Table of Contents

```
clc
clear all
close all
```

Quiz Week 2

```
p = 0.2360; % Pressure Ratio (35,000 ft SL)
rho = .953;   % Density ratio sea level (84F hot day)
PAX = 250;
taper = .35;
Tech = 1; % 1 = Conventional, 0 = Adv. Tech
Airfoil = 1; % 1 = Conventional, 0 = Supercritical
TOFL = 9000; % Feet
Range = 6000; %* 6076.12; % Nautical Miles to Feet
fuel_used = 0;
W_Cargo = 8000; % Pounds
Cruise_Alt = 35000; % Feet
M_cruise = 0.80;
SOS = 576; % @ 35,000 ft in knots
V_approach = 140; %* 1.68781; % Knots to ft/s
Engine_Num = 2;
Engine_Type = 0; % 1 = JT8D
N_aisle= 2;
N_abreast = 7;
IN = 1; % Domestic Rules
```

Initialization

```
AR_min = 5;
AR_max = 20;
AR_step = .5;
i_max = (AR_max - AR_min)/AR_step + 1;

sweep_min = 0;
```

```
sweep_max = 30;
sweep_step = 2.5;
j_max = (sweep_max - sweep_min)/sweep_step + 1;
```

Cl Convergence

```
for sweep1 = 10:5:35

    for i = 1:1:i_max

        AR1 = AR_min + (i-1)*AR_step;
%  AR1 = 15;
        AR(i) = AR1;

        Aj_f=0;
        Range_allout = 1;
        V_cruise = M_cruise*SOS; % Cruise Velocity
        R_ao = Range + 200 + V_cruise*.75;

        while abs(R_ao - Range_allout) > 10

            Cl = .5; % Guess Cl
            Cl_f = .1;

            while abs(Cl_f - Cl) > .05
                if Airfoil == 1
                    DeltaM_div = -0.3496*Cl + 0.1921;
                elseif Airfoil == 0
                    DeltaM_div = 0.8245*Cl^3 - 1.7586*Cl^2 + 1.0304*Cl -
0.1718;
                end
                M_div = (M_cruise + .004) - DeltaM_div;

                if Airfoil == 1
                    if sweep1 < 10
                        TC = ((-0.6171*M_div + 0.5644) - (-0.6348*M_div +
0.5732))*((sweep1 - 5)/(10 - 5)) + (-0.6348*M_div + 0.5732);
                    elseif sweep1 >= 10 && sweep1 < 15
                        TC = ((-0.5933*M_div + 0.5516) - (-0.6179*M_div +
0.5639))*((sweep1 - 10)/(15 - 10)) + (-0.6179*M_div + 0.5639);
                    elseif sweep1 >= 15 && sweep1 < 20
                        TC = ((-0.5667*M_div + 0.5378) - (-0.5933*M_div +
0.5516))*((sweep1 - 15)/(20 - 15)) + (-0.5933*M_div + 0.5516);
                    elseif sweep1 >= 20 && sweep1 < 25
                        TC = ((-0.5323*M_div + 0.5189) - (-0.5667*M_div +
0.5378))*((sweep1 - 20)/(25 - 20)) + (-0.5667*M_div + 0.5378);
                    elseif sweep1 >= 25 && sweep1 < 30
                        TC = ((-0.5047*M_div + 0.5057) - (-0.5323*M_div +
0.5189))*((sweep1 - 25)/(30 - 25)) + (-0.5323*M_div + 0.5189);
                    elseif sweep1 >= 30 && sweep1 < 35
                        TC = ((-0.4673*M_div + 0.4854) - (-0.5047*M_div +
0.5057))*((sweep1 - 30)/(35 - 30)) + (-0.5047*M_div + 0.5057);
                    elseif sweep1 >= 35 && sweep1 < 40
