APPENDIX B

MATRIX NOTATION

The Definition of Matrix Notation is the Definition of Matrix Multiplication

B.1 INTRODUCTION

{ XE "Matrix Multiplication" }{ XE "Matrix Notation" }The use of matrix notations is not necessary to solve problems in the static and dynamic analysis of complex structural systems. However, it does allow engineers to write the fundamental equation of mechanics in a compact form. In addition, it produces equations in a form that can be easily programmed for digital computers. Also, it allows the properties of the structure to be separated from the loading. Therefore, dynamic analysis of structures is a simple extension of static analysis.

To understand and use matrix notation, it is not necessary to remember mathematical laws and theorems. Every term in a matrix has a physical meaning, such as *force per unit of displacement*. Many structural analysis textbooks present the traditional techniques of structural analysis without the use of matrix notation; then, near the end of the book *MATRIX METHODS* are presented as a different method of structural analysis. The fundamental equations of equilibrium, compatibility and material properties, when written using matrix notation, are not different from those used in traditional structural analysis. Therefore, in my opinion, the terminology *matrix methods of structural analysis* should never be used.

B.2 DEFINITION OF MATRIX NOTATION

To clearly illustrate the application of matrix notation, let us consider the joint equilibrium of the simple truss structure shown in Figure B.1.

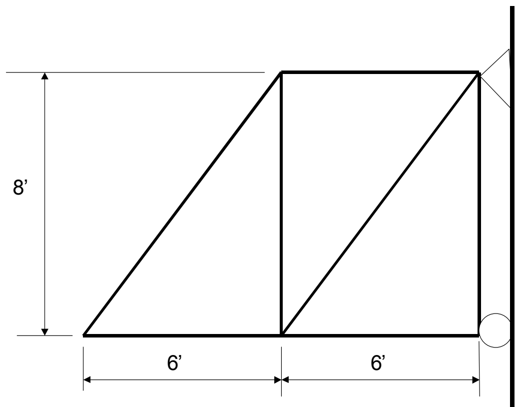

Figure B.1 Simple Truss Structure

Positive external node loads and node displacements, shown in Figure B.2, are in the direction of the x and y reference axes. Axial forces f_i and deformations d_i are positive if tension is produced in the member.

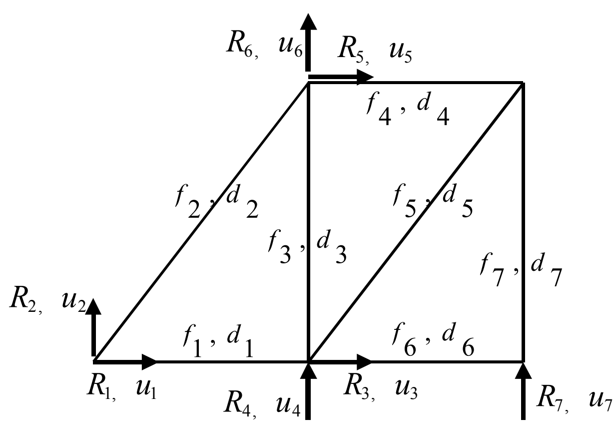

Figure B.2 Definition of Positive Joint Forces and Node Displacements

For the truss structure shown in Figure B.1, the joint equilibrium equations for the sign convention shown are:

$$R_1 = -f_1 - 0.6 f_2 \tag{B.1a}$$

$$R_2 = -0.8 f_2 \tag{B.1b}$$

$$R_3 = f_1 - 0.6 f_5 - f_6 \tag{B.1c}$$

$$R_4 = -f_3 - 0.8 f_5 \tag{B.1d}$$

$$R_5 = 0.6 f_2 - f_4 \tag{B.1e}$$

$$R_6 = 0.8 f_2 + f_3 \tag{B1.f}$$

$$R_7 = -f_7 \tag{B.1g}$$

We can write these seven equilibrium equations in matrix form where each row is one joint equilibrium equation. The resulting matrix equation is

$$\begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \\ R_6 \\ R_7 \end{bmatrix} = \begin{bmatrix} -1.0 & -0.6 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.8 & 0 & 0 & -0.6 & 1.0 & 0 \\ 1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.0 & 0 & -0.8 & 0 & 0 \\ 0 & 0.6 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0.8 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1.0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix} \tag{B.2}$$

Or, symbolically

$$\mathbf{R} = \mathbf{Af} \tag{B.3}$$

Now, if two load conditions exist, the 14 equilibrium equations can be wrtten as one matrix equation in the following form:

$$\begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \\ R_{31} & R_{32} \\ R_{41} & R_{42} \\ R_{51} & R_{52} \\ R_{61} & R_{62} \\ R_{71} & R_{72} \end{bmatrix} = \begin{bmatrix} -1.0 & -0.6 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.8 & 0 & 0 & -0.6 & 1.0 & 0 \\ 1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.0 & 0 & -0.8 & 0 & 0 \\ 0 & 0.6 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0.8 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1.0 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \\ f_{31} & f_{32} \\ f_{41} & f_{42} \\ f_{51} & f_{52} \\ f_{61} & f_{62} \\ f_{71} & f_{72} \end{bmatrix} \quad \text{(B.4)}$$

It is evident that one can extract the 14 equilibrium equations from the one matrix equation. Also, it is apparent that the definition of matrix notation can be written as:

$$f_{il} = \sum_{k=1,7} A_{ik} R_{kl} \quad \text{(B.5)}$$

Equation (B.5) is also the definition of matrix multiplication. Note that we have defined that each load is factored to the right and stored as a column in the load matrix. Therefore, there is no need to state the matrix analysis theorem that:

$$\mathbf{Af} \neq \mathbf{fA} \quad \text{(B.6)}$$

Interchanging the order of matrix multiplication indicates that one does not understand the basic definition of matrix notation.

