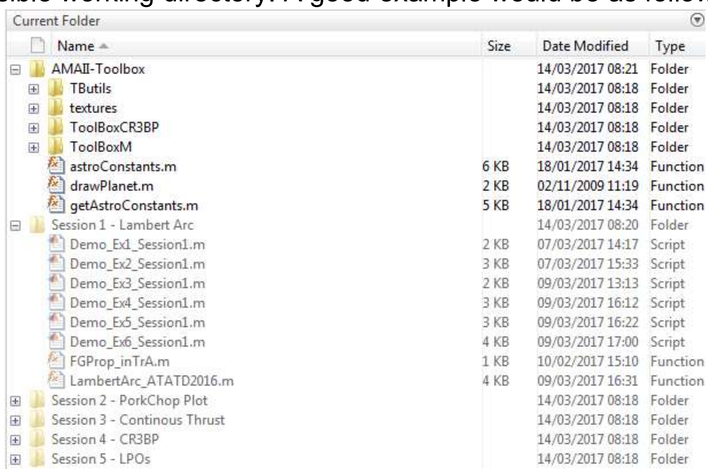
Basic Notes

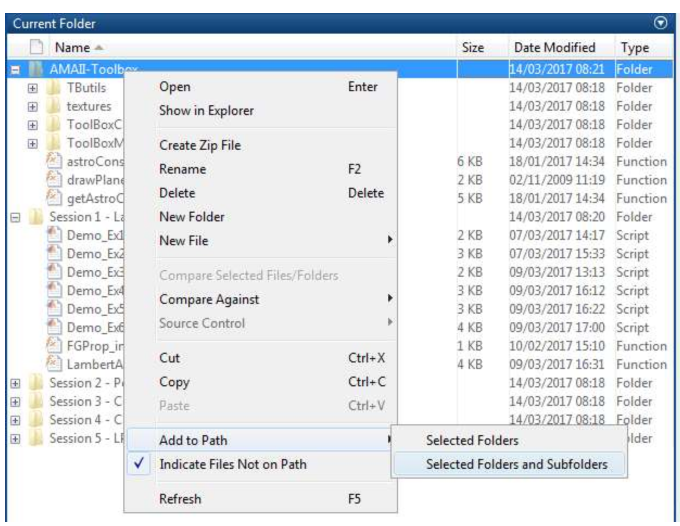
The following bullet points provide some basic hints on how completing the tasks of this course.

- Complete all the exercises by creating one script for each task. This way you will be able to revise easily your calculations, as well as comparing with the exercise solutions. To know more on MATLAB scripts click here.
- Create a sensible working directory. A good example would be as follow:



Note that AMAII-Toolbox is highlighted, while the other folders seem to be more shaded. This is simply because AMAII-Toolbox is already in the path of the working environment.

- In order to add the toolbox folder in the path do as follow:
  1. Righ-click over the folder you want to add into the path (AMAII-Toolbox).
  2. Look for the option “add to path” and hover with the cursor over it.
  3. The option “select folder and subfolders” will appear. Click on it.



Session 1 Lambert Arc Matlab Guide

Exercise 1. ExoMars Trace Gas Orbiter departed Earth on 14/03/2016, and arrived at Mars on 15/10/2016. What is the minimum energy orbit that would link the position of Earth and Mars those two days? What is the time of flight of the minimum energy orbit? Was this the trajectory followed by ExoMars TGO?

1. Add AMAII-Toolbox to Matlab <u>*path*</u>.

2. Become familiar with *EphSS_car* function and its usages.

```
fx >> open EphSS_car
```

3. Become familiar with *date2mjd2000* function and its usages.

4. Use *date2mjd2000* to convert from dates to times in MJD2000, which is the date unit required for *EphSS_car* to give you the positions of Earth and Mars.

5. With the position of the Earth and Mars as $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ compute the semi-major axis and eccentricity of the minimum energy orbit following algorithm in slide 19 of session 1.

6. Compute the time of flight for the short time transfer. Note that you will need the gravitational constant μ of the Sun, use the following instruction to obtain it:

```
%-------------------------------------
muSun=getAstroConstants('Sun','mu');
%-------------------------------------
```

7. Was this the trajectory followed by ExoMars TGO?

Exercise 2. ExoMars Trace Gas Orbiter departed Earth on 14/03/2016, and arrived at Mars on 15/10/2016. Program the F and G solutions to the two body problem. Verify the answer by comparing it to a numerical integration of the differential equations of motion:

$$\boxed{\ddot{\mathbf{r}} + \frac{\mu_E}{r^3}\mathbf{r} = 0}$$

Using the F and G techniques, pot the orbit of Mars, Earth and the minimum energy transfer for ExoMars TGO.

1. Ensure AMAII-Toolbox to Matlab *path*.

2. Use algorithm as described in slide 32 to compute the departure velocity $\dot{\mathbf{r}}_1$.

3. Construct a function in Matlab that provides the functionalities of algorithm *FGKepler_trA* (slide 33).

If you have never created a function in Matlab, check the following tutorial:

https://uk.mathworks.com/help/matlab/matlab_prog/create-functions-in-files.html

4. Using a for-loop now, you can plot the orbits of the Earth, Mars. Note: function FGProp_trA is my implementation of algorithm FGKepler.

```
nPoints=100; % number of points to plot for each orbit.
% F and G solutions
figure
hold
% Plot Earth
Dtheta_Earth=linspace(0,2*pi,nPoints); % linspace(X1, X2, N) generates a
                                       % row vector of N linearly equally
                                       % spaced points between X1 and X2.
rEarth_motion=zeros(nPoints,3); % Initialize rEarth_motion by assiging it
                                % a null matrix of nPointsx3
for iT=1:nPoints
    rEarth_motion(iT,:)=FGProp_trA( Dtheta_Earth(iT), rEarth, vEarth, muSun);
    % FGProp_TrA provides a function implementation of the pseudo-code
    % given in slide 24.
end
plot3(rEarth_motion(:,1),rEarth_motion(:,2),rEarth_motion(:,3))
```

5. Compute the $\Delta\theta$ of the transfer and propagate using a similar approach as the one above, but for the required $\Delta\theta$.

6. Propagate the minimum energy transfer by solving the following first order ordinary differential equations:

```
%Equation of motion ¨r+mu*r/R^3 = 0
F=@(t,x)    [x(4); %dx/dt=Vx
             x(5); %dy/dt=Vy
             x(6); %dz/dt=Vz
             -muSun*x(1)/(sqrt(x(1)^2+x(2)^2+x(3)^2)^3); %dVx/dt
             -muSun*x(2)/(sqrt(x(1)^2+x(2)^2+x(3)^2)^3); %dVx/dt
             -muSun*x(3)/(sqrt(x(1)^2+x(2)^2+x(3)^2)^3)]; %dVx/dt
```

Use one of Matlab ode solvers, for example: ode45.