                        TC = ((-0.4292*M_div + 0.4649) - (-0.4673*M_div +
0.4854))*((sweep1 - 35)/(40 - 35)) + (-0.4673*M_div + 0.4854);
                    elseif sweep1 == 40
                        TC = -0.4292*M_div + 0.4649;
                    end
                end

                if Airfoil == 0
                    if sweep1 >= 0 && sweep1 < 5
                        TC = ((15.1147*exp(-6.6227*M_div) +
7.2596*10^6*exp(-28.8216*M_div)) - (0.2809*exp(-2.2459*M_div) +
463.1043*exp(-12.0909*M_div)))*((sweep1 - 0)/(5 - 0)) +
(0.2809*exp(-2.2459*M_div) + 463.1043*exp(-12.0909*M_div));
                    elseif sweep1 >= 5 && sweep1 < 10
                        TC = ((14.4804*exp(-6.4949*M_div) +
2.2228*10^6*exp(-27.1108*M_div)) - (15.1147*exp(-6.6227*M_div) +
7.2596*10^6*exp(-28.8216*M_div)))*((sweep1 - 5)/(10 - 5)) +
(15.1147*exp(-6.6227*M_div) + 7.2596*10^6*exp(-28.8216*M_div));
                    elseif sweep1 >= 10 && sweep1 < 15
                        TC = ((11.5069*exp(-6.0790*M_div) +
2.0833*10^7*exp(-29.8909*M_div)) - (14.4804*exp(-6.4949*M_div) +
2.2228*10^6*exp(-27.1108*M_div)))*((sweep1 - 10)/(15 - 10)) +
(14.4804*exp(-6.4949*M_div) + 2.2228*10^6*exp(-27.1108*M_div));
                    elseif sweep1 >= 15 && sweep1 < 20
                        TC = ((11.7020*exp(-5.9514*M_div) +
8.2563*10^8*exp(-34.9086*M_div)) - (11.5069*exp(-6.0790*M_div) +
2.0833*10^7*exp(-29.8909*M_div)))*((sweep1 - 15)/(20 - 15)) +
(11.5069*exp(-6.0790*M_div) + 2.0833*10^7*exp(-29.8909*M_div));
                    elseif sweep1 >= 20 && sweep1 < 25
                        TC = ((2.1536*10^8*exp(-31.7782*M_div)
+ 11.5437*exp(-5.7878*M_div)) - (11.7020*exp(-5.9514*M_div) +
8.2563*10^8*exp(-34.9086*M_div)))*((sweep1 - 20)/(25 - 20)) +
(11.7020*exp(-5.9514*M_div) + 8.2563*10^8*exp(-34.9086*M_div));
                    elseif sweep1 >= 25 && sweep1 < 30
                        TC = ((116.7257*exp(-8.5913*M_div) +
0.0008*exp(3.6374*M_div)) - (2.1536*10^8*exp(-31.7782*M_div) +
11.5437*exp(-5.7878*M_div)))*((sweep1 - 25)/(30 - 25)) +
(2.1536*10^8*exp(-31.7782*M_div) + 11.5437*exp(-5.7878*M_div));
                    elseif sweep1 >= 30 && sweep1 < 35
                        TC = ((72.9298*exp(-7.5603*M_div) +
2.2249*10^14*exp(-47.0361*M_div)) - (116.7257*exp(-8.5913*M_div) +
0.0008*exp(3.6374*M_div)))*((sweep1 - 30)/(35 - 30)) +
(116.7257*exp(-8.5913*M_div) + 0.0008*exp(3.6374*M_div));
                    elseif sweep1 >= 35 && sweep1 < 40
                        TC = ((1.3588*10^-11*exp(22.6706*M_div) +
2.0835*10^4*exp(-13.7095*M_div)) - (72.9298*exp(-7.5603*M_div) +
2.2249*10^14*exp(-47.0361*M_div)))*((sweep1 - 35)/(40 - 35)) +
((72.9298*exp(-7.5603*M_div) + 2.2249*10^14*exp(-47.0361*M_div)));
