In [ ]:

```
import pyAVL
import os


def STARTG01_AVL(phase=1,alpha=0,T=1):
    '''
    Legend for flight_stg:

    1 = clean
    2 = takeoff flaps, gear up
    3 = takeoff flaps, gear down
    4 = landing flaps, gear up
    5 = landing flaps, gear down

    '''

    def flaps(deflection):
        AVLsp.addInput('D2 D2 {}'.format(deflection))
        AVLsp.addInput('D4 D4 {}'.format(deflection))
    def trim():
        # print('trimming...')
        AVLsp.addInput('D5 PM 0')
    def untrim():
        AVLsp.addInput('D5 D5 0')
    def runCL(CL):
        AVLsp.addInput('A C {}'.format(CL))
    def runalpha(alpha):
        AVLsp.addInput('A A {}'.format(alpha))

    # setup
    AVLsp = pyAVL.AVL()

    if not os.path.isfile('polar\\P{}_A{}.out'.format(phase,alpha)):
        print('P{}_A{}.out does not exist, running . . .'.format(phase,alpha))
        planeName = 'STARTG01'
        AVLsp.loadPlane(planeName)
        AVLsp.loadMass(planeName)
        AVLsp.setAtmosphere(25e3,0)
        AVLsp.addInput('oper')

        runalpha(alpha) # select alpha

        if T == 1: #trim if necessary
            trim()

        if phase == 1: # set flaps
            flaps(0)
        elif phase == 2:
            flaps(15)
        elif phase == 3:
            flaps(15)
        elif phase == 4:
            flaps(35)
        elif phase == 5:
            flaps(35)

        AVLsp.addInput('X') #run
        AVLsp.saveOutput('FT','polar\\P{}_A{}'.format(phase,alpha) ) # save case data
        AVLsp.runAVL()

    out = AVLsp.readFT( 'polar\\P{}_A{}.out'.format(phase,alpha) )  #read file output
    return out['CLtot'],out['CDind'] # ouput CLtot and CDind
    # print(out['CLtot'],out['CDind'])




if __name__ == "__main__":
    # STARTG01_AVL(1,0)
    planeName = 'STARTG01'
    AVLsp = pyAVL.AVL()
    AVLsp.loadPlane(planeName)
    AVLsp.loadMass(planeName)

    AVLsp.setAtmosphere(25e3,0)
    AVLsp.setVelocity(275*1.6878099)
    AVLsp.addInput('oper')
    AVLsp.addInput('C1\n')
    AVLsp.addInput('D5\nPM\n0')
    AVLsp.addInput('X')
    AVLsp.addInput('ST\n')
    AVLsp.saveOutput('ST','cruise_stab')

    # Trimmed for cruise at 25kft, 275 ktas
    # delevator = -2.025263382235408E+00
    # dCM/da: -2.976505617474167E+00
    # dCN/db: 9.882347864467708E-02
    # dCl/db: -6.180025626223679E-02


    AVLsp.addInput('oper')
    AVLsp.addInput('C1\n')
    # AVLsp.addInput('D1 D1 0')
    AVLsp.addInput('A C 2.2')
    AVLsp.addInput('D2 D2 15')
    # AVLsp.addInput('D3 D3 0')
    AVLsp.addInput('D4 D4 15')
    # AVLsp.addInput('D5 PM 0')
    AVLsp.addInput('X')
    AVLsp.addInput('ST\n')
    AVLsp.saveOutput('ST','takeoff_stab')

    # Trimmed for CL=2.2, flaps 15 deg
    # delevator = -2.079144021503641E+01
    # dCM/da: -3.033447955664601E+00
    # dCN/db: 1.094011615515749E-01
    # dCl/db: -1.057696877033421E-01

    AVLsp.addInput('oper')
    AVLsp.addInput('C1\n')
    # AVLsp.addInput('D1 D1 0')
    AVLsp.addInput('A C 3.3')
    AVLsp.addInput('D2 D2 35')
    # AVLsp.addInput('D3 D3 0')
    AVLsp.addInput('D4 D4 35')
    # AVLsp.addInput('D5 PM 0')
    AVLsp.addInput('X')
    AVLsp.addInput('ST\n')
    AVLsp.saveOutput('ST','landing_stab')

    # Trimmed for CL=3.3, flaps 35 deg
    # delevator = -2.875557496812263E+01
    # dCM/da: -3.006006645280893E+00
    # dCN/db: 1.239651415854076E-01
    # dCl/db: -1.382750681113420E-01


    # AVLsp.addInput('X')

    AVLsp.runAVL()
    # out = AVLsp.readFT('cruise.out')
    # print(out['CLtot'],out['CDind'])
    # out = AVLsp.readFT('takeoff.out')
    # print(out['CLtot'],out['CDind'])
    # out = AVLsp.readFT('landing.out')
    # print(out['CLtot'],out['CDind'])
    # # post



    # interfacing notes
    # flap deflection
    # angle of attack
    #
```