Flight

August 24, 2020

fnc.py - Flight block

The goal of this file is explaining how each and every function contained within the Flight block of the fnc.py file works. The source for each physical equation implemented is provided, in addition to further explanations on what assumptions these models work under.

[2]:

```
#%% Script information
# Name: fnc.py
# Authors: Trajectory Team (Matias Pellegrini, Pablo Lobo)
# Owner: LIA Aerospace
#
#%% Script description
#
# The aim of this module is defining functions to be used in the simulation.
#
#%% Packages
import numpy as np
import c as c

#%% Flight

# This block implements the different functions required for the
# simulation.py file to be able to simulate the flight of the launch
# vehicle. Functions' assumptions and sources are documented in detail
# the Flight (Jupyter Notebook) document.
```

Function #1: mach

The aim of this function is to calculate the local Mach number at the instant of interest.

This equation is straight-forward and can be found in multiple sources exactly in the same way. What is of interest however is the impact of the different effects that appear depending on its magnitude. These are:

- Ma $<$ 0.3: *Incompressible flow*, where density effects are negligible.
- 0.3 $<$ Ma $<$ 0.8: *Subsonic flow*, where density effects are important but no shock waves appear.
- 0.8 $<$ Ma $<$ 1.2: *Transonic flow*, where shock waves first appear, dividing subsonic and supersonic regions of the flow.

- 1.2 < Ma < 3.0: *Supersonic flow*, where shock waves are present but there are no subsonic regions.
- 3.0 < Ma: *Hypersonic flow*, where shock waves and other flow changes are especially strong.

Taken from *Fluid Mechanics*, White F., McGraw Hill.

[3]:
```
def mach(V,Vs):
    # === INPUTS ===
    # V [m/s]        Local flow velocity
    # Vs [m/s]       Speed of sound in the medium at the local temperature
    # === OUTPUTS ===
    # M [adim]       Local Mach Number
    M = V/Vs         # [adim] - Local Mach Number
    return M
```

Function #2: re

The aim of this function is to calculate the local Reynolds number at the instant of interest.

This equation is straight-forward and can be found in multiple sources exactly in the same form. What is of interest is how to choose the characteristic length, named L, for the vehicle at hand. Either the diameter or the length of the vehicle might be used. This is yet to be defined.

[4]:
```
def re(V,kvisc,L):
    # === INPUTS ===
    # V [m/s]                   Local flow velocity
    # kvisc [m^2/s]             Kinematic Viscosity
    # L [m]                     Characteristic length
    # === OUTPUTS ===
    # re [adim]                 Local Reynolds Number
    re = V * L / kvisc   # [adim] - Local Reynolds Number
    return re
```

Function #3: thrust

The aim of this function is to calculate the generated thrust at the instant of interest.

This equation was taken from *Introduction to Rocket Science and Engineering*, Taylor T., CRC Press. Eq. (3.6)

$$F_{thrust} = \dot{m}v_e + (P_e - P_o)\,A_e$$

[6]:
```
def thrust(m_dot,Ve,Pe,Po):
    # === INPUTS ===
    # m_dot [kg/s]              Mass flow of propellant being expelled
    # Ve [m/s]                  Exhaust velocity of the gases
    # Pe [N/m^2]                Exhaust pressure
    # Po [N/m^2]                Pressure outside the nozzle
    # === OUTPUTS ===
    # thrust [N]                Thrust
    # === CONSTANTS ===
    # c.Ae [m~2]                Nozzle exit surface
    thrust = m_dot*Ve + (Pe - Po)*c.Ae
```

[ ]:

```

```