

MAPLE Worksheet Number 11  
Linear Algebra: Elementary Matrix Manipulation

Now we will explore topics from Linear Algebra (stuff covered in MATH 2360 at TTU). To begin we will need to load the linear algebra package. The command is

```
[ > with(linalg);
```

There's lots of stuff here, let's hope we won't have to use it all. The first thing we need to know is how to enter a given matrix. One way is to list each row as a vector. For example to enter the matrix

$$M = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

use the MAPLE command

```
[ > M:=matrix([[1,0,1],[2,1,0],[0,1,1]]);
```

Another way is to give the row-column size of the matrix and then list all the elements. For example to enter the matrix

$$N = \begin{bmatrix} 0 & 1 \\ 2 & -1 \\ 3 & -2 \end{bmatrix}$$

try the MAPLE command

```
[ > N:=matrix(3,2,[0,1,2,-1,3,-2]);
```

In general you can use either command you wish. You should practice using both to see the benefits of each. Now enter the following matrices

$$P := \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad R := \begin{bmatrix} \frac{1}{2} & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \text{ and } Q := \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

It seems to me the latter method is the easiest to use for entering a one column matrix (column vector).

For example the commands

```
[ > C:=matrix(4,1,[1,2,3,4]);C:=matrix([[1],[2],[3],[4]]);
```

Define the same 4x1 matrix.

To view a previously defined matrix, say M, try the command

```
[ > M;
```

This isn't terribly enlightening. To "see" M we use the command

```
[ > evalm(M);
```

I would guess that evalm is short for evaluate the matrix. Next we might want to add two matrices. Try adding M and P in the obvious way

```
[ > M+P;
```

Again we have to use evalm to see the result. Actually to compute and view it is nice to include both commands

```
[ > M+P=evalm(M+P);
```

Try adding N+P, P+Q, and N+R. Explain the results.

Next try squaring P in the obvious way and use evalm to see the results.

```
[ > P^2=evalm(P^2);
```

Just for the heck of it compute and view  $P^{20}$ . That's not a calculation I would want to do by hand.

Try squaring N and viewing the result. Explain the result.

Try multiplying M and P in the "obvious" way and viewing the result. What happens? For some strange reason we have to use `&*` for the matrix multiplication operator. Now compute and view MP.

```
[ > MP:=evalm(M&*P);
```

Does MP=PM ? Demonstrate.

Compute and view MN. Why can't we compute NM? Perform and view all possible products of any two of the five matrices M, N, P, R, and Q.

Explain why the following MAPLE calculation is very misleading, in fact wrong.

```
[ > expand((M+P)^2);
```

For general matrices M and P, what is the correct formula for  $(M + P)^2$  ?

Scalar multiplication works just the way you would expect. Compute and view  $2M$ ,  $-4N$ ,  $aR$

The command for finding a particular entry in the *i*th row and *j*th column for a matrix K is `K[i,j]`. Use it to find the entry in the 2nd row, 3rd column of M; now the 3rd row, 1st column entry of N. What would you expect if you tried the command `N[1,3]`? Why?

Next compute and view  $M^{-1}$ . Symbolically speaking this notation should be the same as  $\frac{1}{M}$ , however

since M is a matrix it isn't clear what  $\frac{1}{M}$  even means. Whatever it means though,  $M \frac{1}{M}$  should equal 1.

But again we are working with matrices and two of them multiplied together should yield another matrix, so the "1" should be a matrix. What matrix should it be and what do we call the matrix

represented by  $\frac{1}{M}$  ? To answer these questions perform the following command sequence:

```
[ > M^(-1);
```

```
[ > L:=evalm(%);
```

```
[ > evalm(M&*L);
```

```
[ > K:=inverse(M);
```

```
[ > M&*K:=evalm(M&*K);
```

Try computing the inverse of N, what happens?

See if MAPLE knows the general formula for the inverse of a 2x2 matrix. In other words define the general matrix GM as below and then compute it's inverse.

```
[ > GM:=matrix([[a,b],[c,d]]);
```

```
[ > inverse(GM);
```

Now in similar fashion see if MAPLE knows the formula for the inverse of a general 3x3 matrix.

Ask MAPLE to compute the inverse of the matrix

$$T = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

What is the resulting error message? What does this mean?

The MAPLE command for computing the determinate of a matrix is `det(name_of_matrix)`; Use it to compute the determinate of T. Next compute `det(M)`, `det(P)`, `det(Q)`, `det(N)`, and `det(R)`. What do you observe?

Compute `det(GM)` to see the general formula for the determinate of a 2x2 matrix. Where does this determinate occur in the inverse of GM? Compute the general formula for a 3x3 matrix. Where does its determinate occur in the inverse?

Next compute `det(M(-1))` and `det(P(-1))`. What is the apparent relation between the determinate of a matrix and that of its' inverse?

Compute the determinate of  $MP$ . How does it compare to the product of `det(M)` and `det(P)`?

How does the determinate of  $M^2$  compare with that of M? How does the determinate of  $M^{20}$  compare with that of M?

Perform the following command sequence:

```
[ > inverse(M)*inverse(P);  
[ > evalm(%);  
[ > inverse(P*M);
```

This suggests the following identity for general invertable matrices M and P

$$(PM)^{(-1)} = M^{(-1)}P^{(-1)}.$$

Is this identity always true for any two invertable matrices M and P? Explain.

Another basic matrix operation is "transpose". Experiment with the command `transpose` to see what it does to a matrix. Compute and view `transpose(R)` times `transpose(N)` and compare the resulting matrix with `transpose(NR)`. What general identity does this suggest?

We'll finish this introduction to basic matrix manipulation by looking at some ways we can enter special types of matrices with MAPLE.

A matrix with constant rows:

```
[ > matrix([[a$4],[b$4],[c$4],[d$4]]);
```

A diagonal matrix:

```
[ > diag(e,e,e,e);
```

The identity matrix:

```
[ > array(identity,1..4,1..4);
```

Constant matrix:

```
[ > matrix(3,4,8);
```

Variable matrix:

```
[ > matrix(2,5,a);
```

Mixed variable and constants:

```
[ > a(1,2):=5;a(2,2):=-2;a(2,5):=Pi;
```

```
[ > matrix(2,5,a);
```

In terms of row/column position:

```
[ > h:=(i,j)->1/(i+j-x);
```

```
[ > matrix(4,3,h);
```

```
[ > matrix(3,6,(i,j)->max(i,j));
```

```
[ > matrix(3,6,(i,j)->min(i,j));
```

```
[ > matrix(3,6,(i,j)->x^max(i,j));
```

Vandermonde matrices:

```
[ > vandermonde([x,y,z]);
```

```
[ >
```

```
[ > vandermonde([1,2,3]);
```

Hilbert matrices:

```
[ > hilbert(6);
```

Random matrices:

```
[ > randmatrix(3,4);
```

```
[ > randmatrix(3,4);
```

Row vectors:

```
[ > vector([1,v,4]);
```

```
[ > randvector(9);
```

```
[ > vector(4,i->x^i);
```

Combinations of matrices:

```
[ > augment(M,N);
```

```
[ > ### WARNING: calling sequence has changed; see help page for  
  details
```

```
[ > blockmatrix(2,2,M,N,P,N);
```

Taking from an existing matrix:

```
[ > evalm(M);
```

```
[ > submatrix(M,2..3,1..2);
```

```
[ >
```