                    elseif sweep1 >= 40
                        TC = 1.3588*10^-11*exp(22.6706*M_div) +
2.0835*10^4*exp(-13.7095*M_div);
                    end
                end
                cc = cosd(sweep1).^2.*TC.^2.*AR1;
                Clmax_takeoff = -15.7160*exp(-4.2422*cc) +
16.7680*exp(-2.9683*cc);
                Clmax_landing = 3.3287*exp(0.3480*cc) +
-1.4142*exp(-13.6025*cc); %
                WS_landing = (V_approach/1.3)^2*((rho*Clmax_landing)/296); %
Wing Loading Landing
                R_ao = Range + 200 + .75*V_cruise;

                if Engine_Type == 1 % Select Engine
                    WF_WT = 0.3804*exp(0*R_ao) - 0.3789*exp(-0.0003*R_ao) +
Aj_f;
                elseif Engine_Type == 0
                    WF_WT = (0.3804*exp(0*R_ao) - 0.3789*exp(-0.0003*R_ao))
* 0.7820512 + Aj_f; % Scaled for JT9D
                end

                WS_takeoff = WS_landing/(1-fuel_used*WF_WT);
                WS_IC = WS_takeoff*0.965;
                Cl_f = WS_IC/(1481*M_cruise^2*p);

                if Cl_f>Cl
                    Cl = Cl + 0.01;
                else
                    Cl = Cl - 0.01;
                end
            end
```

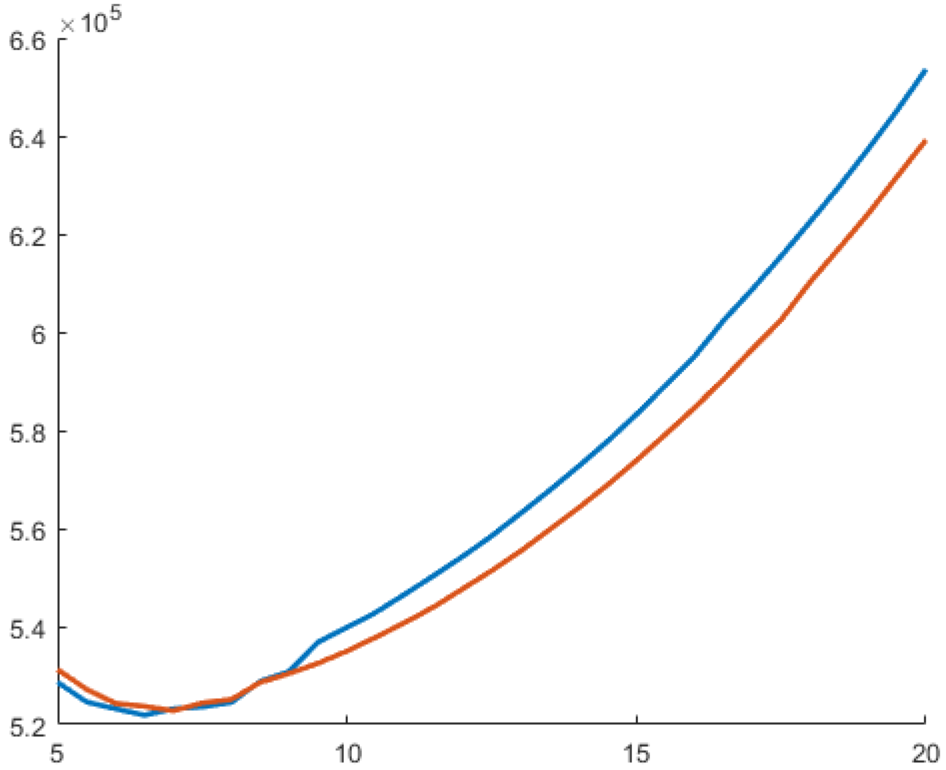
Max Thrust Sizing

```
            if Engine_Num == 2
                X = 28.2045*(TOFL*10^-3) - 8.0939;
                WT_VLO_70 = (X*rho*Clmax_takeoff)/(WS_takeoff);
            end

            V_LO = 1.2*sqrt((296*WS_takeoff)/(rho*Clmax_takeoff));
            M_LO = V_LO/SOS/sqrt(rho);
            M_7LO = .7*M_LO;

            if Engine_Type == 0 % JT9D Selection
                TSLST = 45500;
                T_7M = (40.8149*exp(-1.3944*M_7LO) +
4.6534*exp(1.6605*M_7LO))*1000;
            end

            WT_STATIC = WT_VLO_70*(T_7M/TSLST);
```



Weight Calculations

```
            k_w = 1.00; % Wing Engines
            k_f = 11.5; % PAX > 135
            k_ts = .17; % Wing Engines
            n = 1.5*2.5;

% Wing
            if Tech == 1
                W_Wing = ((.00945*AR1^.8*(1+taper)^.25*k_w*n^.5)/
(TC^.4*cosd(sweep1)*WS_takeoff^.695))*.94;
            elseif Tech == 0
                W_Wing = ((.00945*AR1^.8*(1+taper)^.25*k_w*n^.5)/
(TC^.4*cosd(sweep1)*WS_takeoff^.695))*.70;
            end

% Fuselage
            Fuselage_Length = (3.76*(PAX/N_abreast) + 33.2)*IN;
            Fuselage_Diameter = (1.75*N_abreast + 1.58*N_aisle + 1)*IN;

            if Tech == 1
                W_Fuse =
(.6727*k_f*Fuselage_Length^.6*Fuselage_Diameter^.72*n)*.94;
            elseif Tech == 0
                W_Fuse =
(.6727*k_f*Fuselage_Length^.6*Fuselage_Diameter^.72*n)*.85;
            end

% Landing Gear
            W_LG = .040;

% Nacelle & Pylons
            if Tech == 1
