# Matrix Algebra Review

## 1.0 Matrix Multiplication

Matrix multiplication is a relatively simple operation where the rows of the first matrix are multiplied times the columns of the second matrix. It can be formally defined by letting A be an m (rows) by n (columns) matrix and B an n by p matrix. The product AB is an m by p matrix defined by:

$$C_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} \qquad \textit{where} \;\; i = 1, 2, 3, \ldots, m \qquad j = 1, 2, 3, \ldots, p \tag{1.1}$$

It is very important to note that the product of two matrices is defined only when the columns in the first matrix is equal to the number of rows in the second matrix.

We can illustrate this with two 3 x 3 matrices. The results are shown below.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \times \begin{bmatrix} j & k & l \\ m & n & o \\ p & q & r \end{bmatrix} = \begin{bmatrix} aj+bm+cp & ak+bn+cq & al+bo+cr \\ dj+em+fp & dk+en+fq & dl+eo+fr \\ gj+hm+ip & gk+hn+iq & gl+ho+ir \end{bmatrix} \tag{1.2}$$

In matrix notation we can write:

$$[A]\times[B]=[C] \tag{1.3}$$

which is an equivalent statement. Note that in general, matrix multiplication is not commutative so that:

$$[B]\times[A]\neq[C]. \tag{1.4}$$

You can easily prove this by multiplying matrix [B] times [A]. The product will not be equal to [C]. For the two matrices to be equal, each term of the two matrices must be equal.

We can illustrate this with MATLAB or OCTAVE[1] by defining two matrices `A` and `B` then multiplying them together. Let

```
A=[1,2,3;4,5,6;7,8,9]                                  (1.5)
    1     2     3
    4     5     6
    7     8     9
```

```
B=[1,3,5;9,7,5;2,6,4]                                  (1.6)
    1     3     5
    9     7     5
    2     6     4
```

[1] OCTAVE is an open source program that is almost identical to MATLAB. You can download if for free from http://octave.sourceforge.net/ . Documentation can be found at http://www.gnu.org/software/octave/doc/interpreter/ .

```
A*B                                   (1.7)
    25   35   27
    61   83   69
    97   131  111
```

We can reverse the order the matrices and show that order is important.

```
B*A                                   (1.8)
    48   57   66
    72   93   114
    54   66   78
```

It is very obvious from the two MATLAB/OCTAVE examples above that the relationship shown in equation 1.4 is correct.

## 1.1 Matrix Transpose

A matrix transpose is created by swapping the rows and columns of a matrix. This is shown below.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^T = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix} \tag{1.9}$$

The superscript T indicates the transpose operation.

With MATLAB/OCTAVE, the transpose of a matrix is very easy to create by adding an apostrophe after the name of the matrix. We can illustrate this with the matrix defined in 1.5.

```
A′                                    (1.10)
    1    4    7
    2    5    8
    3    6    9
```

## 1.2 Identity Matrix

The identity matrix is a special matrix composed on 1s on the diagonal and 0s everywhere else.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.11}$$

The identity matrix has a property similar to the scalar number 1. Any matrix multiplied by the identity matrix is equal to itself. This is shown below. [I] is the identity matrix and [M] is any other matrix.

$$[I]\times[M]=[M] \tag{1.12}$$

or

$$[M]\times[I]=[M] \tag{1.13}$$

```
I=[1,0,0;0,1,0;0,0,1]                                    (1.14)
    1    0    0
    0    1    0
    0    0    1

A*I                                                      (1.15)
    1    2    3
    4    5    6
    7    8    9

I*A                                                      (1.16)
    1    2    3
    4    5    6
    7    8    9
```

The identity matrix is a special matrix and the order of multiplication does not matter as shown above.

## 1.3 Matrix Inverse

If the matrix [B] is the inverse of [A] then:

$$[B]\times[A]=[I] \tag{1.17}$$

Another way of stating this is:

$$[A]^{-1}\times[A]=[I] \tag{1.18}$$

The -1 superscript indicates the inverse of a matrix.

