ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION

Paper ID #35046

Static Finite Element analysis of a truss assembly using MATLAB

Dr. Cyrus K Hagigat, The University of Toledo

Dr. Hagigat is an associate professor in the Engineering Technology department of the College of Engineering of the University of Toledo. Dr. Hagigat has an extensive industrial background, and his teaching technique is based on practical aspects of engineering.

Static Finite Element analysis of a Truss Assembly using MATLAB

Abstract:
The theme of this article is to present an approach for the development of matrix based stress analysis equations for trusses and demonstrating use of MATLAB software to expedite the solution of the matrix based equations. The approach involves developing displacement matrices for connected springs based on their spring constants, and then extending the technique to two dimensional bars by determining their equivalent stiffness through their axial displacement formula. The bar matrices are then adjusted for bar orientation angle and global matrices for trusses are assembled using the stiffness method. Techniques for inclusion of the boundary conditions in the global matrix equations are shown. MATLAB software has extensive matrix manipulation capabilities and features. MATLAB instructions are used to find the displacements for an example truss. The MATLAB produced results are compared against results obtained by classical techniques. The stresses in truss components can then be determined by using the displacements. A discussion regarding the application of the techniques to more complex scenarios where solutions cannot be easily obtained through classical techniques are discussed. The technique has uses as a supplement to traditional material in an intermediate undergraduate structural analysis course. The application of the technique in preparing students for actual use of a commercial Finite Element software such as MATLAB is discussed.

Introduction:

Moderately complex trusses are used in structures such as bridges and aircraft wings. The stress analysis of these trusses are done by the Finite Element method in an industrial setting. However, from an educational point of view, students need to understand the theory behind the finite element formulation in order to be able to use commercially available Finite Element software correctly. Small size Finite Element formulation can be done using hand calculation techniques and these formulations can be solved by hand calculation techniques. However, from an educational point of view, it is beneficial for students to also solve moderately large models without using commercially available software. MATLAB matrix analysis capabilities reduce the level of tedious and time consuming calculations that must be performed for solving a moderate size finite element model that has been formulated in matrix form. This article describes a lab exercise that was used to get the students started using MATLAB for this purpose.

The techniques described in this article were used in an undergraduate structural analysis course and in an undergraduate engineering mathematics course.

In the static stress analysis course the technique was used to gain some insight into methods that commercial finite element software use for solving stress analysis of trusses. In the advanced engineering mathematics course, the technique was used to show how to use matrices for solving engineering problems. In both courses, a component of homework and exams was using MATLAB to solve truss problems.

Technical discussion & numerical example:
Figure 1 shows a deformed spring. [1]

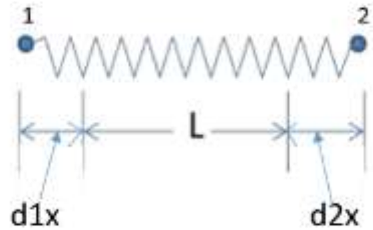

Figure 1: A one dimensional deformed spring

In figure 1, d1x is the displacement at node 1, and d2x is the displacement at node 2 and L is the original length of the spring.

Stiffness matrix for the spring of figure 1 is shown in equation (1). In equation (1), K is the spring stiffness and "[K *spring*]" is the stiffness matrix for the spring where the spring is modeled as a finite element. [1]

$$[K\ spring] = \begin{bmatrix} K & -K \\ -K & K \end{bmatrix} \quad (1)$$

Equation (2) is the force-displacement matrix for spring of figure 1. In equation (2), f1x and f2x are the forces at nodes 1 and 2 respectively.

$$\begin{bmatrix} \text{f1x} \\ \text{f2x} \end{bmatrix} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{bmatrix} d1x \\ d2x \end{bmatrix} \quad (2)$$

There are a number of techniques for assembling a force displacement matrix when springs are attached in series. The superposition technique is presented in this article. The superposition technique is best described by an example.

Consider the spring assemblage shown in figure 2. The nodes in the assemblage of figure 2 are intentionally arbitrarily numbered in order to illustrate the superposition technique. [1]

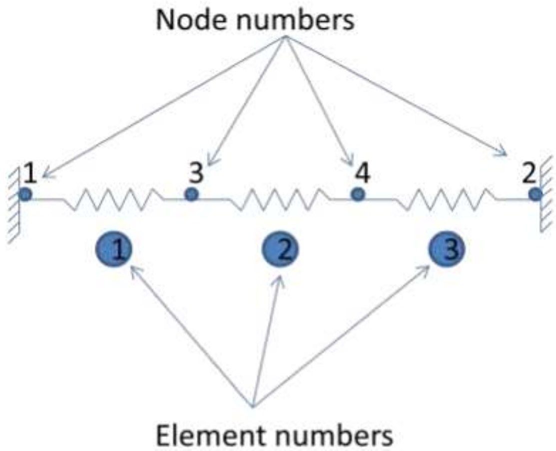

Figure 2: 3 spring elements attached in series

The stiffness matrices for elements 1, 2 and 3 of figure 2 can be determined by using equation (1).

