

---

```

%Greg Soos
%AERO 215-03
%Final
%11.30.2017

clear all; close all; clc

%QUESTION 1

S = 24 ; %wing ref. area [m^2]
q = .27*1000 ; %dynamic pressure [Pa]
Cdo = .075 ;
lift_min = -177.5 ; %N
lift_max = 5452 ; %N

%PART A
Cl_min = lift_min/(q*S) ;
Cl_max = lift_max/(q*S) ;
disp(['Minimum Cl: ', num2str(Cl_min)])
disp(['Maximum Cl: ', num2str(Cl_max)])
%Using Cl=L/(q*S) equation for min and maxes.

Minimum Cl: -0.027392
Maximum Cl: 0.84136

%PART B
Clvec = Cl_min:.05:Cl_max ;
K1 = .4 ;
K2 = .6 ;
K3 = .8 ;
K4 = 1 ;
K5 = 1.2 ;
Cd1 = Cdo + K1*Clvec.^2 ;
Cd2 = Cdo + K2*Clvec.^2 ;
Cd3 = Cdo + K3*Clvec.^2 ;
Cd4 = Cdo + K4*Clvec.^2 ;
Cd5 = Cdo + K5*Clvec.^2 ;
%I know I'll probably get marked off for hard coding the above part in
    but
%for the life of me I was not sure how to multiply each individual
%component of the K-vector by each component of the Cl-vector, and
    then
%create a Cd-vector the size of the K-vector. So for the sake of doing
    the
%problem I hard coded it like this.

figure(1)
hold on
plot(Cd1, Clvec)
plot(Cd2, Clvec)
plot(Cd3, Clvec)
plot(Cd4, Clvec)
plot(Cd5, Clvec)

```

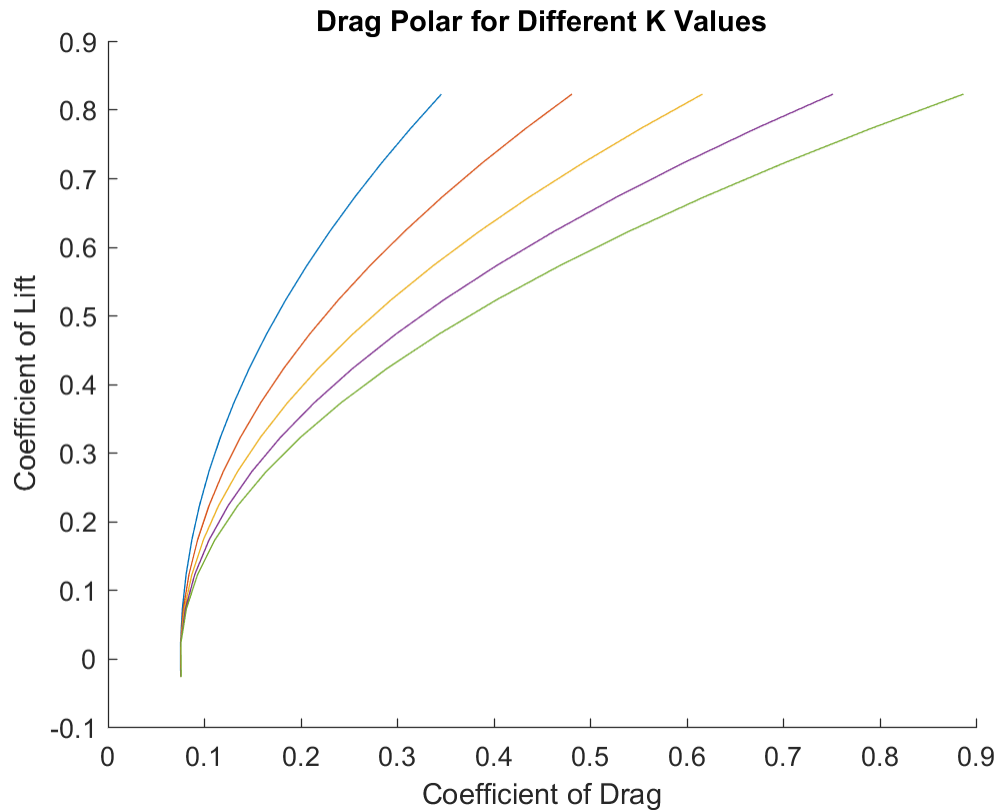
---

---

```

title('Drag Polar for Different K Values')
xlabel('Coefficient of Drag')
ylabel('Coefficient of Lift')

```



```

%PART C
C11 = sqrt(Cdo/K1) ;
C12 = sqrt(Cdo/K2) ;
C13 = sqrt(Cdo/K3) ;
C14 = sqrt(Cdo/K4) ;
C15 = sqrt(Cdo/K5) ;
Cls = [C11, C12, C13, C14, C15] ;
%Calculating Cl values at location of L/Dmax for each K value, then
put
%into one vector.
Cd1n = Cdo + K1*C11^2 ;
Cd2n = Cdo + K2*C12^2 ;
Cd3n = Cdo + K3*C13^2 ;
Cd4n = Cdo + K4*C14^2 ;
Cd5n = Cdo + K5*C15^2 ;
Cds = [Cd1n, Cd2n, Cd3n, Cd4n, Cd5n] ;
%Calculating Cd values at location of L/Dmax for each K value using
drag
%polar equation and Cl values calculated above.
figure(1)
hold on
plot(Cds, Cls, 'k*', 'LineWidth', 3)

