Name	Team Number
Tomoki Koike	R06

AAE 251: Introduction to Aerospace Design

Assignment 4—Orbits, Aircraft Sizing, Lift and Drag

Due Tuesday 19 February, 10:00 am on Blackboard

Instructions

This assignment has four problems—one on Hohmann transfer, one on lift and drag, and two on aircraft sizing. You can work in your teams to develop the sizing code as it will be useful to the aircraft teams in the project. As always, if you work with someone else, please indicate their name(s) on your homework. Start the HW early, or you will run out of time!

Carefully read the lectures notes as they will be helpful to answer the questions. If you have questions, always ask for help from the TA.

Write or type your answers into the appropriate boxes. ***Make sure you submit a single PDF on Blackboard.*** *For the Matlab Code, you can either use Matlab's publishing feature and attach that to your homework or simply copy paste the code in Word and then make a PDF.*

Problem Number	Points Possible	Points Earned
Problem 1	**17**	
Problem 2	**12**	
Problem 3	**8**	
Problem 4	**20**	
Total	**57**	

Problem 1:

NASA wants to move a malfunctioning spacecraft from a circular orbit at 450 km altitude to another circular orbit at 200 km altitude, so a Dragon crew can repair it. Your job as an intern at NASA is to calculate the total ΔV needed to achieve this transfer. Thankfully, you have your notes from AAE251 that illustrate how to calculate the desired velocity. However, keep in mind that in AAE251 we moved from a smaller orbit to a larger one, here NASA is doing the opposite: that is, moving from a larger orbit to a smaller one.

$R_E = 6378\ km$; $\mu_E = 3.986 \times 10^5\ km^3 s^{-2}$

NASA wants you to have a labeled sketch of the transfer, the velocity required to move from larger orbit to the transfer orbit (ΔV_1), the velocity required to move from the transfer orbit to the smaller orbit (ΔV_2), and the ΔV_{total} in your report.

Answer 1:

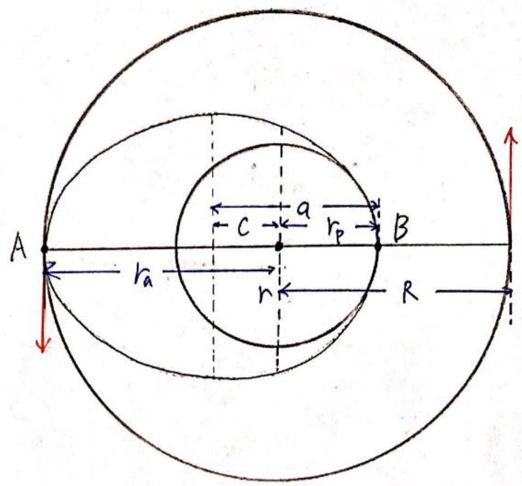
Given $R_e = 6378\ km$, $\mu_e = 3.986 \times 10^5\ km^3/s^2$

radius of outer orbit

$R = 450\ km + R_e$

$= 450\ km + 6378\ km$

$= 6828\ km$



radius of inner orbit

$r = 200\ km + R_e$

$= 6578\ km$

apoapsis of the elliptical transfer orbit

$r_a = R = 6828\ km$

periapsis of the elliptical transfer orbit

$r_p = r = 6578\ km$

because semimajor axis $\equiv a = \frac{r_a + r_p}{2} = \frac{6828 + 6578}{2}\ km$

$a = 6703\ km$

(i) @ point A

the velocity to orbit outer orbit $\equiv v_1 = \sqrt{\frac{\mu_e}{R}} = \sqrt{\frac{3.986 \times 10^5\ km^3/s^2}{6828\ km}}$

$\cong 7.6405\ \frac{km}{s}$

the velocity to enter elliptical orbit $\equiv v_2 = \sqrt{\mu_e\left(\frac{2}{r_a} - \frac{1}{a}\right)}$

$= \sqrt{\left(\frac{3.986\ km^3}{s^2}\right)\left(\frac{2}{6828\ km} - \frac{1}{6703\ km}\right)}$

$= 7.5689\ \frac{km}{s}$

therefore, from outer orbit to elliptical orbit
the velocity change is

$$\Delta V_1 = V_2 - V_1$$
$$= 7.5689 \frac{km}{s} - 7.6405 \frac{km}{s} = -0.0716 \frac{km}{s}$$

(iii) @ point B

the velocity of ellipse @ point $B \equiv V_3 = \sqrt{\mu_e \left(\frac{2}{r_p} - \frac{1}{a}\right)}$

$$= \sqrt{\left(\frac{3.986 \times 10^5\, km^3}{s^2}\right)\left(\frac{2}{6578\, km} - \frac{1}{6703\, km}\right)}$$

$$\cong 7.8566 \frac{km}{s}$$

the velocity to exit elliptical orbit $\equiv V_4 = \sqrt{\frac{\mu_e}{r}}$

$$= \sqrt{\frac{3.986 \times 10^5 \frac{km^3}{s^2}}{6578\, km}}$$

$$\cong 7.7843 \frac{km}{s}$$

therefore, from elliptical orbit to inner orbit
the velocity change is

$$\Delta V_2 = V_4 - V_3 = 7.7843 \frac{km}{s} - 7.8566 \frac{km}{s}$$
$$= -0.0723 \frac{km}{s}$$

Hence, the total velocity change is

$$\Delta V_{Total} = \Delta V_1 + \Delta V_2 = (-0.0716 - 0.0723) \frac{km}{s} = \boxed{-0.1439 \frac{km}{s}}$$

Problem 2:

Consider a rectangular wing with a NACA 2415 airfoil and a chord length of 4 ft. The wing is mounted with a 4 degree angle of attack in the test section of a subsonic wind tunnel, where the air is flowing at 120 mph and the pressure and density are that of the standard atmosphere at sea level. We select the wingspan such that it spans the entire width of the wind tunnel, so the flow essentially "sees" an infinite wing. The wind tunnel force balance measures a lift of 1.1×10^3 lb. What is the width of the wind tunnel? Also calculate the total drag, and the moment about the quarter chord of the wing.

Hint: See Anderson Example 5.1 for guidance on how to solve these types of problems. Always check carefully whether the question is asking for total values, or values per unit span.