                W_NP = .0555/WT_STATIC;
            elseif Tech == 0
                W_NP = (.0555/WT_STATIC)*0.8;
            end

% Tail Surface
            if Tech == 1
                W_TS = (k_ts*W_Wing)*0.94;
            elseif Tech == 0
                W_TS = (k_ts*W_Wing)*0.85;
            end

% Power Plant
            if Tech == 1
                W_PP = 1/(3.58*WT_STATIC);
            elseif Tech == 0
                W_PP = (1/(3.58*WT_STATIC))*1.10;
            end

% Fuel
            W_F = 1.0275*WF_WT;
            W_Tank = .0175*W_F;
            Fuel_Unusable = .01*W_F;

% Payload
            W_P = PAX*215 + W_Cargo;

% Fixed Equipment
            if Tech == 1
                W_FE =  132*PAX + 300*Engine_Num + 260*2 + 170*(PAX/50);
            elseif Tech == 0
                W_FE =  (132*PAX + 300*Engine_Num + 260*2 + 170*(PAX/
50))*0.9;
            end

            FC = 2;
            CA = PAX/50;

            a = W_Wing;
            B = W_Fuse;
            C = W_LG + W_NP + W_PP + W_F + .035 - 1;
            DD = W_P + W_FE;

            W_TO = 400000; % Guess Weight
            while a*W_TO^1.195 + B*W_TO^.235 + C*W_TO + DD > 1000 % Adjust
tolerances as nessesary; these are obviouly too much
```

```
                a*W_TO^1.195 + B*W_TO^.235 + C*W_TO + DD;
                if a*W_TO^1.195 + B*W_TO^.235 + C*W_TO + DD < 1000
                    W_TO = W_TO - 100; % Adjust increment as nessesary to
prevent over/undershooting correct value
                else
                    W_TO = W_TO + 100;
                end
            end

            W_TO1(i) = W_TO;

            S = W_TO/WS_takeoff;
            b = sqrt(AR1*S);
            c_avg = S/b;
            Thrust = W_TO/WT_STATIC;
            T_e = Thrust/Engine_Num;
```

Drag

```
            Rn = (7.36627*10^-4*V_cruise)/(3.01681*10^-7);

% Wing & Tail
            Rn_wing =Rn*c_avg;
            Cf_wing = (230.7517*Rn_wing^-0.2891 + 1.0836)/1000;
            K_wing = 0.2328*exp(9.1022*TC) - 0.2337*exp(-11.4513*TC);
            Swet_wing = 2*1.02*(S - Fuselage_Diameter*30);
            f_wing = K_wing.*Cf_wing.*Swet_wing;
            f_tail = f_wing*.38;

% Fuselage
            LD_fuselage = Fuselage_Length/Fuselage_Diameter;
            Rn_fuselage = Rn*Fuselage_Length;
            Cf_fuselage = (230.7517*Rn_fuselage^-0.2891 + 1.0836)/1000;
            K_fuselage = 4.8900*exp(-0.9110*LD_fuselage) +
1.3902*exp(-0.0243*LD_fuselage);
            Swet_fuselage = .9*pi*Fuselage_Diameter*Fuselage_Length;
            f_fuselage = Cf_fuselage*Swet_fuselage*K_fuselage;

% Nacelle & Pylons
            Swet_nacelle = 2.1*Engine_Num*(T_e)^.5;
            f_nacelle = 1.25*Cf_wing*Swet_nacelle;
            f_pylon = .20*f_nacelle;

