Session 2 Pork-Chop Plots Matlab Guide

Exercise 2. Create a script that computes the total Δv for all the possible transfers of the following grids of departure and arrival times.

```
%-----------------------------------------------
% Earth to Mars from 14/3/16 to 15/10/2016
InTime=date2mjd2000([2016 03 14 0 0 0]);
FinTime=date2mjd2000([2016 10 15 0 0 0]);

% Generate Grid of conditions to analyse
Range=3*30; % days
DepartureGrid=InTime-Range:5:InTime+Range;
ArrivalGrid=FinTime-Range:5:FinTime+Range;
```

Plot these results as a colour map, where X and Y axis show departure and arrival times and colour denotes total Δv.

Note

1. Make sure AMAII-Toolbox is in the Matlab *path*.

2. Initialize the constants that you will need for this exercise. Since these are interplanetary transfers, it is likely that you will need the gravitational constant of the Sun μ_{Sun}.

3. Initialize a matrix where you will store all the Δv solutions by assigning it an array of zeros, using the function *zeros*.

4. Create a nested for-loop, such as:

```
for m = 1:j
   for n = 1:k
      <statements>;
   end
end
```

5. Inside the nested for-loop, compute each instance of the Lambert arc by solving the Lambert arc function:

$$[v_1, v_2] = Lambert\left(r_1, r_2, ToF, t_m, \mu_{Sun}\right)$$

where *lambert* should be the result of Session 1 Part B Exercise 7.

6. Note that *Lambert* requires inputting the flag t_m, which either defines the Lambert arc as a short trajectory or as a long one (see slide 23 of session 1). Hence, in order to ensure all relevant cases are computed, at each iteration of the nester for-loop both cases need to be computed.

$$[v_1^{short}, v_2^{short}] = Lambert\left(r_1, r_2, ToF, +1, \mu_{Sun}\right)$$
$$[v_1^{long}, v_2^{long}] = Lambert\left(r_1, r_2, ToF, -1, \mu_{Sun}\right)$$

7. Compute the Δv for each of the two options:

$$\Delta v^{short} = \left| v_1^{short} - v_{Earth} \right| + \left| v_{Mars} - v_2^{short} \right|$$

$$\Delta v^{long} = \left| v_1^{long} - v_{Earth} \right| + \left| v_{Mars} - v_2^{long} \right|$$

8. Store only the optimal Δv (i.e. minimum) of the two.

```
for iD=1:length(DepartureGrid)
    for iA=1:length(ArrivalGrid)
        % For each Departure date and each arrival date the Lambert arc is
        % computed.
%            [...]
        % only the optimal DV of the two options is stored
        DV_solutions(iD,iA)=min([DV_Short DV_long]);
    end
end
```

9. Plot the results using contourf.

Exercise 3. Compute the actual total DV of ExoMars TGO transfer assuming an initial circular orbit at the Earth of 250 km altitude, and a final Mars operational orbit 400 km altitude circular orbit.

1. Solve the lambert arc for the ExoMars transfer as described previously (i.e. Earth departure on 14/03/2016 and Mars arrival on the 15/10/2016).

$$[v_1, v_2] = Lambert\left(r_1, r_2, ToF, t_m, \mu_{Sun}\right).$$

2. Compute the excess velocities between the transfer and Earth and Mars:

$$v_\infty^{dep} = \left|v_1^{short} - v_{Earth}\right|$$
$$v_\infty^{arr} = \left|v_{Mars} - v_2^{Short}\right|$$

3. Compute the actual manoeuvres for departure and arrival as described in Question 1 – Worked Examples 4 from AMA module. A summary of the equations needed follow:

Earth Departure

$$v_p = \sqrt{\mu_\oplus\left(\frac{2}{r_p} + \frac{v_\infty^2}{\mu_\oplus}\right)} \qquad v_{cir} = \sqrt{\frac{\mu_\oplus}{r_p}}$$

$$\Delta v_1 = v_p - v_{cir}$$

Mars Arrival

$$v_{cirM} = \sqrt{\frac{\mu_M}{r_{pM}}} \qquad v_{pM} = \sqrt{\mu_M\left(\frac{2}{r_{pM}} + \frac{v_\infty^2}{\mu_M}\right)}$$

$$\Delta v_2 = v_{pM} - v_{cirM}$$

$$\Delta v_{total} = \Delta v_1 + \Delta v_2$$

Exercise 4. Perform a complete assessment of launch opportunities using a Proton direct insertion for launch opportunities during 2019 to 2020. Assume a main propulsion system for the spacecraft with an Isp of 310s. How much mass can we insert into a 400km circular orbit? If ExoMars TGO had a dry mass of 1500 kg, what are the launch window opportunities?

