can be computed. The unknowns in this case consist of the member moments M_{BA}, M_{BD}, M_{DB}, M_{BC}, and M_{CB}, as well as, the angular displacements q_B and q_D. The MATLAB solution for the presented example problem, involves the creation of matrices A and B, and the utilization of the left-division operation as explained earlier in the paper. The MATLAB script file for the presented problem and the corresponding results for the case when w = 2 kip/ft, P = 8 kip, E = 29000 ksi, I = 100 in^4, L_{AB} = 15 ft, L_{BC} = 12 ft, L_{BD} = 12 ft, and L_{DE} = 8 ft are provided in Figure 11.

V. Summary and Conclusion

A procedure for complementing a structural analysis course using the MATLAB software is discussed in this paper. The solutions for three sample commonly encountered problems are formulated using several well-known methods of analysis and discussed to fully describe the procedure. The developed scripts files, similar to the ones presented in the paper, can serve as a valuable educational tool to further enhance the students' understanding of the structural analysis concepts. The students have the opportunity to run the developed script files for different beam and loading conditions, examine and compare the results, and learn more in the process. The students can also use the script files to perform a check on the their developed hand-solutions. The project described in this paper has a number of educational outcomes in terms of student and course development. The exercises promote more interactions among the students, and between the students and the instructor, leading to a better teaching and learning environment. Through this project the students also learn how to use various powerful tools and features of the MATLAB computing software, an experience that will serve the students well in their future academic and professional careers.



```
% Analysis of a Statically Indeterminate Frame Utilizing the Slope-Deflection Method.
% ________________________________________________________________________________
% Program objective:
% To compute the moments at joints B, C, and D of the given frame using the method of slope-deflection.
% ________________________________________________________________________________
% Data acquisition:
w=input('Enter the value of the distributed load (kip/ft)\n');
p=input('Enter the value of the concentrated load (kip)\n');
E=input('Enter the value of the modulus of elasticity (ksi)\n');
IAB=input('Enter the moment of inertia of member AB (in^4)\n');
IBC=input('Enter the moment of inertia of member BC (in^4)\n');
IBD=input('Enter the moment of inertia of member BD (in^4)\n');
LAB=input('Enter the length of member AB (ft)\n');
LBC=input('Enter the length of member BC (ft)\n');
LBD=input('Enter the length of member BD (ft)\n');
LDE=input('Enter the length of member DE (ft)\n');
% ________________________________________________________________________________
% Computation of the fixed end moments:
FEMBA=w*LAB^2/8;
FEMBD=-w*LBD^2/12;
FEMDB=+w*LBD^2/12;
FEMBC=0;
FEMCB=0;
% ________________________________________________________________________________
% Computation of the relative stiffness factor of the members:
kAB=IAB/LAB;
kBC=IBC/LBC;
kBD=IBD/LBD;
% ________________________________________________________________________________
% Computation of the angular displacements and moments:
A=[-3*E*kAB,0,1,zeros(1,4);-4*E*kBD,-2*E*kBD,0,1,zeros(1,3);-2*E*kBD,-4*E*kBD,0,0,1,0,0;...
    -4*E*kBC,zeros(1,4),1,0;-2*E*kBC,zeros(1,5),1;0,0,1,1,0,1,0;zeros(1,4),1,0,0];
B=[FEMBA;FEMBD;FEMDB;FEMBC;FEMCB;0;p*LDE];
F=A\B;
% ________________________________________________________________________________
% Outputing the angular displacements and moments:
fprintf('Angular displacement at B = %10.3e rad\n\n',F(1))
fprintf('Angular displacement at D = %10.3e rad\n\n',F(2))
fprintf('MBA = %6.1f kip.ft\n',F(3))
fprintf('MBD = %6.1f kip.ft\n',F(4))
fprintf('MDB = %6.1f kip.ft\n',F(5))
fprintf('MBC = %6.1f kip.ft\n',F(6))
fprintf('MCB = %6.1f kip.ft\n',F(7))
```

Output:

```
Angular displacement at B = -2.300e-005 rad
Angular displacement at D = 5.288e-005 rad
MBA =   42.9 kip.ft
MBD =  -20.7 kip.ft
MDB =   64.0 kip.ft
MBC =  -22.2 kip.ft
MCB =  -11.1 kip.ft
```

Figure 11. The MATLAB Script File and a Sample Output for the Analysis of a Statically Indeterminate Frame Using the Slope-Deflection Method

Bibliography

1. Hibbeler, R.C. Structural Analysis, Fifth Edition, Prentice Hall, 2002.

SHAHNAM NAVAEE

Shahnam Navaee is currently an Associate Professor in the Engineering Studies Program at Georgia Southern University where his primary responsibility is teaching freshman and sophomore level courses to engineering transfer students. Dr. Navaee received his B.S. and M.S. degrees in Civil Engineering from Louisiana State University in 1980 and 1983 and his Ph.D. degree from the Department of Civil Engineering at Clemson University in 1989.

NIRMAL K. DAS

Nirmal K. Das is an Associate Professor of Civil Engineering Technology at Georgia Southern University, He received his Bachelor of Civil Engineering Degree from Jadavpur University, India, and his M.S. and Ph.D. degrees in Civil Engineering (structures) from Texas Tech University. His areas of interest include structural analysis, structural design, and wind engineering. Dr. Das is a registered professional engineer in Ohio and Georgia.