The inverse of a matrix cannot be easily created with simple row column operations as could the transpose of a matrix but it does have important uses. In solving problems that can be represented mathematically as shown in equation (1.19)

$$[M]\times\{F\}=\{E\} \tag{1.19}$$

We can multiply both sides by the inverse of [M]

$$[M]^{-1}\times[M]\times\{F\}=[M]^{-1}\times\{E\} \tag{1.20}$$

or

$$\{F\}=[M]^{-1}\times\{E\} \tag{1.21}$$

which simplifies solving the equations to a simple matrix – vector multiplication. This can be a useful technique if the forces are needed for many different load cases.

We can illustrate this with either MATLAB or OCTAVE. Here the inverse is computed with the built in function *inv()*.

```
inv(B)                                                        (1.22)
    -0.01667  0.15000   -0.16667
    -0.21667  -0.05000  0.33333
    0.33333   0.00000   -0.16667

B*inv(B)                                                      (1.23)
    1.00000   0.00000   -0.00000
    -0.00000  1.00000   0.00000
    0.00000   0.00000   1.00000

inv(B)*B                                                      (1.24)
    1.00000   0.00000   0.00000
    0.00000   1.00000   0.00000
    0.00000   0.00000   1.00000
```

The two answers are very close but not exactly the same. The difference is caused by the computer truncating decimal digits.

## 1.4 Row and Column Vectors

Matrices with a single column or row are called vectors. If they have only one column they are called a column vector and if they only have one row, they are called a row vector.

$$\text{Column Vector} \longrightarrow \begin{Bmatrix} a \\ b \\ c \end{Bmatrix} \quad \begin{Bmatrix} a & b & c \end{Bmatrix} \longleftarrow \text{Row Vector} \tag{1.25}$$

You will frequently see a column vector written as $\{a \;\; b \;\; c\}^T$ where the transpose of a row vector is a column vector.

## 1.5 Vector – Vector Multiplication

Vectors are multiplied using equation 1.1 just like any other matrix. This is shown in the following example.

$$\begin{Bmatrix} a & b & c \end{Bmatrix} \times \begin{Bmatrix} d \\ e \\ f \end{Bmatrix} = ad + be + cf \tag{1.26}$$

In this case, we are multiplying a row vector times a column vector. The results are a scalar quantity. You cannot multiply two row or two column vectors. Remember, the number of columns in the first matrix must equal the number of rows in the second matrix.

We can illustrate vector multiplication with MATLAB by defining two vectors. First we define a row vector

```
D=[9, 6, 2]                                                        (1.27)
    9    6    2
```

Next we define a column vector. We do this by separating the elements with semi-colons.

```
E=[5; 4; 3]                                                        (1.28)
    5
    4
    3
```

Multiplying

```
D*E                                                                (1.29)
    75
```

In this case, the results of the multiplication are the scalar number 75. If we change the order of the vectors we will get a completely different result.

```
E*D                                                                (1.30)
    45   30   10
    36   24   8
    27   18   6
```

In this case, the result is a matrix. Here again, you can duplicate these results using equation 1.1.

## 1.6 Matrix Vector Multiplication

A vector can be multiplied by a matrix. The result is a vector as shown below.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \times \begin{Bmatrix} j \\ k \\ l \end{Bmatrix} = \begin{Bmatrix} aj + bk + cl \\ dj + ek + fl \\ gj + hk + il \end{Bmatrix} \tag{1.31}$$

We can illustrate this with MATLAB/OCTIVE by multiplying the matrix A times the vector E defined above in equations 1.5 and 1.28 respectively.

```
A*E                                                                (1.32)
    22
    58
    94
```

Here the result is a column vector. We can also multiply a vector times a matrix. This is illustrated by multiplying the transpose of E times A. Note that E is a column vector and the transpose of E is a row vector.

```
E'*A                                                               (1.33)
    42   54   66
```

## 1.7 Gaussian Elimination

Many problems in engineering require solving large systems of linear equations. This is especially true with finite element solutions to stress and heat transfer problems. In finite element work, it is not uncommon to have problems with thousands of equations.