Answer 2:

-GIVEN

NACA 2415

- Chord length $\equiv c = 4\ ft$
- angle of attack $\equiv \alpha = 4\ deg$
- free stream velocity $\equiv V_\infty = 120\ mph = 176\ ft/s$
- pressure $\equiv p$ & density $\equiv \rho$ are sea level
- Assume infinite wing
- lift $\equiv L = 1.1 \times 10^3\ lb$

FIND

width of wind-tunnel, t

Total drag, D

moment about the quarter chord, M

SOLN

(a) first we calculate the Reynold's # $\equiv Re$

$$Re = \frac{\rho V_\infty c}{\mu} \quad \text{where } \mu \equiv \text{viscosity} = 3.737\times10^{-7}\ \frac{slug}{ft\cdot s}$$

$$\rho = 0.002378\ \frac{slug}{ft^3}$$

$$Re = \frac{\left(0.002378\ \frac{slug}{ft^3}\right)\left(176\ ft/s\right)\left(4\ ft\right)}{3.737\times10^{-7}\ \frac{slug}{ft\cdot s}} \cong 4.48\times10^6$$

@ this Reynolds #

from the NACA 2415 C_L vs α graph

$$C_L = 0.600$$

and because

$$L = \frac{1}{2}\rho V_\infty^2 C_L S$$

$$S = \frac{2L}{\rho V_\infty^2 C_L} = \frac{2\left(1.1\times10^3\ lb\right)}{\left(0.002378\ \frac{slug}{ft^3}\right)\left(176\ \frac{ft}{s}\right)^2\left(0.600\right)} \cong 49.78\ ft^2$$

Answer 2:

thus, width of wing tunnel L is

$$t = \frac{S}{c} = \frac{49.78\ ft^2}{4\ ft} = 12.45\ ft$$

$$t = \boxed{12.45\ ft}$$

(b)

from NACA 2415 C_d vs C_l graph

@ $C_l = 0.610$ $Re = 4.48 \times 10^6$

approximately $C_d = 0.007$

then

$$D = \frac{1}{2}\rho V_\infty^2 C_d S$$

$$= \frac{1}{2}\left(0.002378 \frac{slug}{ft^3}\right)\left(176 \frac{ft}{s}\right)^2 (0.007)(49.78\ ft^2)$$

$$\approx 12.83\ lb$$

$$D = \boxed{12.83\ lb}$$

(c)

because this is a rectangular wing

moment coefficient $\equiv C_m = -0.2$

$$M = q_\infty S c C_m$$

$$= \frac{1}{2}\rho V_\infty^2 S c C_m$$

$$= \frac{1}{2}\left(0.002378 \frac{slug}{ft^3}\right)\left(171 \frac{ft}{s}\right)^2 (49.78\ ft^2)(4\ ft)(-0.2)$$

$$= -1467\ lb\text{-}ft$$

$$\boxed{-1467\ lb\text{-}ft}$$

Problem 3:

Aircraft design teams will be using this code in their designs. And all of you will use it again in subsequent homeworks, so make sure you code and comment well!

Chapter 3 in Raymer (download from BB) concludes with trade studies for calculating the takeoff weights of an aircraft for ranges of 1000, 1500, and 2000 nautical miles (see Boxes 1, 2, and 3, which summarize the lessons of the entire chapter).

Write a well commented Matlab algorithm that estimates the takeoff weight based on the information from the chapter. Your inputs will be range, payload weight, engine type (see Tables 3.3-4), unit type (metric, English), as well as any other inputs you deem necessary. Paste your code here:

Iterative Sizing Algorithm

- Calculating the Takeoff Weights for Aircrafts Using

Function description

```
function [W_gross, T, Wempty_W0, W0_new] = takeoff_weight_sizing(W_payload, W_crew,...
    range, V_mach, engineType, unitType, category, LDratio, loiterTime1, loiterTime2)
```

INPUTS

1. W_payload: The weight of the payload [lb]
2. W_crew: The weight of the crew [lb]
3. range: The flight range [nautical miles]
4. V_mach: The mach number of the aircraft [Mach]
5. engineType: The type of engine (e.g. pure turbojet, turboprop, etc.)
6. unitType: Metric or English
7. category: The category of the aircraft (e.g. single engine, military cargo, etc.)
8. LDratio: The lift and drag ratio
9. loiterTime1 & loiterTime2: The time of loitering [hr]

OUTPUTS

1. W_gross: The estimated gross weight of the designed aircraft
2. T: The table with the guessed gross weight, Wempty and Wgross ratio, Wempty, and calculated Wgross
3. Wempty_W0: The vector with the outputs of the Wempty and Wgross ratio
4. W0_new: The vector of the outputs of the calculated Wgross

Preparation

Convert based on the unit input and also designate constants based on the engine type that is given.

