

```

1 import math
2 import numpy as np
3 # import sys
4 # import os
5
6 class CLalfa:
7
8     def __init__ (self, clalfa,rangeAR,Mach,sweepHalfChord):
9         self.clalfa = clalfa
10        self.rangeAR = rangeAR
11        self.Mach = Mach
12        self.sweepHalfChord = sweepHalfChord
13
14    def ellipticalCLalfa(self):
15        ellipticalCLalfa = self.clalfa/(1+(self.clalfa/(math.pi*self.rangeAR)))
16        return ellipticalCLalfa
17
18    def hemboldCLalfa(self):
19        hemboldCLalfa = (2*math.pi*self.rangeAR)/(2+math.sqrt(self.rangeAR**2 + 4 ))
20        return hemboldCLalfa
21
22    def polhamusCLalfa (self):
23        beta = (1-self.Mach**2)**(0.5)
24        k = self.clalfa/(2*math.pi)
25        #remember to remove 57.3 later
26        polhamusCLalfa = (2*math.pi*self.rangeAR)/( 2 + math.sqrt(((self.rangeAR*beta )/k)**2 *(1+((
math.tan(self.sweepHalfChord/57.3))**2/beta **2))+4))
27        return polhamusCLalfa
28
29
30 class classCLmax:
31
32    def __init__ (self, clmaxRoot,yMGC,wingSpan,clmaxTip,sweepQuarterChord):
33        self.clmaxRoot = clmaxRoot
34        self.yMGC = yMGC
35        self.wingSpan = wingSpan
36        self.clmaxTip = clmaxTip
37        self.sweepQuarterChord = sweepQuarterChord
38
39    def rapidCLmax(self):
40        clmax = self.clmaxRoot + (self.yMGC/self.wingSpan)*(self.clmaxTip-self.clmaxRoot)
41        # IVE SKIPPED THE root and tip Re calculations to help reading the equivalent clmax on the
NACA graphs
42        CLmaxo = 0.9*clmax
43        kA = math.cos(self.sweepQuarterChord/57.3)
44        rapidCLmax = CLmaxo*kA
45        return rapidCLmax
46
47

```