```

---

```

legend('K = .4', 'K = .6', 'K = .8', 'K = 1.0', 'K = 1.2', 'Max L/D
Locations')
LDmax = CIs./Cds ;
disp(['Max L/D for K1: ', num2str(LDmax(1))])
disp(['Max L/D for K2: ', num2str(LDmax(2))])
disp(['Max L/D for K3: ', num2str(LDmax(3))])
disp(['Max L/D for K4: ', num2str(LDmax(4))])
disp(['Max L/D for K5: ', num2str(LDmax(5))])

Max L/D for K1: 2.8868
Max L/D for K2: 2.357
Max L/D for K3: 2.0412
Max L/D for K4: 1.8257
Max L/D for K5: 1.6667

%PART D
Kb = 0:.1:10 ;
LDmaxid = (sqrt(Cdo*(Kb.^-1)))/(2*Cdo) ;
Kbest = Cdo/((2*max(LDmaxid)*Cdo)^2) ;
disp(['Best L/D: ', num2str(max(LDmaxid))])
disp(['K Value for Best L/D: ', num2str(Kbest)])
%Okay, so I'm starting to guess that I copied down the location of L/
D max
%equation incorrectly. I have it occurring where Cdo = K*Cl^2, so
hence Cl
%would equation square root of Cdo/K. But this begets a situation
where max
%L/D approaches infinity... huh.
%Nevermind, I actually just derived the equation for the location of
L/Dmax
%and I was right. So I guess I'm more confused than anything now. Is
this a
%trick question? Or, more likely, did I screw up somewhere?

Best L/D: Inf
K Value for Best L/D: 0

%QUESTION 2

RadiusE = 6378 ; %km
RadiusV = 6052 ; %km
a_e = 149.6e6 ; %km
a_v = 108.2e6 ; %km
r_e = 300 ; %km
r_v = 1000 ; %km
mu_e = 398600 ; %km^3/s^2
mu_v = 324900 ; %km^3/s^2
mu_sun = 1.327e11 ; %km^3/s^2
g_e = 9.81 ; %m/s^2

%PART A
%Part 1: Heliocentric Hohmann 1
SpecE_e = -mu_sun/(2*a_e) ;
v_e = sqrt(2*(mu_sun/a_e + SpecE_e)) ;
SpecE_te = -mu_sun/(a_e + a_v) ;

```

---

---

```

v_te = sqrt(2*(mu_sun/a_e + SpecE_te)) ;
v_infe = abs(v_te - v_e) ;

%Part 2: Leaving Earth
SpecE_esc = ((v_infe)^2)/2 ;
v_escpark = sqrt(2*(mu_e/(RadiusE + r_e) + SpecE_esc)) ;
v_parke = sqrt(mu_e/(RadiusE + r_e)) ;
DeltaVBoost = abs(v_escpark - v_parke) ;

%Part 3: Heliocentric Hohmann 2
SpecE_v = -mu_sun/(2*a_v) ;
v_v = sqrt(2*(mu_sun/a_v + SpecE_v)) ;
SpecE_tv = -mu_sun/(a_e + a_v) ;
v_tv = sqrt(2*(mu_sun/a_v + SpecE_tv)) ;
v_infv = abs(v_tv - v_v) ;

%Part 4: Arriving at Venus
SpecE_arr = ((v_infv)^2)/2 ;
v_arrpark = sqrt(2*(mu_v/(RadiusV + r_v) + SpecE_arr)) ;
v_parkv = sqrt(mu_v/(RadiusV + r_v)) ;
DeltaVRetro = abs(v_arrpark - v_parkv) ;
DeltaVTTotal = DeltaVBoost + DeltaVRetro ;
disp(['Total Delta V for Earth to Venus Transfer: ',
    num2str(DeltaVTTotal), ' km/s'])

Total Delta V for Earth to Venus Transfer: 6.6676 km/s

%PART B
m_i = 10000 ; %kg
F_thrust = 900*1000 ; %N
m_flowrate = 300 ; %kg/s
e = 2.718281828 ; %Not sure if natural number is already coded into
    MATLAB... couldn't find it if it is.

Isp = F_thrust/(m_flowrate*g_e) ;
m_f = m_i/(e^((DeltaVTTotal*1000)/(Isp*g_e))) ;
m_fuel = m_i - m_f ;
disp(['Final Mass of HOOPLAH: ', num2str(m_f), ' kg'])
disp(['Mass of Fuel Spent: ', num2str(m_fuel), ' kg'])

%Thanks for good quarter David, hope you enjoy your break

Final Mass of HOOPLAH: 1083.3312 kg
Mass of Fuel Spent: 8916.6688 kg

```

*Published with MATLAB® R2017a*