The constant to determine is the $\text{Specific Fuel Consumption} \equiv C$ for cruise and loiter

```
%By default if the W_crew is undetermined or zero 800 will be the weight of crew
if W_crew == 0
    W_crew = 800; %[lb]
end

if unitType == "metric"
    %Converting the input values to metric
    W_payload = W_payload * 0.4536; %[lb] to [kg]
    W_crew = W_crew * 0.4536; %[lb] to [kg]
    range = range * 1852; %[nmi] to [m]
    %Converting inputs to proper units
    loiterTime1 = loiterTime1 * 3600; %[hr] to [s]
    loiterTime2 = loiterTime2 * 3600; %[hr] to [s]
    V = V_mach * 343; %[m/s] (approximate to sea level)
```

```
%Assigning proper specific fuel consumption values for each engine type
if engineType == "pure turbojet"
    C_cruise = 25.5; %Specific fuel consumption for cruise [mg/Ns]
    C_loiter = 22.7; %Specific fuel consumption for loiter [mg/Ns]
    %Convert to proper units
    C_cruise = C_cruise * 10^(-6); %[kg/Ns]
    C_loiter = C_loiter * 10^(-6); %[kg/Ns]
elseif engineType == "low-bypass turbofan"
    C_cruise = 22.7; %Specific fuel consumption for cruise [mg/Ns]
    C_loiter = 19.8; %Specific fuel consumption for loiter [mg/Ns]
    %Convert to proper units
    C_cruise = C_cruise * 10^(-6); %[kg/Ns]
    C_loiter = C_loiter * 10^(-6); %[kg/Ns]
elseif engineType == "high-bypass turbofan"
    C_cruise = 14.1; %Specific fuel consumption for cruise [mg/Ns]
    C_loiter = 11.3; %Specific fuel consumption for loiter [mg/Ns]
    %Convert to proper units
    C_cruise = C_cruise * 10^(-6); %[kg/Ns]
    C_loiter = C_loiter * 10^(-6); %[kg/Ns]
elseif engineType == "piston-prop fixed pitch"
    C_cruise = 0.068; %Specific fuel consumption for cruise [mg/W-s]
    C_loiter = 0.085; %Specific fuel consumption for loiter [mg/W-s]
    %Convert to proper units
    C_cruise = C_cruise * 10^(-6) * V / 0.7; %[kg/Ns]
    C_loiter = C_loiter * 10^(-6) * V / 0.7; %[kg/Ns]
elseif engineType == "piston-prop variable pitch"
    C_cruise = 0.068; %Specific fuel consumption for cruise [mg/W-s]
    C_loiter = 0.085; %Specific fuel consumption for loiter [mg/W-s]
    %Convert to proper units
    C_cruise = C_cruise * 10^(-6) * V / 0.8; %[kg/Ns]
    C_loiter = C_loiter * 10^(-6) * V / 0.8; %[kg/Ns]
else %Turboprop
    C_cruise = 0.085; %Specific fuel consumption for cruise [mg/W-s]
    C_loiter = 0.101; %Specific fuel consumption for loiter [mg/W-s]
    %Convert to proper units
    C_cruise = C_cruise * 10^(-6) * V / 0.8; %[kg/Ns]
    C_loiter = C_loiter * 10^(-6) * V / 0.8; %[kg/Ns]
end

%The identification of the category is essential to estimate the empty weight
%fraction
if category == "military cargo" || category == "military bomber"
    %Variable sweep constant Kvs
    Kvs = 1.00;
    %Other constants
    A = 0.88;
    c = -0.07;
elseif category == "general aviation single engine"
    %Variable sweep constant Kvs
    Kvs = 1.00;
    %Other constants
    A = 2.05;
    c = -0.18;
```

```
    elseif category == "general aviation twin engine"
        %Variable sweep constant Kvs
        Kvs = 1.00;
        %Other constants
        A = 1.4;
        c = -0.10;
    elseif category == "jet transport"
        %Variable sweep constant Kvs
        Kvs = 1.04;
        %Other constants
        A = 0.97;
        c = -0.06;
    end

else %English unit
    %Converting the input value to proper unit
    loiterTime1 = loiterTime1 * 3600; %[hr] to [s]
    loiterTime2 = loiterTime2 * 3600; %[hr] to [s]
    range = range * 6076.12; %[nmi] to [m]
    V = V_mach * 994.8; %[ft/s] (crude estimate of altitude 30000ft/9144m)

    %Assigning proper specific fuel consumption values for each engine type
    if engineType == "pure turbojet"
        C_cruise = 0.9; %Specific fuel consumption for cruise [1/hr]
        C_loiter = 0.8; %Specific fuel consumption for loiter [1/hr]
        %Convert to proper units
        C_cruise = C_cruise / 3600; %[1/s]
        C_loiter = C_loiter / 3600; %[1/s]
    elseif engineType == "low-bypass turbofan"
        C_cruise = 0.8; %Specific fuel consumption for cruise [1/hr]
        C_loiter = 0.7; %Specific fuel consumption for loiter [1/hr]
        %Convert to proper units
        C_cruise = C_cruise / 3600; %[1/s]
        C_loiter = C_loiter / 3600; %[1/s]
    elseif engineType == "high-bypass turbofan"
        C_cruise = 0.5; %Specific fuel consumption for cruise [1/hr]
        C_loiter = 0.4; %Specific fuel consumption for loiter [1/hr]
        %Convert to proper units
        C_cruise = C_cruise / 3600; %[1/s]
        C_loiter = C_loiter / 3600; %[1/s]
    elseif engineType == "piston-prop fixed pitch"
        C_cruise = 0.4; %Specific fuel consumption for cruise [lb/hr-bhp]
        C_loiter = 0.5; %Specific fuel consumption for loiter [lb/hr-bhp]
        %Convert to proper units
        C_cruise = C_cruise * V / 550 / 0.7 / 3600; %[1/s]
        C_loiter = C_loiter * V / 550 / 0.7 / 3600; %[1/s]
    elseif engineType == "piston-prop variable pitch"
        C_cruise = 0.4; %Specific fuel consumption for cruise [lb/hr-bhp]
        C_loiter = 0.5; %Specific fuel consumption for loiter [lb/hr-bhp]
        %Convert to proper units
        C_cruise = C_cruise * V / 550 / 0.8 / 3600; %[1/s]
        C_loiter = C_loiter * V / 550 / 0.8 / 3600; %[1/s]
    else %Turboprop
```

```
        C_cruise = 0.5; %Specific fuel consumption for cruise [lb/hr-bhp]
        C_loiter = 0.6; %Specific fuel consumption for loiter [lb/hr-bhp]
        %Convert to proper units
        C_cruise = C_cruise * V / 550 / 0.8 / 3600; %[1/s]
        C_loiter = C_loiter * V / 550 / 0.8 / 3600; %[1/s]
    end

    %The identification of the category is essential to estimate the empty weight
    %fraction
    if category == "military cargo" || category == "military bomber"
        %Variable sweep constant Kvs
        Kvs = 1.00;
        %Other constants
        A = 0.93;
        c = -0.07;
    elseif category == "general aviation single engine"
        %Variable sweep constant Kvs
        Kvs = 1.00;
        %Other constants
        A = 2.36;
        c = -0.18;
    elseif category == "general aviation twin engine"
        %Variable sweep constant Kvs
        Kvs = 1.00;
        %Other constants
        A = 1.52;
        c = -0.10;
    elseif category == "jet transport"
        %Variable sweep constant Kvs
        Kvs = 1.04;
        %Other constants
        A = 1.02;
        c = -0.06;
    end
end

%Depending on if the engine is jet engine or propeller the maximum value of L/D
%ratio differs; therefore, we must adjust it depending on the engineType
if engineType == "pure turbojet" || engineType == "low-bypass turbofan"...
        || engineType == "high-bypass turbofan"
    LDratio_cruise = 0.866 * LDratio;
    LDratio_loiter = LDratio;
else %propellers
    LDratio_cruise = LDratio;
    LDratio_loiter = 0.866 * LDratio;
end
```

Main

Now that the preparations are complete, the next step is the main part of this program.