% Total
            f_total = (f_wing + f_fuselage + f_tail + f_nacelle +
f_pylon)*1.06;
            Cd_0 = f_total/S;
            e = 1/(1.035 + .38*Cd_0*pi*AR1);
```

Climb

```
            W_climb = 0.9825*W_TO;
            V_climb = (1.3*12.9)/(f_total*e)^(1/4)*sqrt(W_climb/(.5702*b));
```

```
% Knots
            M_climb = V_climb/SOS;
            T_climb = (.5702*f_total*V_climb^2)/296 + 94.1/
(.5702*.852)*(W_climb/b)^2*(1/(V_climb^2)); % Required Climb Thrust

% Climb Thrust and SFC at 20,000 ft
            if Engine_Type == 0
                if Tech == 1
                    T_a_JT9D_20k = (((-7.7840*M_climb^4 +
15.3111*M_climb^3 - 7.1619*M_climb^2 - 2.7651*M_climb + 16.3830) +
(5.0094*exp(-4.5144*M_climb) + 22.3030*exp(-0.3502*M_climb)))/2)*1000;
                    SFC_20k = ((0.3684*M_climb + 0.3434) + (0.1104*M_climb^2
+ 0.3203*M_climb + 0.3448))/2;
                elseif Tech == 0
                    T_a_JT9D_20k = (((-7.7840*M_climb^4 +
15.3111*M_climb^3 - 7.1619*M_climb^2 - 2.7651*M_climb + 16.3830) +
(5.0094*exp(-4.5144*M_climb) + 22.3030*exp(-0.3502*M_climb)))/2)*1000;
                    SFC_20k = (((0.3684*M_climb + 0.3434) +
(0.1104*M_climb^2 + 0.3203*M_climb + 0.3448))/2)*.9;
                end
            end

            T_a = (T_e/TSLST)*T_a_JT9D_20k;

            RC = (101*V_climb*((T_a*Engine_Num) - T_climb))/W_climb; % ft/min
            time_climb = Cruise_Alt/RC; % min
            range_climb = V_climb*(time_climb/60); % nautical miles
            W_fuel_climb = SFC_20k*Engine_Num*T_a*(time_climb/60); % lbs
```

Range

```
            W_0 = W_TO - W_fuel_climb;
            W_1 = (1 - WF_WT) * W_TO;

            Cl_avg = ((W_0 + W_1)/(2*S))/(1481*p*M_cruise);
            C_Di = Cl_avg^2/(pi*AR1*e);
            C_D = Cd_0 + C_Di + .001;
            LD = Cl_avg/C_D;
            T_req = ((W_0 + W_1)/2)/LD;
            T_req_JTD = (T_req/Engine_Num)*(TSLST/T_e);

            if Engine_Type == 0
                if Tech == 1
                    SFC_35k = 0.9367*exp(-0.5761*(T_req_JTD/1000)) +
0.5352*exp(0.0124*(T_req_JTD/1000));
                elseif Tech == 0
                    SFC_35k = 0.9367*exp(-0.5761*(T_req_JTD/1000)) +
0.5352*exp(0.0124*(T_req_JTD/1000));
                end
            end

            R_cruise = (V_cruise./SFC_35k).*LD.*log(W_0./W_1);
            Range_allout = R_cruise + range_climb;
            WF_WT0 = (W_0-W_1)/W_TO;

            if Range_allout < R_ao
                Aj_f = Aj_f + .0001; % Adjust tolerances as nessesary; these
are obviouly too much
            else
                Aj_f = Aj_f - .0001; % Adjust tolerances as nessesary; these
are obviouly too much
            end

        end
```

Thrust Check

