MATH1019 Linear Algebra and Statistics for Engineers
Laboratory Session 6

Learning outcomes for this session

At the end of this session, you will be able to

1. Use matlab to define vectors, perform simple arithmetic operations on them, and evaluate dot products.
2. Use matlab for cross products.
3. Use matlab to define matrices and perform simple arithmetic operations on them.

Overview

1. Vectors and their arithmetic in matlab
2. Some of the matlab commands introduced for vectors: `norm`, `dot`, `cross`
3. Matrices, vectors and their arithmetic in matlab
4. Some of the matlab commands introduced for matrices: `size`, `ones`, `zeros`, `eye`, `diag`, `toeplitz`

matlab – with vectors

When doing these labs at Curtin we recommend using matlab directly on the computer, i.e. *not* through the web. Much of this lab is revisiting, but away from the web, material you saw in the Matlab OnRamp.
A vector with the three components 1,2 and 2 can be entered as
```
v=[1,2,2]
```
The magnitude of a vector **u** can be computed by using the norm command `norm(u)`.
(Caution: matlab's `length` gives the number of elements in the vector.)
So for example the magnitude of the vector **v** is found by typing in
```
norm(v)
```
Just as with scalars we can add and subtract vectors **u** and **v**, this is done by using the usual u+v and u-v commands respectively. For the scalar multiplication of a vector **v** by a scalar k, we simply type k*v.
(It is important to know the difference between row vectors and column vectors, but we rely on the OnRamp to teach you this. It will be very clear when we get to matrices and transposes.)

Exercises

1. Define $\mathbf{u} = [1, 4, -3]$ and $\mathbf{v} = [2, -4, 4]$ then determine the following
(a) $\mathbf{u} + \mathbf{v}$
(b) $2\mathbf{u} - \mathbf{v}$
(c) The magnitude of $\mathbf{u}$
(d) A unit vector in the direction of $\mathbf{v}$
(e) The magnitude of $2\mathbf{u} - \mathbf{v}$
(f) A vector of length 4 units in the opposite direction of $\mathbf{v}$

An aside, for now. If you need algebra using symbols, matlab's Symbolic Toolbox will do it. As an example, a vector with components a, b, and c, can be entered as

```
syms('a','b','c'); vSym=[a,b,c]
```

2. Given vectors $\mathbf{u}$ and $\mathbf{v}$, the dot product can be obtained using the command `dot(u,v)`. With the numeric vectors $\mathbf{u}$ and $\mathbf{v}$ given at the beginning of item 1 above, and also $\mathbf{w} = [0, 3, 4]$
(a) Show the vectors $\mathbf{u}$ and $\mathbf{w}$ are perpendicular
(b) Calculate the angle between $\mathbf{u}$ and $\mathbf{v}$ giving your answer to 5 decimal places (you will need to use the command `acos` for this)
(c) Determine the scalar and vector projection of $\mathbf{u}$ on $\mathbf{v}$

cross product and scalar triple product

1. Get help on the matlab command `cross`. Given the vectors $\mathbf{u}$ and $\mathbf{v}$, the vector product or cross product is computed using the `cross` command. Define
u = [1, 3, -5] ; v = [2, 7, 9]
then determine the area of the parallelogram formed by $\mathbf{u}$ and $\mathbf{v}$. Also given
w = [4, 2, 1]
evaluate the scalar triple product, $\mathbf{w}.(\mathbf{u} \text{ x } \mathbf{v})$. (Recall the `dot` command from the information on vectors earlier in this lab.) Based on the result of the scalar triple product, decide whether the three vectors $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$ are coplanar or not. Having entered these vectors, you can construct a matrix for which they are rows with the command

```
A = [u; v; w]
```

(Later in MATH1019 we will see determinants and the scalar triple product as `det(A)`.)

matlab – with matrices

1. You can enter matrices componentwise. Type in the following:

```
C= [1,2,3,4; 4,3,2,1; 1,2,1,2]
size(C)
```

Use semicolon separators to build a vector as a column matrix. The column vector with components 1,2,3,4 can be entered as
`v=[1;2;3;4]`
You can ask for its `size` or its `length` with these matlab commands. And you can multiply to form the matrix product $C\,\mathbf{v}$ as follows
`Cv = C*v`
(There is also a command `mtimes`.)

2. If a matrix has a nice structure, then its creation is usually simpler. Type the following commands onto your worksheet and note what matrix is created in each case.

```
M1= ones(3)
M2= zeros(3)
I3= eye(3)
M3= diag([1,2,3])
n=4;
M4=toeplitz([2 -1 zeros(1, n-2)], [2 -1 zeros(1, n-2)])
```

There are lots of different sorts of matrices, but, for now this is enough. If you are interested in the definition for the last, use either the matlab help, or
https://en.wikipedia.org/wiki/Toeplitz_matrix

3. You can extract elements from matrices, e.g. from M4 above:

```
M4(1,2)  % gives the element in the 1st row and 2nd column
M4(:,2) % gives the whole 2nd column
```

4. As mentioned in item 1 above, multiplying 2 matrices is done by using the * symbol. Using the matrix C from item 1 and the matrix M3 defined in item 2, the product is obtained by the command
`C*M4` Since C is a 3×4 matrix and M4 is 4×4, the product `C*M4` is defined but `M4*C` is not. Now try to type in
`M4*C` You will get an error message.

Exercise

From what you have learnt in the OnRamp and in item 3 above, get matlab to form the 3×3 submatrix of the first 3 rows and first 3 columns of `M4`.