As shown in figure 2, element 1 is between nodes 1 and 3, element 2 is between nodes 3 and 4, and element 3 is between nodes 4 and 2. Based on the location of the nodes, the stiffness matrices for elements 1, 2 and 3 from a node location point of view can be written as shown in equations (3), (4) and (5). [K1], [K2] & [K3] are the element stiffness matrices for elements 1, 2 & 3 of figure 2.

$$[K1] = \begin{bmatrix} (1,1) & (1,3) \\ (3,1) & (3,3) \end{bmatrix} \tag{3}$$

$$[K2] = \begin{bmatrix} (3,3) & (3,4) \\ (4,3) & (4,4) \end{bmatrix} \tag{4}$$

$$[K3] = \begin{bmatrix} (4,4) & (4,2) \\ (2,4) & (2,2) \end{bmatrix} \tag{5}$$

The global stiffness matrix [K *global*] for the assemblage of elements 1, 2 and 3 for figure 2 based on sequential numbering of the nodes can be written as shown in equation (6).

$$[K\ global] = \begin{bmatrix} (1,1) & (1,2) & (1,3) & (1,4) \\ (2,1) & (2,2) & (2,3) & (2,4) \\ (3,1) & (3,2) & (3,3) & (3,4) \\ (4,1) & (4,2) & (4,3) & (4,4) \end{bmatrix} \tag{6}$$

The global stiffness matrix for the assemblage of figure (2) is calculated by equation (7).

$$[K\ global] = [K1] + [K2] + [K3] \tag{7}$$

In equation (7), terms that are missing from a matrix are assigned a value of “0’. Implementing equation (7) for the model of figure 2 results in the global stiffness matrix of equation (8). In equation (8), K1, K2 & K3 are the spring stiffness for elements 1, 2 & 3 of figure 2.

$$[K\ global] = \begin{bmatrix} \mathrm{K1} & 0 & -\mathrm{K1} & 0 \\ 0 & \mathrm{K3} & 0 & -\mathrm{K3} \\ -\mathrm{K1} & 0 & \mathrm{K1}+\mathrm{K2} & -\mathrm{K2} \\ 0 & -\mathrm{K3} & -\mathrm{K2} & \mathrm{K2}+\mathrm{K3} \end{bmatrix} \tag{8}$$

The global stiffness matrix relates global forces to global displacements as shown in equation (9). In equation (9), f1x, f2x, f3x & f4x are the forces at nodes 1 through 4 of figure 2, d1x, d2x, d3x & d4x are the displacements at nodes 1 through 4 of figure 2 and f1x, f2x, f3x & f4x are the forces applied to nodes 1 through 4 of figure 2.

$$\begin{bmatrix} \mathrm{f1x} \\ \mathrm{f2x} \\ \mathrm{f3x} \\ \mathrm{f4x} \end{bmatrix} = \begin{bmatrix} \mathrm{K1} & 0 & -\mathrm{K1} & 0 \\ 0 & \mathrm{K3} & 0 & -\mathrm{K3} \\ -\mathrm{K1} & 0 & \mathrm{K1}+\mathrm{K2} & -\mathrm{K2} \\ 0 & -\mathrm{K3} & -\mathrm{K2} & \mathrm{K2}+\mathrm{K3} \end{bmatrix} \begin{bmatrix} \mathrm{d1x} \\ \mathrm{d2x} \\ \mathrm{d3x} \\ \mathrm{d4x} \end{bmatrix} \tag{9}$$

As shown on figure (2), the displacements at nodes 1 and 2 are 0. Consequently the matrices of equation (9) can be reduced as follows (The lines show the rows and columns that are taken out of the matrix).

$$\begin{bmatrix} \mathrm{f1x} \\ \mathrm{f2x} \\ \mathrm{f3x} \\ \mathrm{f4x} \end{bmatrix} = \begin{bmatrix} \mathrm{K1} & 0 & -\mathrm{K1} & 0 \\ 0 & \mathrm{K3} & 0 & -\mathrm{K3} \\ -\mathrm{K1} & 0 & \mathrm{K1}+\mathrm{K2} & -\mathrm{K2} \\ 0 & -\mathrm{K3} & -\mathrm{K2} & \mathrm{K2}+\mathrm{K3} \end{bmatrix} \begin{bmatrix} \mathrm{d1x} \\ \mathrm{d2x} \\ \mathrm{d3x} \\ \mathrm{d4x} \end{bmatrix}$$

Taking out the deleted rows and columns out of the matrices of equation (9) results in equation (10).

$$\begin{bmatrix} \mathrm{f3x} \\ \mathrm{f4x} \end{bmatrix} = \begin{bmatrix} \mathrm{k1}+\mathrm{k2} & -\mathrm{k2} \\ -\mathrm{k2} & \mathrm{K2}+\mathrm{K3} \end{bmatrix} \begin{bmatrix} \mathrm{d3x} \\ \mathrm{d4x} \end{bmatrix} \tag{10}$$

Assuming the following values for spring stiffness and force values for the model of figure 2 and substituting these values into equation (10) results in equation (11).

