The notes are not meant as an Introduction to MATLAB for total newcomers. If a newcomer wants to learn a bit about MATLAB, see me and I can help you get started.

These notes are for students who already know a little about MATLAB. You can look at the commands and probably decipher the general pattern of similar commands.

Define a matrix in MATLAB: Example A = [1 2 3;4 5 6;7 8 9] or, alternatively,
A = [1,2,3;4,5,6;7,8,9]

defines the matrix $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ *Use semicolons ";" to separate the rows of the matrix; the entries within each row are separated by spaces, or commas, whichever you prefer.*

Create the matrix that is the i^{th} row of a matrix A: R2=A([2], :)
Here, [2] picks out the first element in row 2, and " : " means "all columns" so

R2 = [4 5 6]

Similarly, the command B=A([2,3],:) gives B=$\begin{bmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ that contains the second and third rows of A.

The command ' transposes a matrix: the rows and columns are interchanged:

the command A' produces the matrix $\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$

Commands to perform EROs on A:

Interchange Row 1 and Row 3 of A: A([1 3], :) = A([3 1], :)

Rescale Row 3 by a factor of 2: A([3], :) = 2*A ([3], :)

Add 5*Row2 to Row1: A([1], :) = A([1], :) + 5*A([2], :)

The preceding commands may be useful to practice row reductions at the beginning; or, occasionally, for other reasons, But usually, one you understand the method, the row reduced echelon form is what you want to see. There is a built-in command to get to the row reduced echelon form of A in a single step: rref

$$\text{rref}(A) = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$

Arithmetic

MATLAB computes using floating point arithmetic, not fractions. Answers may be displayed, rounded to just 4 decimal places: 1/3=0.3333, although more digits are actually in memory.

The command “format long”increases the display to 15 places:
1/3 = 0.333333333333333.

However, that's still only an approximation to the exact value 1/3.

In some very "delicate" situations, even a very small round off error might lead you to an incorrect theoretical conclusion about a matrix.

If for example, the exact rref for some 2×2 matrix A is $\begin{bmatrix} 1 & 0 \\ 0 & 10^{-100} \end{bmatrix}$

MATLAB will tell you (incorrectly) that rref(A) = $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$. It turns out that the matrices

$\begin{bmatrix} 1 & 0 \\ 0 & 10^{-100} \end{bmatrix}$ and $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ have some very different properties – differences that can

matter.

Such “delicacies” related to roundoff errors don't usually matter in introductory courses with simple and well-behaved examples. But people who use linear algebra seriously on large scale problems use computers heavily and do need, sometimes, to worry about the effect of roundoff error. Sometimes, for example, there may be two or more ways to compute an answer, and some of the ways may be likely to generate more roundoff error than others.