Note: Since the objective is to insert ExoMars TGO in its operational circular orbit, a launch opportunity will be defined as any launch date that allows you to insert ExoMars dry mass into the 400 km circular orbit. The inclination of the Mars circular orbit is not constraint (which is to say that is ignored at this stage).

Hence, the objective is to construct a script or a function that given a departure date and a time of flight (or an arrival date) it computes how much mass Proton inserts into interplanetary orbit. Next, with this mass and the Isp, the final mass into the final Mars orbit can be computed.

Note that Proton launch performance are provided in slide 23.

1. Make sure AMAII-Toolbox is in the Matlab *path*.

2. Initialize the constants that you will need for this exercise. You will certainly need the Sun μ_{Sun}, Mars μ_{Mars}, and Mars' radius R_M.

3. Initialize a matrix where you will store all the Final Mass solutions by assigning it an array of zeros, using the function *zeros*.

4. Create a nested for-loop, such as:

```
for m = 1:j
   for n = 1:k
      <statements>;
   end
end
```

5. Inside the nested for-loop, compute each instance of the Lambert arc by solving the Lambert arc function:

$$[v_1, v_2] = Lambert\left(r_1, r_2, ToF, t_m, \mu_{Sun}\right)$$

6. Be watchful of the two types of transfers you can compute; the long path transfer and the short path transfer. Use the sum of the Excess velocities to distinguish whish one is a better option. Hence, at each iteration of the nester for-loop both cases need to be computed.

$$[v_1^{short}, v_2^{short}] = Lambert\left(r_1, r_2, ToF, +1, \mu_{Sun}\right)$$
$$[v_1^{long}, v_2^{long}] = Lambert\left(r_1, r_2, ToF, -1, \mu_{Sun}\right)$$

7. Compute the Δv for each of the two options:

$$\Delta v^{short} = \left|v_1^{short} - v_{Earth}\right| + \left|v_{Mars} - v_2^{short}\right|$$
$$\Delta v^{long} = \left|v_1^{long} - v_{Earth}\right| + \left|v_{Mars} - v_2^{long}\right|$$

8. Identify which one of the two offers a better transfer option:

```
% Check which of the two transfers is optimal

[~,indexMin]=min([DV_Short DV_long]);
```

9. Using the relevant excess velocity v_∞, compute the wet mass that proton can launch:

```
%-------------------------------------------------------------------------
% Compute Departure mass using v_inf_Earth
if indexMin==1
    r1dot=r1dot_Short;
    r2dot=r2dot_Short;
else
    r1dot=r1dot_Long;
    r2dot=r2dot_Long;
end

v_inf_Earth=norm(vEarth-r1dot);
C3_vector=[0^2 0.5^2  1^2 1.5^2 2^2 2.5^2 3^2 3.5^2 4^2 4.5^2 5^2 5.5^2 6^2];

WetMass_vector=[6475 6445 6355 6205 6002 5748 5454 5124 4745 4361 3971 3556 3111];

C3_Ex4=v_inf_Earth^2;
massWet = interp1(C3_vector,WetMass_vector,C3_Ex4);
%-----------------------------------------------------------------
```

10. Compute the final mass at arrival at the Mars' operational orbit. Recall:

Mars Arrival

$$v_{cirM} = \sqrt{\frac{\mu_M}{r_{pM}}} \qquad v_{pM} = \sqrt{\mu_M\left(\frac{2}{r_{pM}} + \frac{v_\infty^2}{\mu_M}\right)}$$

$$\Delta v_2 = v_{pM} - v_{cirM}$$

11. Store the final mass at completion of each loop iteration

```
% Nested for-loop to scan through all the potential Lambert arcs
for iD=1:length(LaunchWindow)
    for iA=1:length(TOF)
%       [...]
        %-----------------------------------------------------------------
        % Storing Solutions
        FinalMass_Matrix(iD,iA)=Mass_final;
    end
end
```

9. Plot the results using *contourf*.