In this part, there will be an algorithm that will initialize a loop with a guessed W-gross value and calculate a new W_gross_new value and compare this value with the guessed W_gross. The condition of the loop will be that until the W_gross and W_gross_new have an 0.0001 accuracy.

```
%Setting the temporary/initial W_gross, W_gross_new, and W_save to initiate the loop
if unitType == "metric"
    W_gross_temp = 22679.619; %[kg]
    W_save = 22679.619; %[kg]
    W_gross_new = 10000000; %[kg]
else %English
    W_gross_temp = 50000; %[lb]
    W_save = 50000; %[lb]
    W_gross_new = 10000000; %[lb]
end
```

These are the thoeries implemented in the loop

$$W_{\text{gross}} = \frac{W_{\text{crew}} + W_{\text{payload}}}{1 - \dfrac{W_{\text{fuel}}}{W_{\text{gross}}} - \dfrac{W_{\text{empty}}}{W_{\text{gross}}}} \qquad \cdots (1)$$

$$\frac{W_{\text{empty}}}{W_{\text{gross}}} = AW_{\text{gross}}^{C}K_{\text{vs}} \qquad \cdots (2)$$

$$\frac{W_{\text{mission}}}{W_{\text{gross}}} = \frac{W_{\text{end-to}}}{W_{\text{start-to}}} \cdot \frac{W_{\text{end-cl}}}{W_{\text{start-cl}}} \cdot \frac{W_{\text{end-cr}}}{W_{\text{start-cr}}} \cdot \frac{W_{\text{end-lr}}}{W_{\text{start-lr}}} \cdot \frac{W_{\text{end-ds}}}{W_{\text{start-ds}}} \cdot \frac{W_{\text{end-land}}}{W_{\text{start-land}}} \qquad \cdots (3)$$

$$\frac{W_{\text{fuel}}}{W_{\text{gross}}} = 1.06\left(1 - \frac{W_{\text{mission}}}{W_{\text{gross}}}\right) \qquad \cdots (4)$$

```
%Weight for the mission
%Warmup and takeoff
%From the table2 in chapter 3 Raymer
W1_W0 = 0.97; %[ratio unitless]

%Climb
%From the table2 in chapter 3 Raymer
W2_W1 = 0.985; %[ratio unitless]

%Cruise
W3_W2 = exp((-1) * range * C_cruise / (V * LDratio_cruise));

%Loiter
%Set conditions for where there is no loitering
if loiterTime1 ~= 0
    W4_W3 = exp((-1) * loiterTime1 * C_loiter / LDratio_loiter);
else
    W4_W3 = 1;
end

%Cruise (second)
```

```
W5_W4 = W3_W2;

%Loiter (second)
%Set conditions for where there is no loitering
if loiterTime2 ~= 0
    W6_W5 = exp((-1) * loiterTime2 * C_loiter / LDratio_loiter);
else
    W6_W5 = 1;
end

%Landing
%From the table2 of Raymer Chapter3
W7_W6 = 0.995; %[ratio unitless]

%Fuel fraction
%Calculate the W_mission and W_gross fraction
W7_W0 = W1_W0*W2_W1*W3_W2*W4_W3*W5_W4*W6_W5*W7_W6;
%Using the above value calculate the fuel fraction
Wf_W0 = 1.06 * (1 - W7_W0);

%Initialize the arrays
%Preallocate
W0_guess = zeros(100,1);
Wempty_W0 = zeros(100,1);
Wempty = zeros(100,1);
W0_new = zeros(100,1);

%Initialize counter
n = 1;

%Loop: The condition to exit this loop is going to be that the calculated W_gross_new must
%be within 0.0001 accuracy of the temporary guessed W_gross_temp value
while (W_gross_temp - W_gross_temp*0.0001) >= W_gross_new || ...
        W_gross_new >= (W_gross_temp + W_gross_temp*0.0001)

    %Updating the temporary W_gross
    W_gross_temp = W_save;

    %Empty weight fraction
    We_W0 = A * Kvs * W_gross_temp^(c);

    %Finally the new gross weight is computed
    W_gross_new = (W_crew + W_payload) / (1 - Wf_W0 - We_W0);

    %Save the value for W_gross_new
    W_save = W_gross_new;

    %Creating a table with the data (inserts data at end)
    W0_guess(n) = W_gross_temp;
    Wempty_W0(n) = We_W0;
    Wempty(n) = We_W0 * W_gross_temp;
    W0_new(n) = W_gross_new;
```

```
    %Increment counter
    n = n + 1;
end
```

After this loop ends we have a final answer which is the W_gross or the optimal takeoff weight for the designed aircraft

```
W_gross = W_gross_new; %Assigning the final output
```

Additionally we will create a table for that will give all of the data throughout the process of the iterative loop

```
%Creating table
%Rule out the trivial rows
W0_guess = W0_guess(W0_guess > 0);
Wempty_W0 = Wempty_W0(Wempty_W0 > 0);
Wempty = Wempty(Wempty > 0);
W0_new = W0_new(W0_new > 0);
%Table
T = table(W0_guess, Wempty_W0, Wempty, W0_new)
%This table will also be an output
```

Terminate function

```
end
```

Using the antisubmarine warfare aircraft example for other parameters and the information in Boxes 1, 2 and 3 as a test case for your sizing code, report the converged calculated gross vehicle take-off weight (W_{gross}) for a range requirement of:

1) 3000 n mi and a payload of 15000 lb., Mach = 0.75

   Assume no loitering

W0_guess	Wempty_W0	Wempty	W0_new
50000	0.43607	21803	1.2355e+05
1.2355e+05	0.40931	50569	1.0124e+05
1.0124e+05	0.41506	42019	1.0532e+05
1.0532e+05	0.41391	43593	1.0448e+05
1.0448e+05	0.41414	43269	1.0465e+05
1.0465e+05	0.4141	43334	1.0461e+05
1.0461e+05	0.41411	43321	1.0462e+05

```
ans =

   1.0462e+05
```

Answer : 1.0462e+05 lb

2) 1500 n mi and a payload of 30,000 lb., Mach = 0.75

W0_guess	Wempty_W0	Wempty	W0_new
50000	0.43607	21803	1.0247e+05
1.0247e+05	0.41471	42497	95505
95505	0.41676	39802	96133
96133	0.41656	40045	96074
96074	0.41658	40023	96079

```
ans =

   9.6079e+04
```

Answer: 9.6079e+04 lb

Problem 4:

In this problem you will create your own version of the empty weight fraction chart in Raymer (also shown in the lecture slides). Our aim is to develop a sense of how and why aircraft designs differ for different missions.