There are several methods for solving systems of equations with and some methods being better than others. The two major requirements are that the method is very fast and can be used on a wide range of problems. Gaussian elimination meets these requirements. It has the following attributes:

a. It works for most reasonable problems
b. It is computationally very fast
c. It is not to difficult to program

The major drawback is that it can suffer from the accumulation of round off errors.

## 1.8 Upper Triangular Form (Forward Sweep)

We can illustrate how it works with the following set of equations.

$$\begin{aligned} a_{11}X_1 + a_{12}X_2 + a_{13}X_3 &= d_1 \\ a_{21}X_1 + a_{22}X_2 + a_{23}X_3 &= d_2 \\ a_{31}X_1 + a_{32}X_2 + a_{33}X_3 &= d_3 \end{aligned} \tag{1.34}$$

We would like to solve this set of equations for $X_1$ , $X_2$ , and $X_3$ . We can rewrite the equations in matrix form as:

$$\left.\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & d_1 \\ a_{21} & a_{22} & a_{23} & d_2 \\ a_{31} & a_{32} & a_{33} & d_3 \end{array}\right. \tag{1.35}$$

We want to multiply the first equation by some factor so that when we subtract the second equations the $a_{21}$ is eliminated. We can do this by multiplying it by $a_{21} / a_{11}$ . This yields:

$$\left.\begin{array}{ccc|c} a_{21} & \dfrac{a_{12}a_{21}}{a_{11}} & \dfrac{a_{13}a_{21}}{a_{11}} & \dfrac{a_{21}d_1}{a_{11}} \\ a_{21} & a_{22} & a_{23} & d_2 \\ a_{31} & a_{32} & a_{33} & d_3 \end{array}\right. \tag{1.36}$$

Subtracting the first equation in 3.3 from the second equation in 3.3 then replacing the first equation with its original form yields:

$$\left.\begin{matrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}-\dfrac{a_{12}a_{21}}{a_{11}} & a_{23}-\dfrac{a_{13}a_{21}}{a_{11}} \\ a_{31} & a_{32} & a_{33} \end{matrix}\right| \begin{matrix} d_1 \\ d_2-\dfrac{a_{21}d_1}{a_{11}} \\ d_3 \end{matrix} \qquad (1.37)$$

We do the same thing for the third row by multiplying the first row by $a_{31} / a_{11}$ and subtracting it from the third row.

$$\left.\begin{matrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}-\dfrac{a_{12}a_{21}}{a_{11}} & a_{23}-\dfrac{a_{13}a_{21}}{a_{11}} \\ 0 & a_{32}-\dfrac{a_{13}a_{31}}{a_{11}} & a_{33}-\dfrac{a_{13}a_{31}}{a_{11}} \end{matrix}\right| \begin{matrix} d_1 \\ d_2-\dfrac{a_{21}d_1}{a_{11}} \\ d_3-\dfrac{a_{31}d_1}{a_{11}} \end{matrix} \qquad (1.38)$$

We can reduce the complexity of the terms symbolically by substituting new variable names for the complex terms. This yields

$$\left.\begin{matrix} a_{11} & a_{12} & a_{13} \\ 0 & b_{22} & b_{23} \\ 0 & b_{32} & b_{33} \end{matrix}\right| \begin{matrix} d_1 \\ e_2 \\ e_3 \end{matrix} \qquad (1.39)$$

Now multiply the second row by $b_{32}/b_{22}$ and subtracting it from the third row.