```
        Cl_IC = (W_0/S)/(1481*p*M_cruise^2);
        Cdi_IC = Cl_IC^2/(pi*AR1*e);
        CD_IC = Cd_0 + Cdi_IC + .001;
        LD_IC = Cl_IC/CD_IC;
        Treq_IC = (W_0/LD_IC)/Engine_Num;

        TReqJT9D_IC = Treq_IC*(TSLST/T_e);

        if Engine_Type == 0
            T_avail_cruise = 10000;
        end

        if TReqJT9D_IC > T_avail_cruise
            fprintf('NOT ENOUGH THRUST TOP OF CLIMB')
        end
```

Climb Gradients (TO BE CONTINUED)

```
% 1st Segment %
        Cmb1C_TO = Clmax_takeoff/(1.2^2);
        DeltaCmb1_CD0 = 0.1529*(Cmb1C_TO/Clmax_takeoff)^4 - 0.1377*(Cmb1C_TO/
Clmax_takeoff)^3 + 0.0846*(Cmb1C_TO/Clmax_takeoff)^2 - 0.0701*(Cmb1C_TO/
Clmax_takeoff) + 0.0327;
        Cmb1C_D = Cd_0 + DeltaCmb1_CD0 + .0145 + Cmb1C_TO^2/(pi*AR1*e);
        Cmb1LD_TO = Cmb1C_TO/Cmb1C_D;
        Cmb1T_R = W_TO/Cmb1LD_TO;

        if Engine_Type == 0
            Cmb1T_a = (T_e/TSLST) * T_7M;
        end

        GCmb1Grad = (Engine_Num - 1)*((Cmb1T_a - Cmb1T_R)/W_TO)*100;

        if Engine_Num == 2
            if GCmb1Grad < 0
                fprintf('C 1 Fail')
            end
        elseif Engine_Num == 3
            if GCmb1Grad < .3
```

```
                fprintf('C 1 Fail')
            end
        elseif Engine_Num == 4
            if GCmb1Grad < .5
                fprintf('C 1 Fail')
            end
        end

% 2nd Segment %
        Cmb2C_D  = Cd_0 + DeltaCmb1_CD0 + Cmb1C_TO^2/(pi*AR1*e);
        Cmb2LD_TO = Cmb1C_TO/Cmb2C_D;
        Cmb2T_R  = W_TO/Cmb2LD_TO;
        GCmb2Grad = (Engine_Num - 1)*((Cmb1T_a - Cmb2T_R)/W_TO)*100;

        if Engine_Num==2
            if GCmb2Grad<2.4
                fprintf('C 2 Fail')
            end
        elseif Engine_Num==3
            if GCmb2Grad<2.7
                fprintf('C 2 Fail')
            end
        elseif Engine_Num==4
            if GCmb2Grad<3
                fprintf('C 2 Fail')
            end
        end

% 3rd Segment %
        if sweep1 < 15
            Cmb3Cl_Max = ((-329.6602*TC^3 + 86.4451*TC^2 - 2.7607*TC +
0.8982) - (-316.2418*TC^3 + 81.6642*TC^2 - 2.3432*TC + 0.9419))*((sweep1 -
0)/(15 - 0)) + (-316.2418*TC^3 + 81.6642*TC^2 - 2.3432*TC + 0.9419);
        elseif sweep1 >= 15 && sweep1 < 35
            Cmb3Cl_Max = ((-330.2443*TC^3 + 87.5851*TC^2 - 3.0298*TC +
0.8612) - (-329.6602*TC^3 + 86.4451*TC^2 - 2.7607*TC + 0.8982))*((sweep1 -
15)/(35 - 15)) + (-329.6602*TC^3 + 86.4451*TC^2 - 2.7607*TC + 0.8982);
        elseif sweep1 >= 35
            Cmb3Cl_Max = (-330.2443*TC^3 + 87.5851*TC^2 - 3.0298*TC +
0.8612)*((sweep1 - 35)/(55 - 35)) + (-330.2443*TC^3 + 87.5851*TC^2 -
3.0298*TC + 0.8612);
        end

        Cmb3V = 1.2*sqrt((296*WS_IC)/(.925*Cmb3Cl_Max)); % Altiude
properties at 1000 ft.