Proceed as follows:

Select three aircraft in each of the following categories of aircraft (so you will have 12 aircraft in total):

- General Aviation: Single Engine
- General Aviation: Twin Engine
- Jet Transport
- Military Cargo

You may choose any in-service aircraft you like.

Find the empty weight and gross takeoff weight for each aircraft. Now create a clearly labeled plot of empty weight fraction versus gross takeoff weight. Show and label each aircraft, and within each aircraft type, fit a straight line between your three points.

Discuss your results.

- Do the different types of aircraft occupy different parts of your plot, and why/why not?
- Within a given type of aircraft, does the empty weight fraction vary, and why/why not?
- Are your results similar to Raymer's table, and why/why not?

In this question, you will be graded both on your analysis and on your presentation of your analysis.

Execution

- Problem 4: Executing the algorithm made in the previous problem for 4 types of aircrafts: general aviation-single engine, general aviation-twin engine, jet transport, and military cargo.
- (Assumption) No loitering

The aircrafts are listed below

General Aviation-Single Engine

1. Cessna TTX: Range 1250 nmi, payload 578 lb, 0.312 Mach, piston-prop fixed pitch
2. Cirrus Vision SF50: 796 nmi, payload 1328 lb, 0.45 Mach, high-bypass turbofan
3. Valcanair V1.0: 575 nmi, payload 970 lb, 0.192 Mach, piston-prop fixed pitch

General Aviation-Twin Engine

1. Piper Seneca: 828 nmi, payload 1331 lb, 0.300 Mach, piston-prop fixed pitch
2. Beechcraft B55 Baron: 933 nmi, payload 900 lb, 0.300 Mach, piston-prop fixed pitch
3. Honda Jet HA-420: 1223 nmi, payload 547 lb, 0.633 Mach, high-bypass turbofan

Jet Transport

1. Boeing 737-900: Range 2950 nmi, payload 44622 lb, 0.79 Mach, high-bypass turbofan
2. Boeing 787-8: Range 7355 nmi, 90500 lb, 0.90 Mach, high-bypass turbofan
3. Airbus A380: Range 8000 nmi, 185000 lb, 0.89 Mach, high-bypass turbofan

Military Cargo

1. Kawasaki C-1: Range 1810 nmi, payload 34128 lb, 0.53 Mach, high-bypass turbofan
2. Boeing C-17 Globemaster 3: Range 2420 nmi, 170900 lb, 0.74 Mach, high-bypass turbofan
3. Lockheed C-5 Galaxy: Range 5500 nmi, payload 285000 lb, 0.79 Mach, high-bypass turbofan

Now, callout function created in problem three for each of the aircrafts above. Entering each specification as the inputs.

```
%Cessna TTX
[W_gross1, T1, Wempty_W01, W0_new1, We_W0_final1] = takeoff_weight_sizing(578, 0, 1250, 0.312,
    "general aviation single engine", 11, 0, 0);
```

```
W3_W2 = 0.8193
Wf_W0 = 0.3835
T = 19×4 table
```

	W0_guess	Wempty_W0	Wempty	W0_new
1	1.0000e+18	0.0014	1.3580e+15	2.2402e+03
2	2.2402e+03	0.5887	1.3187e+03	4.9525e+04
3	4.9525e+04	0.3372	1.6698e+04	4.9335e+03
4	4.9335e+03	0.5107	2.5194e+03	1.3024e+04

	W0_guess	Wempty_W0	Wempty	W0_new
5	1.3024e+04	0.4288	5.5847e+03	7.3425e+03
6	7.3425e+03	0.4754	3.4906e+03	9.7677e+03
7	9.7677e+03	0.4516	4.4111e+03	8.3574e+03
8	8.3574e+03	0.4644	3.8816e+03	9.0641e+03
9	9.0641e+03	0.4577	4.1488e+03	8.6795e+03
10	8.6795e+03	0.4613	4.0039e+03	8.8801e+03
11	8.8801e+03	0.4594	4.0796e+03	8.7731e+03
12	8.7731e+03	0.4604	4.0392e+03	8.8295e+03
13	8.8295e+03	0.4599	4.0605e+03	8.7995e+03
14	8.7995e+03	0.4602	4.0492e+03	8.8154e+03
15	8.8154e+03	0.4600	4.0552e+03	8.8070e+03
16	8.8070e+03	0.4601	4.0520e+03	8.8114e+03
17	8.8114e+03	0.4600	4.0537e+03	8.8091e+03
18	8.8091e+03	0.4601	4.0528e+03	8.8103e+03
19	8.8103e+03	0.4601	4.0533e+03	8.8097e+03

```
%Cirrus Vision SF50
[W_gross2, T2, Wempty_W02, W0_new2, We_W0_final2] = takeoff_weight_sizing(1328, 0, 796, 0.45, '
    "general aviation single engine", 12, 0, 0);
```

```
W3_W2 = 0.8655
Wf_W0 = 0.3051
T = 12×4 table
```

	W0_guess	Wempty_W0	Wempty	W0_new
1	1.0000e+18	0.0014	1.3580e+15	3.0681e+03
2	3.0681e+03	0.5563	1.7067e+03	1.5344e+04
3	1.5344e+04	0.4163	6.3882e+03	7.6380e+03
4	7.6380e+03	0.4720	3.6054e+03	9.5466e+03
5	9.5466e+03	0.4535	4.3290e+03	8.8122e+03
6	8.8122e+03	0.4600	4.0540e+03	9.0591e+03
7	9.0591e+03	0.4578	4.1469e+03	8.9719e+03
8	8.9719e+03	0.4586	4.1141e+03	9.0022e+03
9	9.0022e+03	0.4583	4.1255e+03	8.9916e+03
10	8.9916e+03	0.4584	4.1215e+03	8.9953e+03
11	8.9953e+03	0.4583	4.1229e+03	8.9940e+03