$$\left.\begin{matrix} a_{11} & a_{12} & a_{13} \\ 0 & b_{22} & b_{23} \\ 0 & 0 & b_{33}-\dfrac{b_{23}b_{32}}{b_{22}} \end{matrix}\right| \begin{matrix} d_1 \\ e_2 \\ e_3-\dfrac{b_{32}e_2}{b_{22}} \end{matrix} \qquad (1.40)$$

Again we simplify by substituting in for the complex terms.

$$\left.\begin{matrix} a_{11} & a_{12} & a_{13} \\ 0 & b_{22} & b_{23} \\ 0 & 0 & c_{33} \end{matrix}\right| \begin{matrix} d_1 \\ e_2 \\ f_3 \end{matrix} \qquad (1.41)$$

This is what we call upper triangular form. We started at the first equation and worked our way to the last equation in what is called a forward sweep. The forward sweep results in a matrix with values on the diagonal and above and zeros below the diagonal.

## 1.9 Backward Sweep

We can solve each equation of the upper triangular form by moving through the list starting with the bottom equation and working our way up to the top.

$$X_3 = f_3 / c_{33} \tag{1.42}$$
$$X_2 = (e_2 - b_{22} X_3) / b_{22} \tag{1.43}$$
$$X_1 = (d_1 - a_{12} X_2 - a_{13} X_3) / a_{11} \tag{1.44}$$

This is called back substitution and the process as a whole is called the backward sweep.

## 1.10 Equation Normalization

You can see from the equations that it is very important for the diagonal coefficients to be non-zero. A zero valued diagonal term will lead to a divide by zero error.

In fact, there will be fewer problems with accumulated round off error if the diagonal coefficients have a larger magnitude than the off diagonal terms. The reason for this is that computers have limited precision. If the off diagonal terms are much larger than the diagonal term, a very large number will be created when the right-hand-side of the equation is divided by the diagonal term. The size of this number dominates the precision of the computer and causing smaller values to be ignored when added to the larger value. For example, adding $1.3 \times 10^{11}$ to 94.6 in a computer that has 8 digits of precision does results in a value of $1.3 \times 10^{11}$. The 94.6 is completely lost in the process. This can be seen in the previous example

$$a_{22} - \frac{a_{12} a_{21}}{a_{11}} \tag{1.45}$$

If the diagonal term $a_{11}$ is small compared to the diagonal term $a_{22}$ then we could end up with

Small – large

And loose the value of small ($a_{22}$) altogether.

This problem is solved by using two steps in the processing.

a. Normalize the rows so that the largest term in the equation is 1.
b. Swap the rows around so that the largest term in any equation occurs on the diagonal. This is called partial pivoting.

Diagonal dominance can usually be improved by moving the rows or the columns in the set of equations. Moving the columns is somewhat more difficult than moving the rows because the position of $X_1$, $X_2$, … changes when you move the columns. If you move the columns you must keep track of the change in position of the Xs. When the rows are moved, the Xs stay in the same place so rows can be moved without the added complication.

Moving both rows and columns is called full pivoting and it is the best method for achieving diagonal dominance. We are going to look at partial pivoting because it works for many problems and is much simpler to implement. With partial pivoting, we are only going to move the rows.

## 1.11 Row Normalization

Before we can perform either full or partial pivoting, we must normalize the rows. The process is illustrated in the following example. We start with the equations

$$\left.\begin{array}{ccc|c} 0.5 & -8.9 & 3.2 & 7.1 \\ 4.9 & -7.2 & 22.3 & 4.2 \\ 0.2 & -.03 & .06 & 2.2 \end{array}\right. \qquad (1.46)$$

We normalize each row of the equations by dividing through by the largest term in magnitude in the row of the matrix. The values in the vector at the right are not considered when selecting the maximum value. The maximum value for a row is selected from the coefficients of the matrix. In this case, we will divide the first row by 8.9, the second row by 22.3 and the third row by 0.2. This results in:

$$\left.\begin{array}{ccc|c} .00562 & -1 & .35955 & .79775 \\ .21973 & -.32287 & 1 & .18834 \\ 1 & -.15 & .3 & 11 \end{array}\right. \qquad (1.47)$$

