MATHEMATICS 201-105-RE
Linear Algebra
Martin Huard
Winter 2006

Matrices with Maple

When working with matrices on Maple, the first thing to do is to load the *linalg* package, which contains a lot of commands that we need.

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

Definition of Matrices with Maple

Matrices can defined with the command `matrix( )` by entering each row of the matrix. The command has the form

matrix([[row 1], [row 2],…,[row n]]);

For example, let us define A as the matrix $\begin{bmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \end{bmatrix}$.

```
> A:=matrix( [ [1,2,3], [5,6,7] ] );
```

For small matrices, you can use the matrix palette on your left. The numbers are filled in by typing them one at a time, and pressing the TAB key between two entries. This would give

```
> A:=<<1 | 2 | 3>, <5 | 6 | 7>>;
```

We can call elements in the matrix simply by specifying its location with the command

A[row #, column #]

For example, suppose we want the 5, which is in the second row and first column..

```
> A[2,1];
```

We can also call columns or rows in a matrix with the `col( )` or `row( )` commands from the *linalg* package. For example, let us call the second row and the third column.

```
> row(A,2);
> col(A,3);
```

Algebra of Matrices

Addition

The addition of matrices is done using the `evalm( )` command. For example,

let $A = \begin{bmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \end{bmatrix}$ and $B = \begin{bmatrix} 3 & 5 & -1 \\ 3 & 2 & 0 \end{bmatrix}$.

```
> A:=matrix( [ [1,2,3], [5,6,7] ] );
  B:=matrix([ [3,5,-1], [3,2,0] ]);
```

Then $A + B$ is the matrix given by :

```
> evalm(A+B);
```

There is an other way to add matrices, using the `matadd( , )` command in the *linalg* package.

> **matadd(A,B);**

Scalar multiplication

Multiplication by a scalar is handled in the same way as addition. For example, let us find 4A. Using the `evalm( )` command

> **evalm(4*A);**

or using `scalarmul( , )` command from the *linalg* package.

> **scalarmul(A,4);**

Multiplication

Let us multiply the matrices $A = \begin{bmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \end{bmatrix}$ and $C = \begin{bmatrix} 5 & -2 \\ 1 & 3 \\ -4 & 7 \end{bmatrix}$.

> **A:=matrix([[1,2,3], [5,6,7]]);**
> **C:=matrix([[5,-2], [1,3], [-4,7]]);**

Using the `evalm( )` command with ' &* ' as the symbol for matrix multiplication, we have

> **evalm(A&*C);**

or, with the `multiply( )`command from the *linalg* package.

> **multiply(A,C);**

If you want the power of a matrix, for example A^4, then we have

> **evalm(A^4);**

Other operations

There are other operations in the *linalg* package that we can do with matrices, such as taking the *trace*, the *transpose*, the *adjoint*, the *determinant* and finding the *inverse*. Let us look at these operations using, as an example, the matrix *A* defined below.

> **A:=matrix([[0,-3,5],[-4,4,-10],[0,0,4]]);**

> **trace(A);**
> **transpose(A);**
> **adjoint(A);**
> **det(A);**
> **inverse(A);**