Introduction to Scientific Computing Lab 7
MATLAB II
2017

1 Objectives

After completing these exercises you will be able to:

- Write functions in MATLAB
- Use control flow statements in MATLAB

2 Notes

Work individually

Create a directory for each week so you can come back to your codes in the future. Create files for each of the different exercises and name them in a logical manner, so `exercise1.m` for example.

When changing the source code, MATLAB will automatically save the code for you when you run.

When you have completed all exercises, ask a demonstrator to assess your work. They will test your code and ensure it is formatted well with good commenting, structure and variable names. This is a useful feedback mechanism, so listen to what the demonstrator has to say and their recommendations for improving your code.

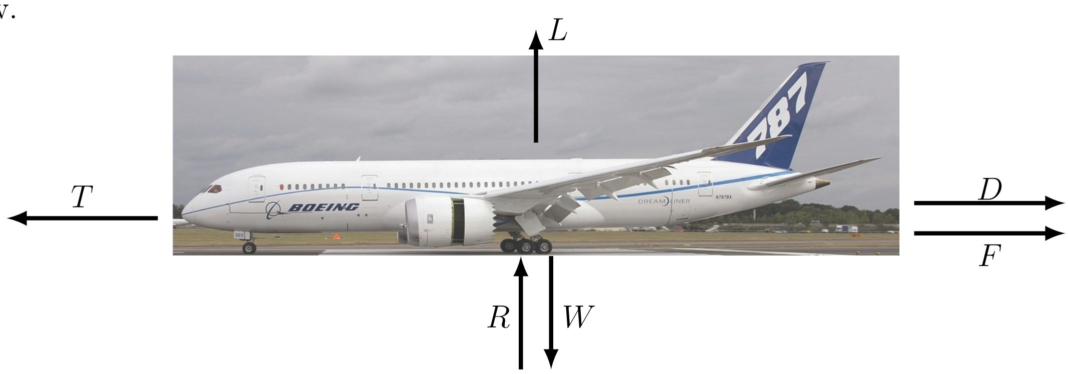
In case of an error, read the compiler error. This will often tell you the line (or close to the line) where the error is occurring. Fix it, test and repeat for the errors you have. If you are getting nowhere then it can often be useful to copy the error into google, or use some keyword searches. If you are really stuck on one error then call over an assistant who will be able to point you in the right direction.

3 Simulating Take-Off of an Aircraft

In this exercise you will use simple Newtonian mechanics to simulate the take-off roll and estimate the take-off time of an aircraft. You will work through four stages to obtain the final answer:

- Determine the reaction force
- Plot the reaction force
- Determine the acceleration
- Simulate the take-off roll

A simplified illustration of the forces acting on an aircraft during take-off roll are shown below.



All of the fixed quantities you will need are shown in table 1. Hard-code these as variables where appropriate in your scripts and functions.

Table 1: Variables to include

Variable	Nomenclature	Value
Mass	m	15000 kg
Wing area	S	50 m^2
Air density	ρ	1.225 kg/m^3
Gravitational acceleration	g	9.81 m/s^2
Thrust	T	110,000 N
Lift coefficient	C_L	0.6
Friction coefficient	μ	0.02

3.1 Exercise 1: Function to Determine the Reaction Force

Write a **function** that has the aircraft velocity, v, as an input, and outputs the reaction force. The reaction force is given by:

$$R = W - L \tag{1}$$

where the weight W and the lift L are given by:

$$W = mg \quad , \quad L = \frac{1}{2}\rho v^2 S C_L \tag{2}$$

To test your function, call it from the command window:

```
>> R = freaction(testv)
```