## 1.13 Partial Pivoting

Now we can move the equations around (partial pivoting) so that the ones are on the diagonal. It is important to notice that normalization must be done first because without normalization it would be difficult to know how to rearrange the equation. There would be no basis for the row by row comparison.

$$\left.\begin{array}{ccc|c} 1 & -.15 & .3 & 11 \\ .00562 & -1 & .35955 & .79775 \\ .21973 & -.32287 & 1 & .18834 \end{array}\right. \qquad (1.48)$$

These are the equations we solve using Gaussian elimination.

## 1.14 Upper Triangular Form

The next step in the solution process is to put the matrix into upper triangular form. We will call this the forwards sweep. We will multiply the top row by 0.00562/1 and subtract it from the second row. This will place a zero in the first column of the second row. We will then repeat the process by multiplying the first row by 0.21973/1 and subtraction it from the third row. The resulting matrix is shown below.

$$\left.\begin{array}{ccc|c} 1 & -.15 & .3 & 11 \\ 0 & -.9916 & .3347 & .1798 \\ 0 & -.2899 & .9341 & -2.2284 \end{array}\right. \qquad (1.49)$$

We complete the process by multiplying the second row by -.2889/-.9916 and subtracting it from the third row. This leaves zeros in both the first and second columns of the third row. The resulting matrix is shown below.

$$\begin{array}{ccc|c} 1 & -.15 & .3 & 11 \\ 0 & -.9916 & .3427 & .1798 \\ 0 & 0 & .8339 & -2.2813 \end{array} \tag{1.50}$$

## 1.15 Back Solving

The back sweep is the last stage in the solving process. Here we solve for the unknown. The process starts with the last equation or row of the matrix and works up a row at a time till it reaches the top row.

$$X_3 = -2.2813 / .8339 = -2.7357 \tag{1.51}$$

$$X_2 = (.1798 - .3427 \times -2.7357) / -.9916 = -1.1268 \tag{1.52}$$

$$X_1 = (11 - (-.15 \times -1.1268) - (.3 \times -2.7357)) / 1 = 11.6517 \tag{1.53}$$

## 1.16 Overall Process

The overall process becomes

a. Normalize each equation
b. Move the equations to achieve diagonal dominance (partial pivoting).
c. Do a forward sweep that transforms the matrix to upper triangular form
d. Do a backwards sweep that solves for the values of the unknowns.

This algorithm can be written into very compact and efficient computer algorithm and is one of the more common ways of solving large numbers of simultaneous linear equations.

## 1.17 MATLAB / OCTAVE

Both MATLAB and OCTAVE can be used to solve this problem using Gaussian elimination. We will define the equations as matrix A and the values of the equations as vector B. This is shown below.

```
A=[0.5,-8.9,3.2;4.9,-7.2,22.3;0.2,-.03,.06]          (1.54)
    0.5  -8.9      3.2
    4.9  -7.2      22.3
    0.2  -0.3      0.6
```

```
B=[7.1; 4.2; 2.2]                                      (1.55)
    7.1
    4.2
    2.2
```

We can solve these simply by dividing A by B using the "\" symbol.

```
A\B                                                    (1.56)
    11.6517
    -1.1268
    -2.7357
```

Another way to solve this system of equations it to use the inverse of the matrix as prescribed in equation 1.21. Using that method, we would write:

```
inv(A)*B                                   (1.57)
    11.6517
    -1.1268
    -2.7357
```

This gives us the same answer using another method. In this latter case, MATLAB / OCTAVE does not use Gaussian elimination to solve the problem.

**PROBLEMS**

In each of the problems below use either MATLAB or OCTAVE to compute $(AB)^T$ and $B^TA^T$. Are they equivalent?