B.3 MATRIX TRANSPOSE AND SCALAR MULTIPLICATION

{ XE "Matrix Transpose" }Referring to Figure B.1, the energy, or work, supplied to the structure is given by:

$$W = \frac{1}{2} \sum_{i=1,7} R_i u_i \quad \text{(B.7)}$$

We have defined:

$$\mathbf{R} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \\ R_6 \\ R_7 \end{bmatrix} \quad \text{and} \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} \tag{B.8a and B.8b}$$

The definition of the transpose of a matrix is "the columns of the original matrix are stored as rows in the transposed matrix." Therefore:

$$\mathbf{R}^{\mathsf{T}} = \begin{bmatrix} R_1 & R_2 & R_3 & R_4 & R_5 & R_6 & R_7 \end{bmatrix} \tag{B.9a}$$

$$\mathbf{u}^{\mathsf{T}} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \end{bmatrix} \tag{B.9b}$$

It is now possible to express the external work, Equation (B.7), as the following matrix equation:

$$W = \frac{1}{2}\mathbf{R}^{\mathsf{T}}\mathbf{u} \text{ or } W = \frac{1}{2}\mathbf{u}^{\mathsf{T}}\mathbf{R} \tag{B.10}$$

Also, the internal strain energy Ω, stored in the truss members, is defined by the following:

$$\Omega = \frac{1}{2}\mathbf{f}^{\mathsf{T}}\mathbf{d} \text{ or } \Omega = \frac{1}{2}\mathbf{d}^{\mathsf{T}}\mathbf{f} \tag{B.11}$$

Therefore, the purpose of the transpose notation is to use a matrix that has been defined column-wise as a matrix that has been defined row-wise. The major use of the notation, in structural analysis, is to define work and energy. Note that the scalar 1/2 has been factored out of the equations and is applied to each term in the transposed matrix.

From the above example, it is apparent that if:

$$\mathbf{A} = \mathbf{BC} \text{ then } \mathbf{A}^{\mathsf{T}} = \mathbf{C}^{\mathsf{T}}\mathbf{B}^{\mathsf{T}} \tag{B.12}$$

It important to point out that within a computer program, it is not necessary to create a new transformed matrix within the computer storage. One can use the data from the original matrix by interchanging the subscripts.

B.4 DEFINITION OF A NUMERICAL OPERATION

{ XE "Numerical Operation, Definition" }One of the most significant advantages of using a digital computer is that one can predict the time that is required to perform various numerical operations. It requires computer time to move and store numbers and perform floating-point arithmetic, such as addition and multiplication. Within a structural analysis program, a typically arithmetic statement is of the following form:

$$\mathbf{A = B + C \times D} \tag{B.13}$$

The execution of this statement involves removing three numbers from storage, one multiplication, one addition, and then moving the results back in high-speed storage. Rather than obtaining the time required for each phase of the execution of the statement, it has been found to be convenient and accurate to simply define the evaluation of this statement as ***one numerical operation***. In general, the number of operations per second a computer can perform is directly proportional to the clock-speed of the computer. For example, for a 150 MHz Pentium, using Microsoft Power FORTRAN, it is possible to perform approximately ***6,000,000 numerical operations each second***.

B.5 PROGRAMMING MATRIX MULTIPLICATION

Programming matrix operations is very simple. For example, the FORTRAN-90 statements required to multiply the N-by-M-matrix-**A** by the M-by-L-matrix-**B** to form the N-by-L-matrix-**C** are given by:

```
C = 0.0
DO   I=1,N
  DO   J=1,L
    DO K=1,M
   C(I,J) =  C(I,J) + A(I,K)*B(K,L)
    ENDDO       ! end K do loop
```

```
    ENDDO        ! end J  do loop
   ENDDO         ! end I  do loop
```

Note that the number of times the basic arithmetic statement is executed is the product of the limits of the DO LOOPS. Therefore, the number of numerical operations required to multiply two matrices is:

$$Nop = NML \ , \ \text{ or, if } \ M = L = N, \ \text{ then } \ Nop = N^3 \tag{B.14}$$

It will later be shown that this is a large number of numerical operations compared to the solution of a set of N linear equations.

B.6 ORDER OF MATRIX MULTIPLICATION

Consider the case of a statically determinate structure where the joint displacements can be calculated using the following matrix equation:

$$\mathbf{u} = \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{R} = [[\mathbf{A}^T \mathbf{C}]\mathbf{A}]\mathbf{R} = \mathbf{A}^T [\mathbf{C}[\mathbf{A} \mathbf{R}]] \tag{B.15}$$

If A and B are N by N matrices and R is an N by 1 matrix, the order in which the matrix multiplication is conducted is very important. If the evaluation is conducted from left to right, the total number of numerical operations is $2N^3 + N^2$. On the other hand, if the matrix equation is evaluated from right to left, the total number of numerical operations is $3N^2$. This is very important for large matrices such as those encountered in the dynamic response of structural systems.

B.7 SUMMARY

Matrix notation, as used in structural analysis, is very logical and simple. There is no need to remember abstract mathematical theorems to use the notation. The mathematical properties of matrices are of academic interest; however, it is far more important to understand the physical significance of each term within every matrix equation used in structural analysis.

There is no need to create a transpose of a matrix within a computer program. Because no new information is created, it is only necessary to interchange the subscripts to access the information in transposed form. There are a large number of computational techniques that exploit symmetry, sparseness and compact storage and eliminate the direct storage of large rectangular matrices.

Different methods of structural analysis can be evaluated by comparing the number of numerical operations. However, very few modern research papers on structural analysis use this approach. There is a tendency of many researchers to make outrageous claims of numerical efficiency without an accurate scientific evaluation of computer effort required by their proposed *new* method.