K1 = 1000 lbf/inch; K2= 2000 lbf/inch; K3= 3000 lbf/inch; f4x = 5000 lbs

$$\begin{bmatrix} 0 \\ 5000 \end{bmatrix} = \begin{bmatrix} 3000 & -2000 \\ -2000 & 5000 \end{bmatrix} \begin{bmatrix} \mathrm{d3x} \\ \mathrm{d4x} \end{bmatrix} \tag{11}$$

A solution of the matrix of equation (11) results in the following values for d3x and d4x.

d3x= 10/11 inch;   d4x= 15/11 inch.

Substituting the values of 0 for d1x and d2x (because these nodes are fixed), and substituting the calculated values for d3x and d4x into equation (9) results in equation (12).

$$\begin{bmatrix} f1x \\ f2x \\ f3x \\ f4x \end{bmatrix} = \begin{bmatrix} 1000 & 0 & -1000 & 0 \\ 0 & 3000 & 0 & -3000 \\ -1000 & 0 & 3000 & -2000 \\ 0 & -3000 & -2000 & 5000 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 10/11 \\ 15/11 \end{bmatrix} \tag{12}$$

A solution of equation (12) (obtained by hand calculation or by using MATLAB) results in the following values for the forces at nodes 1 through 4 of the finite element of figure 2.

f1x = -10000/11 lb. ; f2x=-45000/11 lb. ; f3x= 0 ; f4x= 55000/11 lb.

The calculation of forces of equation (12) is possible by classical hand calculation techniques because the matrix sizes are small. For larger systems, hand calculation techniques are not practical, and MATLAB can be used to perform the calculations.

A bar can be modeled similar to a spring by using standard bar deflection formulas. Figure 3 illustrates the concept.

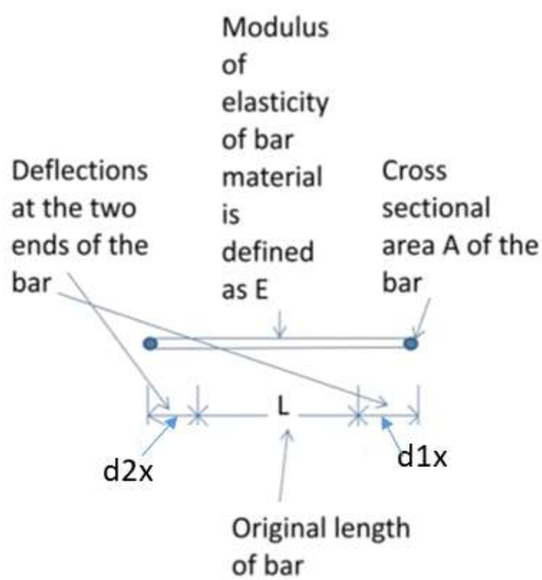

Figure 3: A one dimensional deformed bar

Since the bar element is developed by using the same deflection technique as a spring, assemblage of a number of finite elements that are based on bar formulation is done by the same technique that is used for assembling a number of spring elements.

Bar elements are used for modeling truss assemblies. In a truss, various truss elements can form an angle with the global coordinates as shown in figure 4.

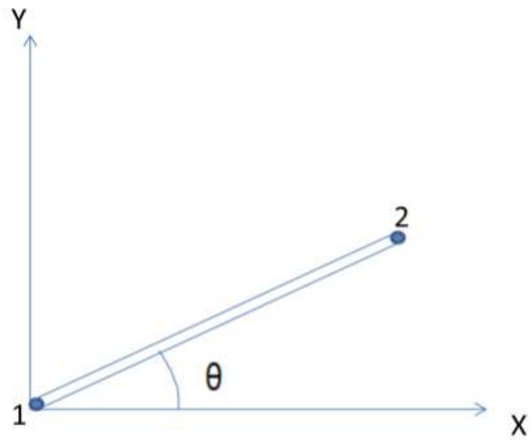

Figure 4: A bar element making an angle θ with X axis
of global coordinate system

The global stiffness matrix relating global forces to global displacements for the element shown in figure 5 is given in equation (13). [3]

$$\begin{bmatrix} f1x \\ f1y \\ f2x \\ f2y \end{bmatrix} \begin{bmatrix} C*C & C*S & -C*C & -C*S \\ C*S & S*S & -C*S & -S*S \\ -C*C & -C*S & C*C & C*S \\ -C*S & -S*S & C*S & S*S \end{bmatrix} \begin{bmatrix} d1x \\ d1y \\ d2x \\ d2y \end{bmatrix} [AE/L] \qquad (13)$$

In equation (13), C is COS θ, S is SIN θ, f are forces and numbers and letters after f are showing node number and force direction, and d is displacement and numbers and letters after d are showing node number and deflection direction.