1. $$A = \begin{bmatrix} 2 & 3 & 1 \\ -1 & 0 & 2 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

2. By hand, compute the transpose of the matrix

$$\begin{bmatrix} 3 & 4 & 2 \\ 1 & 6 & 9 \\ 2 & 4 & 3 \end{bmatrix}$$

Solve the following equations using Gaussian elimination. Do not use row normalization or pivoting for these solutions.

3. $$\left.\begin{matrix} 1 & 3 & 2 \\ 2 & 2 & 4 \\ 1 & 2 & 6 \end{matrix}\right| \begin{matrix} 2 \\ 0 \\ 6 \end{matrix}$$

4. $$\left.\begin{matrix} 2 & 4 & 3 \\ 8 & 18 & 17 \\ 6 & 16 & 16 \end{matrix}\right| \begin{matrix} 1 \\ 12 \\ 13 \end{matrix}$$

5. Put the following system of equations into upper triangular form. Do not use partial pivoting or row normalization.

$$\left.\begin{matrix} 3 & 4 & 2 \\ 6 & 2 & 1 \\ 3 & -2 & 5 \end{matrix}\right| \begin{matrix} 8 \\ 3 \\ -7 \end{matrix}$$

Use the back solving method to solve the following systems of equations.

6.

$$\left[\begin{array}{ccc|c} 1 & 4 & 3 & 7 \\ 0 & 5 & 2 & 13 \\ 0 & 0 & 4 & 16 \end{array}\right]$$

7.

$$\left[\begin{array}{cccc|c} 1 & 6 & 3 & 7 & 50 \\ 0 & 2 & 4 & 3 & 28 \\ 0 & 0 & 6 & 3 & 30 \\ 0 & 0 & 0 & 9 & 36 \end{array}\right]$$

Rewrite the following system of equations using row normalization and partial pivoting.

8.

$$\left[\begin{array}{ccc|c} 1 & 3 & 2 & 2 \\ 1 & 0 & -2 & -6 \\ 1 & 2 & 6 & 6 \end{array}\right]$$

9.

$$\left[\begin{array}{ccc|c} 5 & -2 & 1 & 22 \\ 4 & 0 & 5 & 2 \\ 1 & 4 & -5 & 10 \end{array}\right]$$

10. Solve the following system of equations using Gaussian elimination. Demonstrate equation normalization and partial pivoting. Show all work.

$$\begin{aligned} 6X_1 - 3X_2 - 3X_3 + 2X_4 &= -1 \\ X_1 + 4X_2 + 6X_3 - 7X_4 &= -1 \\ 8X_1 + 2X_2 - 3X_3 - X_4 &= -1 \\ 2X_1 + 3X_2 + 5X_3 - 6X_4 &= -1 \end{aligned}$$

Solve the following system of equations using MATLAB or OCTAVE.

11.

$$\begin{aligned} 3X_1 + 2X_2 + X_3 - 4X_4 &= -6 \\ 2X_1 + X_2 - 4X_3 + 2X_4 &= 0 \\ 2X_1 + 3X_2 - 7X_3 + 2X_4 &= -5 \\ 2X_1 + 5X_2 + 2X_3 - 4X_4 &= 2 \end{aligned}$$

12.

$$\left[\begin{array}{cccc|c} -4 & 9 & 3 & 4 & 36 \\ 8 & -3 & 4 & -2 & 4 \\ 4 & -6 & -12 & 1 & 5 \\ 6 & -3 & 6 & 8 & 2 \end{array}\right]$$

13.

$$\left[\begin{array}{cccc|c} 8 & 6 & 2 & -5 & -2 \\ 2 & 9 & 4 & -6 & 31 \\ 4 & 3 & 12 & -3 & 9 \\ 4 & 1 & 3 & 10 & 16 \end{array}\right]$$