	W0_guess	Wempty_W0	Wempty	W0_new
12	8.9940e+03	0.4584	4.1224e+03	8.9944e+03

```
%Valcanair V1.0
[W_gross3, T3, Wempty_W03, W0_new3, We_W0_final3] = takeoff_weight_sizing(970, 0, 575, 0.192, '
    "general aviation single engine", 10.5, 0, 0);
```

```
W3_W2 = 0.9084
Wf_W0 = 0.2284
T = 11×4 table
```

	W0_guess	Wempty_W0	Wempty	W0_new
1	1.0000e+18	0.0014	1.3580e+15	2.2979e+03
2	2.2979e+03	0.5860	1.3465e+03	9.5336e+03
3	9.5336e+03	0.4536	4.3242e+03	5.5651e+03
4	5.5651e+03	0.4997	2.7810e+03	6.5096e+03
5	6.5096e+03	0.4858	3.1625e+03	6.1930e+03
6	6.1930e+03	0.4902	3.0358e+03	6.2894e+03
7	6.2894e+03	0.4888	3.0745e+03	6.2591e+03
8	6.2591e+03	0.4893	3.0623e+03	6.2685e+03
9	6.2685e+03	0.4891	3.0661e+03	6.2656e+03
10	6.2656e+03	0.4892	3.0649e+03	6.2665e+03
11	6.2665e+03	0.4892	3.0653e+03	6.2662e+03

```
%Piper Seneca
[W_gross4, T4, Wempty_W04, W0_new4, We_W0_final4] = takeoff_weight_sizing(828, 0, 828, 0.300, '
    "general aviation twin engine", 13, 0,0);
```

```
W3_W2 = 0.8943
Wf_W0 = 0.2540
T = 14×4 table
```

	W0_guess	Wempty_W0	Wempty	W0_new
1	1.0000e+18	0.0241	2.4090e+16	2.2552e+03
2	2.2552e+03	0.7023	1.5838e+03	3.7273e+04
3	3.7273e+04	0.5305	1.9774e+04	7.5560e+03
4	7.5560e+03	0.6223	4.7022e+03	1.3165e+04
5	1.3165e+04	0.5887	7.7504e+03	1.0351e+04
6	1.0351e+04	0.6030	6.2423e+03	1.1389e+04
7	1.1389e+04	0.5973	6.8027e+03	1.0950e+04
8	1.0950e+04	0.5997	6.5661e+03	1.1126e+04
9	1.1126e+04	0.5987	6.6611e+03	1.1054e+04

	W0_guess	Wempty_W0	Wempty	W0_new
10	1.1054e+04	0.5991	6.6222e+03	1.1083e+04
11	1.1083e+04	0.5989	6.6380e+03	1.1071e+04
12	1.1071e+04	0.5990	6.6316e+03	1.1076e+04
13	1.1076e+04	0.5990	6.6342e+03	1.1074e+04
14	1.1074e+04	0.5990	6.6331e+03	1.1075e+04

```
%Beechcraft B55 Baron
[W_gross5, T5, Wempty_W05, W0_new5, We_W0_final5] = takeoff_weight_sizing(900, 0, 933, 0.300, '
    "general aviation twin engine", 13, 0,0);
```

```
W3_W2 = 0.8817
Wf_W0 = 0.2765
T = 16×4 table
```

	W0_guess	Wempty_W0	Wempty	W0_new
1	1.0000e+18	0.0241	2.4090e+16	2.4307e+03
2	2.4307e+03	0.6971	1.6944e+03	6.4367e+04
3	6.4367e+04	0.5023	3.2332e+04	7.6869e+03
4	7.6869e+03	0.6213	4.7755e+03	1.6631e+04
5	1.6631e+04	0.5751	9.5647e+03	1.1459e+04
6	1.1459e+04	0.5969	6.8401e+03	1.3435e+04
7	1.3435e+04	0.5875	7.8935e+03	1.2504e+04
8	1.2504e+04	0.5918	7.3992e+03	1.2906e+04
9	1.2906e+04	0.5899	7.6131e+03	1.2725e+04
10	1.2725e+04	0.5907	7.5171e+03	1.2805e+04
11	1.2805e+04	0.5903	7.5595e+03	1.2770e+04
12	1.2770e+04	0.5905	7.5406e+03	1.2785e+04
13	1.2785e+04	0.5904	7.5490e+03	1.2778e+04
14	1.2778e+04	0.5905	7.5453e+03	1.2782e+04
15	1.2782e+04	0.5905	7.5469e+03	1.2780e+04
16	1.2780e+04	0.5905	7.5462e+03	1.2781e+04

```
%Honda Jet HA-420
[W_gross6, T6, Wempty_W06, W0_new6, We_W0_final6] = takeoff_weight_sizing(547, 0, 1223, 0.633,
    "general aviation twin engine", 15, 0, 0);
```

```
W3_W2 = 0.8815
Wf_W0 = 0.2770
T = 18×4 table
```

	W0_guess	Wempty_W0	Wempty	W0_new
1	1.0000e+18	0.0241	2.4090e+16	1.9274e+03
2	1.9274e+03	0.7134	1.3750e+03	1.4114e+05
3	1.4114e+05	0.4644	6.5543e+04	5.2092e+03
4	5.2092e+03	0.6459	3.3646e+03	1.7479e+04
5	1.7479e+04	0.5723	1.0002e+04	8.9378e+03
6	8.9378e+03	0.6120	5.4696e+03	1.2134e+04
7	1.2134e+04	0.5935	7.2019e+03	1.0407e+04
8	1.0407e+04	0.6027	6.2723e+03	1.1202e+04
9	1.1202e+04	0.5983	6.7018e+03	1.0804e+04
10	1.0804e+04	0.6005	6.4876e+03	1.0995e+04
11	1.0995e+04	0.5994	6.5907e+03	1.0902e+04
12	1.0902e+04	0.5999	6.5402e+03	1.0947e+04
13	1.0947e+04	0.5997	6.5647e+03	1.0925e+04
14	1.0925e+04	0.5998	6.5528e+03	1.0936e+04
15	1.0936e+04	0.5997	6.5586e+03	1.0931e+04
16	1.0931e+04	0.5998	6.5557e+03	1.0933e+04
17	1.0933e+04	0.5997	6.5571e+03	1.0932e+04
18	1.0932e+04	0.5998	6.5564e+03	1.0932e+04

```
%Boeing 737-900
[W_gross7, T7, Wempty_W07, W0_new7, We_W0_final7] = takeoff_weight_sizing(44622, 0, 2950, 0.79
    "jet transport", 17, 0,0);
```

```
W3_W2 = 0.8064
Wf_W0 = 0.4047
T = 10×4 table
```