        Cmb3M = Cmb3V/SOS;
        Cmb3Cl = Cmb3Cl_Max/(1.2)^2;
        Cmb3C_D = Cd_0 + Cmb3Cl^2/(pi*AR1*e);
        Cmb3LD = Cmb3Cl/Cmb3C_D;
        Cmb3T_R = W_TO/Cmb3LD ;
        Cmb3T_a = T_e/TSLST * 26500;

        GCmb3Grad  = (Engine_Num - 1)*((Cmb3T_a - Cmb3T_R)/W_TO)*100;
        if Engine_Num==2
```

```
            if GCmb3Grad<1.2
                fprintf('C 3 Fail')
            end
        elseif Engine_Num==3
            if GCmb3Grad<1.5
                fprintf('C 3 Fail')
            end
        elseif Engine_Num==4
            if GCmb3Grad<1.7
                fprintf('C 3 Fail')
            end
        end

% Approach %
        ApCl        = Clmax_takeoff/1.3^2;
        Ap_ClClMax = ApCl/Clmax_takeoff;
        ApDeltaCD0 = 0.1529*(ApCl/Clmax_takeoff)^4 - 0.1377*(ApCl/
Clmax_takeoff)^3 + 0.0846*(ApCl/Clmax_takeoff)^2 - 0.0701*(ApCl/
Clmax_takeoff) + 0.0327;
        ApC_D = Cd_0 + ApDeltaCD0 + ApCl^2/(pi*AR1*e);
        ApLD = ApCl/ApC_D;
        ApT_R = (WS_landing*S)/ApLD;
        Ap_V = sqrt((296*WS_landing)/(.953*ApCl)); % Sea Level Hot Day
        ApM = Ap_V/SOS;
        ApTa = T_e/TSLST * 29500;

        GApGrad = (Engine_Num - 1)*((ApTa - ApT_R)/W_TO)*100;

        if Engine_Num==2
            if GApGrad<2.1
                fprintf('Ap Fail')
            end
        elseif Engine_Num==3
            if GApGrad<2.4
                fprintf('Ap 2 Fail')
            end
        elseif Engine_Num==4
            if GApGrad<2.7
                fprintf('Ap 2 Fail')
            end
        end

% Landing %
        LCl = Clmax_landing/1.3^2;
        LClClM = LCl/Clmax_landing;
        LDeltaCD0 = 0.0775*(LCl/Clmax_landing)^3 + 0.0104*(LCl/
Clmax_landing)^2 - 0.0692*(LCl/Clmax_landing) + 0.0412;
        LCD = Cd_0 + LDeltaCD0 + .0145 + LCl^2/(pi*AR1*e);
        LLD = LCl/LCD;
        LT_R = (WS_landing*S)/LLD;
        LV = sqrt((296*WS_landing/(.953*LCl))); % Sea Level Hot Day
        LM = LV/SOS; % Sea Level Hot Day
        LTa = T_e/TSLST * 37200;
        GLGrad = (Engine_Num - 1)*((LTa - LT_R)/W_TO)*100;

        if Engine_Num==2
            if GLGrad<3.2
                fprintf('Landing Fail')
            end
        elseif Engine_Num==3
            if GLGrad<3.2
                fprintf('Landing Fail')
            end
        elseif Engine_Num==4
            if GLGrad<3.2
                fprintf('Landing Fail')
            end
        end
```

```
Ap FailLanding Fail

Ap FailLanding Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Ap FailLanding Fail

Ap FailLanding Fail

Ap FailLanding Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Ap FailLanding Fail
Ap FailLanding Fail
Ap FailLanding Fail
Landing Fail
Landing Fail
Landing Fail
Landing Fail
Landing Fail
Landing Fail
Landing Fail
Landing Fail
Landing Fail
Landing Fail
C 2 FailAp FailLanding Fail
Ap FailLanding Fail
Ap FailLanding Fail
Ap FailLanding Fail
Landing Fail
Landing Fail
Landing Fail
Landing Fail
Landing Fail
Landing Fail
Landing Fail
Landing Fail
Landing Fail
Landing Fail
C 2 FailAp FailLanding Fail
Ap FailLanding Fail
Ap FailLanding Fail
Ap FailLanding Fail
Ap FailLanding Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

C 2 FailAp FailLanding Fail

Ap FailLanding Fail

Ap FailLanding Fail

Ap FailLanding Fail

Ap FailLanding Fail

Ap FailLanding Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail

Landing Fail
```

Direct Operating Cost

```
% Block Speed
        K_a = 1.02;
        D = Range * 1.15; % Statute Miles
        D_CL = range_climb*1.15;
        T_GM = 0.25; % Hours
        T_CL = time_climb/60; % Time to CLimb in Hours
        T_D = 0;
        T_AM = .1;
        T_CR = (D*K_a + 20 - D_CL)/(V_cruise*1.15);

        V_B = D/(T_GM + T_CL + T_D + T_AM + T_CR);

% Block Time
        T_B = T_AM + T_CL + T_D + T_CR + T_GM;

% Block Fuel
        F_CL = W_fuel_climb;
        F_CR_AM = T_req*SFC_35k*(T_CR + T_AM);

        F_B = F_CL + F_CR_AM;

% Flight Operations Cost

% Flight Crew
        P = ((165*PAX+50*PAX)+W_Cargo)/2000; % Tons
        dollar_blockhour = 17.849*(V_B*(W_TO/(1*10^5)))^.3 + 40.83;     %
What is Vc?