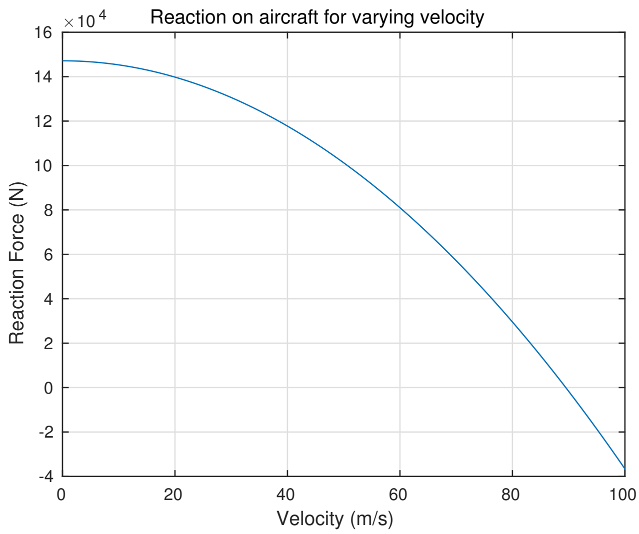
If $v = 0$, then $R = 147150$, and if $v = 100$, then $R = -36600$.

3.2 Exercise 2: Script to Plot the Reaction Force

Write a **script** that plots the reaction force for velocity values of 0 to 100m/s in 1m/s increments. Ensure your plot has:

- A title
- Axis labels
- A background grid
- Sensible limits

Your script should produce a plot that looks something like:



3.3 Exercise 3: Function to Determine the Horizontal Acceleration

Write another **function** that has the aircraft velocity, v, as an input, and outputs the horizontal acceleration of the aircraft, a. The horizontal acceleration comes from Newton's Second Law of Motion:

$$a = \frac{1}{m} F_{horizontal} \tag{3}$$

where the horizontal force pushing the aircraft forward is thrust, and the resistance force is from the aerodynamic drag, D, and the friction between the runway and the wheels, F. Therefore:

$$a = \frac{1}{m}(T - (D + F)) \tag{4}$$

The resistance forces are given by:

$$F = \mu R \quad , \quad D = \frac{1}{2}\rho v^2 S C_D \quad , \quad C_D = 0.04 + 0.06 C_L^2 \tag{5}$$

If $v = 0$, then $a = 7.14$, and if $v = 100$, then $a = 6.12$.

3.4 Exercise 4: Simulate the Take-off Roll

Write a new **script** to determine the time needed for the aircraft to take-off. We will simulate the take-off numerically using a simple forward-difference approach. This is another numerical integration method similar to the beam example from Lab 3, except the integration is slightly less accurate (but good enough for this job).

You will need to integrate until the reaction force becomes zero (i.e. the aircraft has taken off). The algorithm for performing this simulation is as follows:

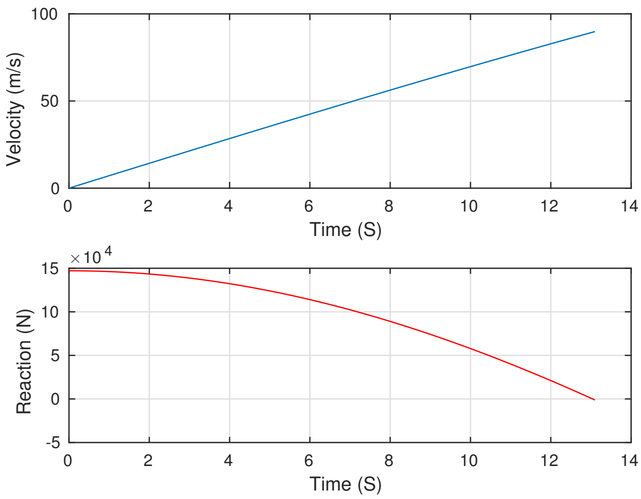
1. Set suitabel initial values for x, v, t and calculate an initial R
2. Until suitable exit value
   (a) Calculate the horizontal acceleration
   (b) Update t, x and v:

$$\begin{aligned} x(t + \delta t) &= x(t) + v(t)\delta t \\ v(t + \delta t) &= v(t) + a(t)\delta t \\ t &= t + \delta t \end{aligned}$$

(c) Update reaction force $R(t + \delta t)$

3. Plot v and R versus t

A suitable value for δt is 0.1s. Remember to add axis titles and a grid to you plot. The final plots should look like:



HINT: if you would like to export these two graphs in the same figure window, have a look at the `subplot` command.