	W0_guess	Wempty_W0	Wempty	W0_new
1	1.0000e+18	0.0882	8.8234e+16	8.9578e+04
2	8.9578e+04	0.5352	4.7940e+04	7.5554e+05
3	7.5554e+05	0.4709	3.5579e+05	3.6516e+05
4	3.6516e+05	0.4919	1.7962e+05	4.3932e+05
5	4.3932e+05	0.4865	2.1372e+05	4.1741e+05
6	4.1741e+05	0.4880	2.0369e+05	4.2323e+05
7	4.2323e+05	0.4876	2.0635e+05	4.2163e+05
8	4.2163e+05	0.4877	2.0562e+05	4.2207e+05

	W0_guess	Wempty_W0	Wempty	W0_new
9	4.2207e+05	0.4877	2.0582e+05	4.2195e+05
10	4.2195e+05	0.4877	2.0577e+05	4.2198e+05

```
%These two aircrafts have too high range that the algorithm returns error
%So I will use values on the internet
%Boeing 787-8
% [W_gross8, T8, Wempty_W08, W0_new8, We_W0_final8] = takeoff_weight_sizing(90500, 0, 7355, 0.9
%     "jet transport", 18, 0,0);
W_gross8 = 502,500;
```

```
W_gross8 = 502
```

```
We_W0_final8 = 0.5264;
% %Airbus A380
% [W_gross9, T9, Wempty_W09, W0_new9, We_W0_final9] = takeoff_weight_sizing(185000, 0, 8000, 0
%     "jet transport", 18.5, 0, 0);
W_gross9 = 1234600;
We_W0_final9 = 0.4947;

%Kawasaki C-1
[W_gross10, T10, Wempty_W010, W0_new10, We_W0_final10] = takeoff_weight_sizing(34128, 0, 1810,
    "military cargo", 16,0,0);
```

```
W3_W2 = 0.8113
Wf_W0 = 0.3967
T = 7×4 table
```

	W0_guess	Wempty_W0	Wempty	W0_new
1	1.0000e+18	0.0511	5.1107e+16	6.3250e+04
2	6.3250e+04	0.4290	2.7131e+04	2.0030e+05
3	2.0030e+05	0.3957	7.9260e+04	1.6822e+05
4	1.6822e+05	0.4006	6.7384e+04	1.7226e+05
5	1.7226e+05	0.3999	6.8886e+04	1.7170e+05
6	1.7170e+05	0.4000	6.8677e+04	1.7177e+05
7	1.7177e+05	0.4000	6.8706e+04	1.7176e+05

```
%Boeing C-17
[W_gross11, T11, Wempty_W011, W0_new11, We_W0_final11] = takeoff_weight_sizing(170900, 0, 2420,
    "military cargo", 18, 0,0);
```

```
W3_W2 = 0.8370
Wf_W0 = 0.3541
T = 6×4 table
```

	W0_guess	Wempty_W0	Wempty	W0_new
1	1.0000e+18	0.0511	5.1107e+16	2.8866e+05
2	2.8866e+05	0.3857	1.1134e+05	6.5985e+05

	W0_guess	Wempty_W0	Wempty	W0_new
3	6.5985e+05	0.3640	2.4020e+05	6.0909e+05
4	6.0909e+05	0.3661	2.2296e+05	6.1354e+05
5	6.1354e+05	0.3659	2.2448e+05	6.1313e+05
6	6.1313e+05	0.3659	2.2434e+05	6.1317e+05

```
%Lockheed C-5
[W_gross12, T12, Wempty_W012, W0_new12, We_W0_final12] = takeoff_weight_sizing(285000, 0, 5500
    "military cargo", 18, 0, 0);
```

```
W3_W2 = 0.6846
Wf_W0 = 0.5877
T = 10×4 table
```

	W0_guess	Wempty_W0	Wempty	W0_new
1	1.0000e+18	0.0511	5.1107e+16	7.9120e+05
2	7.9120e+05	0.3594	2.8437e+05	5.4018e+06
3	5.4018e+06	0.3142	1.6972e+06	2.9125e+06
4	2.9125e+06	0.3281	9.5554e+05	3.3925e+06
5	3.3925e+06	0.3246	1.1012e+06	3.2577e+06
6	3.2577e+06	0.3255	1.0605e+06	3.2923e+06
7	3.2923e+06	0.3253	1.0709e+06	3.2832e+06
8	3.2832e+06	0.3253	1.0682e+06	3.2856e+06
9	3.2856e+06	0.3253	1.0689e+06	3.2850e+06
10	3.2850e+06	0.3253	1.0687e+06	3.2851e+06

Polyfit the best-fit line for each type of aircraft as W0_new# on x-axis and We_W0_final# as y-axis

```
%First create vector from the given data to plot
%Single Engine
W0_singEng = [W_gross1 W_gross2 W_gross3]; %x-values
WeFrac_singEng = [We_W0_final1 We_W0_final2 We_W0_final3]; %y-values

%Twin Engines
W0_twinEng = [W_gross4 W_gross5 W_gross6];
WeFrac_twinEng = [We_W0_final4 We_W0_final5 We_W0_final6];

%Jet Transport
W0_JT = [W_gross7 W_gross8 W_gross9];
WeFrac_JT = [We_W0_final7 We_W0_final8 We_W0_final9];

%Military Cargo
W0_MC = [W_gross10 W_gross11 W_gross12];
WeFrac_MC = [We_W0_final10 We_W0_final11 We_W0_final12];
```