        CTM_Crew = dollar_blockhour/(V_B*P);

% Fuel & Oil
        C_F = .40*(1/6.4);
        C_O_T = 2.15; % Cost of Oil
        CTM_Fuel = (1.02*F_B*C_F + Engine_Num*C_O_T*T_B*.135)/(D*P);

% Hull Insurance
        W_A = W_TO*(1-.390) - W_F - W_P - W_TO*(.1046);
        C_A = 2.4*10^6 + 87.5*W_A;
        C_E = 590000 + 16*T_e;
        C_T = C_A + Engine_Num*C_E;
        IR = .01; % Insurance Rate
        U = 630 + 4000/(1 + 1/(T_B + .5));
        CTM_Hull = (IR*C_T)/(U*V_B*P);

% Direct Maintenance

% Airframe-Labor
        K_FHA = 4.9169*log10(W_A/(1*10^3)) - 6.425;
        K_FCA = 0.21256*(log10(W_A/(1*10^3)))^3.7375;
        T_F = T_B - T_AM;
        R_L = 8.60; % Labor Rate
        CTM_AFL = R_L*(K_FHA*T_F + K_FCA)/(V_B*T_B*P);
```

```
% Airframe Material
        C_FHA = (1.5994*C_A)/(1*10^6) + 3.4263;
        C_FCA = (1.9229*C_A)/(1*10^6) + 2.2504;
        CTM_AFM = (C_FHA*T_F + C_FCA)/(V_B*T_B*P);

% Engine-Labor
        K_FHE = (Engine_Num*(T_e/(1*10^3)))/(.82715*(T_e/(1*10^3) + 13.639));
        K_FCE = .20*Engine_Num;

        if Tech == 1
            CTM_EL = R_L*(K_FHE*T_F + K_FCE)/(V_B*T_B*P);
        elseif Tech == 0
            CTM_EL = (R_L*(K_FHE*T_F + K_FCE)/(V_B*T_B*P))*1.10;
        end

% Engine-Material
        C_FHE = Engine_Num*(28.2353*C_E/(1*10^6) - 6.5176);
        C_FCE = Engine_Num*(3.6698*C_E/(1*10^6) + 1.3685);

        if Tech == 1
            CTM_EM = (C_FHE*T_F + C_FCE)/(V_B*T_B*P);
        elseif Tech == 0
            CTM_EM = ((C_FHE*T_F + C_FCE)/(V_B*T_B*P))*1.10;
        end

% Total Maintenance - Burdened
        CTM_TM = (CTM_AFL + CTM_AFM + CTM_EL + CTM_EM)*2;

% Depreciation
        Da = 14; % Years to 10% Value
        CTM_DA = 1/(V_B*P)*(C_T + .06*(C_T - Engine_Num*C_E) +
.3*Engine_Num*C_E)/(Da*U);

% Total DOC

        DOC_Total = CTM_Crew + CTM_Fuel + CTM_Hull + CTM_TM + CTM_DA; % $/
Ton*Mile
        DOC_Passenger = DOC_Total*(P/PAX);

        DOC(i) = DOC_Passenger;

    end
    figure(1)
    hold on
    plot(AR,W_TO1,'LineWidth', 2);
end

legend('Sweep = 15 degs','Sweep = 20 degs','Sweep = 25 degs','Sweep = 30
degs','Sweep = 35 degs')
title('Takeoff Weight vs. Aspect Ratio for Conventional Aircraft')
    xlabel('Aspect Ratio (AR)')
    ylabel('Weight (lbs)')
    grid on
    hold off
```

Published with MATLAB® R2023b