**Matlab Commands (By Chris Kheng)**

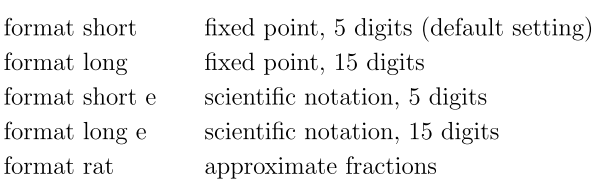
Type **>>cd C:\MA1101R** before you start

1. help *topic* : Give infor about the *topic*

2. A^n : raise matrix A to the power n

3. A’ : Transpose of A

4. inv(A): inverse of A

5.

# Note if you use **format rat**, then an \* may appear when you expect the quantity to be 0.

# Scientific notation:

e.g. **3.6000e+012 means 3**.6 x 1012

6. **clc** : clear the command window

**Clear**: clear all previously assigned variables

# To clear individual variable, type:

clear *variablename*

# Double clicking the variable name in the

Workplace allows you to edite the variable

7. To **access** an entry in a matrix, e.g. matrix A:

A(i, j)

8. To **assign** a value / variable to an i,j entry in

Matrix A, type:

A(i, j) = value / variable

**9. Commands for Graphing**

~ **plotline**

~ **plotplane**

~ after executed, click on **Reset** first

~ azimuth represents the x-y view, i.e.

turning to left or turning to right

**IMPORTANT FUNCTIONALITIES**

1. **[A B]**

~ combine matrix A and matrix B to form an

augmented matrix

**~** Note that you can combine more than 2

matrices, e.g. [A B C]

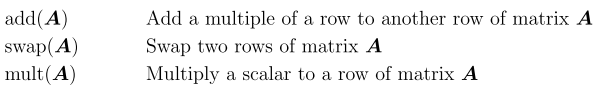
2. **rref(A)**

3. **det(A)**

4. **A \ B** solve the linear system Ax = B

~ Only works if the solution of the system exists and it's UNIQUE

5. **Gaussian Elimination**



~ This will **not** change the original matrix A

6. **dot(v1, v2) :** Dot product of the vector v1 and v2

~ Can also use v1’ \* v2, i.e. v1Tv2

7. **maxcol(A)** : Give a maximum set of LI column

vectors in Matrix A (To find that of row

vectors, simply transpose A first before

applying maxcol).

8. **solbasis(A) :** Give a basis for the solution space of

the system AX = 0

9. **rank(A)** : Give rank of matrix A

9. **A(:, j)**

~ to take out the jth column from a matrix A

10. **eigval(A)** : Give eigenvalues of matrix A

**eigvec(A)**: Give eigenvectors of matrix A

11. **[V X] = eigvec(A)**

~ store eigenvectors and eigenvalues

of A in V and X respectively

**[V] = eigvec(A)**

~ store **eigenvectors** of A in V

**[X] = eigvec(A)**

~ store **eigenvalues** of A in X

12. **poly(A)** : return the coefficients of the

characteristic polynomial of matrix A,

including the constant

~ Note if square matrix A is of order n, then the

characteristic polynomial is of degree n

13. **Swapping Columns**

**A(:, [i j]) = A(:, [j i])**

~ e.g swapping column 6 with column 1 in matrix B

~ B(:, [6 1]) = B(:, [1 6])

~ technically this is an assignment expression where you assign the columns of the matrix on the RHS of = to the columns of the matrix of the LHS; therefore you can assign the columns in other matrix to the columns of the matrix on the LHS

14. **Quick inserting columns of a matrix into a new matrix**

~ e.g. inserting the column 4 ~ 6 of a matrix B into a new matrix

~ First declare a new empty matrix A

A = [ ];

Then: A(:, [1, 2, 3]) = B(:, [4, 5, 6])

~ The new matrix A will have its first 3 columns as the column 4~6 of matrix B

15. **Application type (Finding Limit)**





~ x0 is the initial value