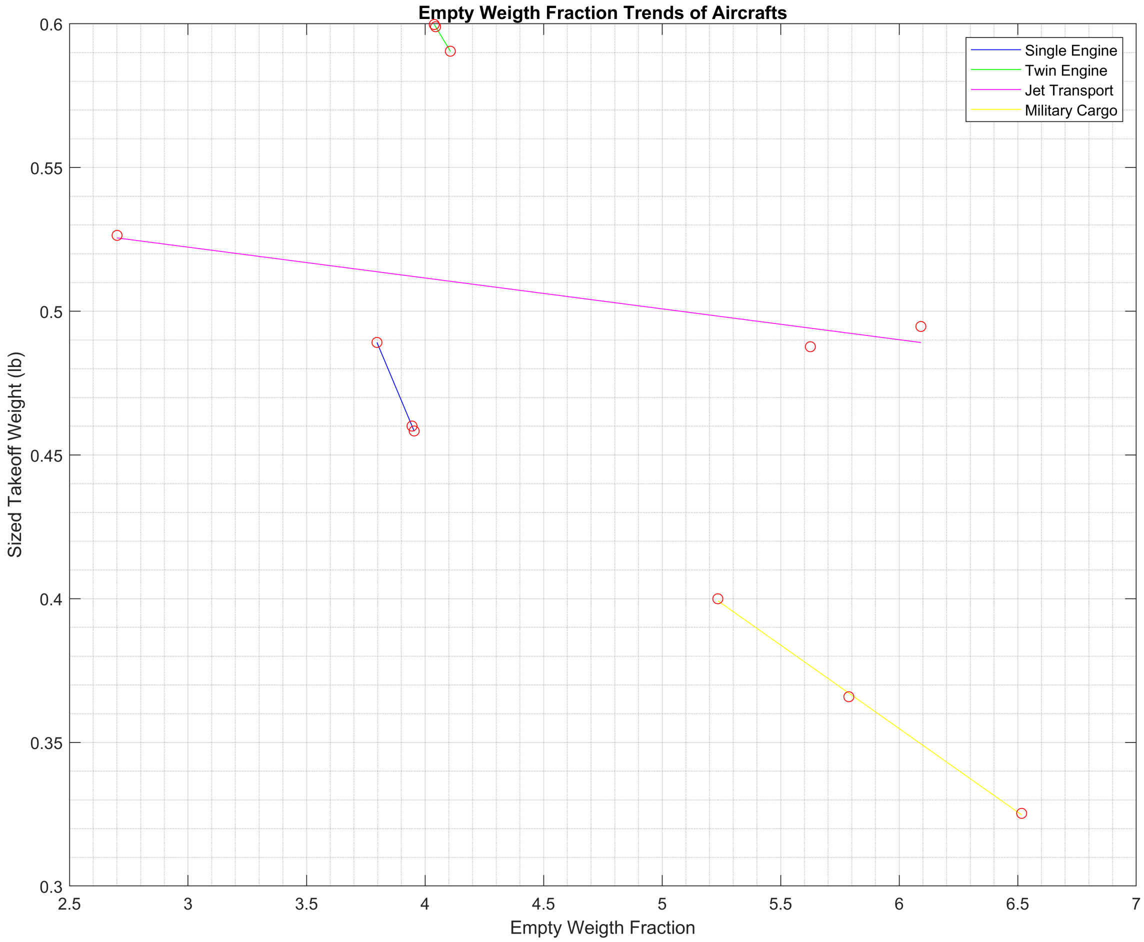
```
%Calibrate the data by semilogX which gives a clean linear
%graph (better than normal plot). Take log10 for all x-values.
%Then polyfit each of them
%Single Engine
W0_singEng = log10(W0_singEng);
p1 = polyfit(W0_singEng, WeFrac_singEng, 1);
fit1 = polyval(p1, W0_singEng);

%Twin Engine
W0_twinEng = log10(W0_twinEng);
p2 = polyfit(W0_twinEng, WeFrac_twinEng,1);
fit2 = polyval(p2, W0_twinEng);

%Jet Transport
W0_JT = log10(W0_JT);
p3 = polyfit(W0_JT, WeFrac_JT,1);
fit3 = polyval(p3, W0_JT);

%Military Cargo
W0_MC = log10(W0_MC);
p4 = polyfit(W0_MC, WeFrac_MC,1);
fit4 = polyval(p4, W0_MC);

%Plotting
figure(1)
plot(W0_singEng, fit1, "-b")
title("Empty Weigth Fraction Trends of Aircrafts")
ylabel("Empty Weigth Fraction")
xlabel("Sized Takeoff Weight (log10(lb))")
grid on
grid minor
box on
hold on
plot(W0_twinEng, fit2, "-g")
plot(W0_JT, fit3, "-m")
plot(W0_MC, fit4, "-y")
plot(W0_singEng, WeFrac_singEng, "or")
plot(W0_twinEng, WeFrac_twinEng, "or")
plot(W0_JT, WeFrac_JT, "or")
plot(W0_MC, WeFrac_MC, "or")
hold off
legend("Single Engine", "Twin Engine", "Jet Transport", "Military Cargo");
set(gcf,'PaperPositionMode','auto','Position',[0 0 1100 850]) % Control where plots are positio
```



Analysis

- Do the different types of aircraft occupy different parts of your plot, and why/why not?

ANS:

Yes, they do. For the jet transport the high values of the range and payload giver larger results for the takeoff weight; but, lower empty weight fractions.

And vice versa for the light single engine and twin engine aircrafts.

- Within a given type of aircraft, does the empty weight fraction vary, and why/why not?

ANS:

Yes, they vary. As stated above the takeoff weight is dependent of the payload and the range, and within a category of aircrafts there are aircrafts with different payloads and ranges even though the flight mission might be alike. With such change of parameters, the empty weight fraction can vary. Such trend can be observed in the graph above for all aircraft categories.

- Are your results similar to Raymer's table, and why/why not?

ANS:

Yes, my graph and the graph on Raymer is somewhat alike. For example, the general aviation-twin engine category range in the empty weight fraction of around 6 with a negative slope. This is the very close to the example in Raymer. Jet transport and military cargo appear at the right hand side and low empty weight ratio (where military cargo ranges lower than jet transport). These trends are similar which makes it reasonable to conclude that the example graph and my graph are similar to each other.