The same assembly technique that was illustrated for the finite element of figure 2 (demonstrated in equation 9) can be used to assemble a finite element consisting of bar elements.

Figure 5 is the finite element model of a truss. Equation (14) is the formulation of the truss of figure 5 analogous to equation (11).

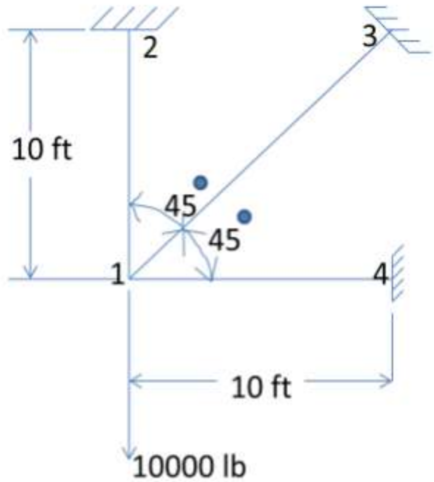

Figure 5: Finite Element model of a truss

$$\begin{bmatrix} 0 \\ -10000 \end{bmatrix} = \begin{bmatrix} 1.354 & 0.354 \\ 0.354 & 1.354 \end{bmatrix} \begin{bmatrix} d1x \\ d1y \end{bmatrix} \tag{14}$$

Equation (14) can be written in a compact matrix form as shown in equation (15).

$$[f] = [K][d] \tag{15}$$

Both sides of equation (15) can be multiplied by the inverse of matrix [K]. This results in equation (16). [4]

$$[d] = [K]^{-1}[f] \tag{16}$$

MATLAB can be used to find the displacements of equation (16). The stresses can then be calculated from the displacements.

In summary, the finite element solution of a truss using MATLAB matrix operations consist of the following steps.

1. Develop the finite element formulation of truss elements using spring analogy.
2. Use matrices to develop the finite element matrix equations relating forces, displacements and stiffness values of the truss elements.
3. Use MATLAB matrix operations to solve the system equations.

Educational value of the technique:
The example presented in this article was one of the earlier assignments in an undergraduate structural analysis course. The article set the stage for more complicated assignments that were solved by the semi-automated technique described.

After most students were comfortable with the semi-automated technique, the actual Finite-Element modeling module of MATLAB (not described in this article) was used to analyze moderately complex trusses representing bridges and aircraft wings.

End of the semester comments by students indicated that the students had a better understanding when using a commercial finite element software (Finite Element module of MATLAB) because they had gone through the process of manually setting up Finite Element equations and solving them.

Summary & Conclusion:
In this article the Finite Element formulation of trusses were demonstrated by using an analogy with spring deflections. An example of solving a simple truss Finite Element formulation by semi-manual techniques using MATLAB matrix capabilities was presented. The technique helped the students to better understand the Finite Element capabilities of MATLAB where the entire process is automated and handled internally in the MATLAB software.

End of the semester student feedback were mostly positive. Students indicated that the technique when combined with hand calculation techniques (classical techniques) enabled them to better understand both techniques. Some unique issues regarding take home exams were encountered as a result of the introduction of the completely automated Finite Element capabilities of MATLAB. The exam questions had asked for a complete classical technique solution. Some students used the Finite Element technique and provided the correct answers without doing the hand calculation solutions. These cases were not treated as academic dishonesty cases, and the students were given a chance to do the hand calculation solutions and turn them in.

The techniques presented here can be expanded and be included as a supplement for a vibration analysis course. MATLAB (which stands for Matrix Laboratory) has extensive matrix analysis capabilities and the basis of vibration analysis is matrix algebra. The inclusion of MATLAB in an undergraduate course has the potential to get the students interested in more advanced finite element software such as NASTRAN/PATRAN and ANSYS. The author has developed a graduate level stress analysis course using NASTRAN/PATRAN and a second graduate level course using NASTRAN/PATRAN is under development. The subject of the second graduate level course is advanced vibration analysis.

References:
[1] Chapter 2 of “A First Course in the Finite Element Method” by Daryl L. Logan.
[2] Chapter 4 of “Mechanics of Materials; Fourth edition by Higdon, Ohlsen; Stiles; Weese & Riley).
[3] Chapter 3 of “A First Course in the Finite Element Method” by Daryl L. Logan.
[4] Numerical methods in engineering practice by “Amir Wadi Al-Khafaji & John R. Tooley”.