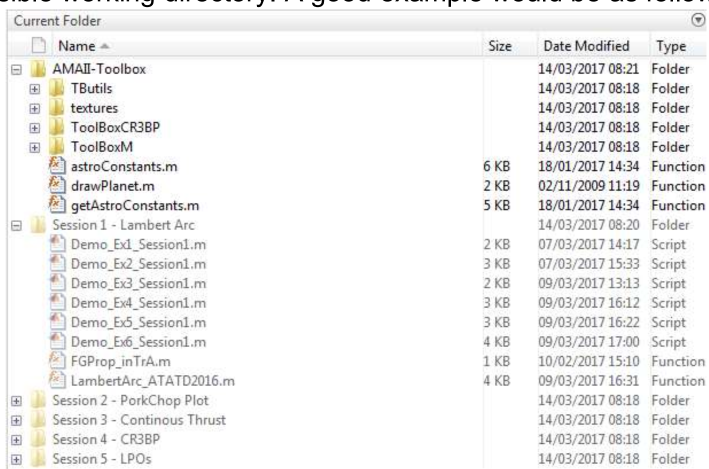
Basic Notes

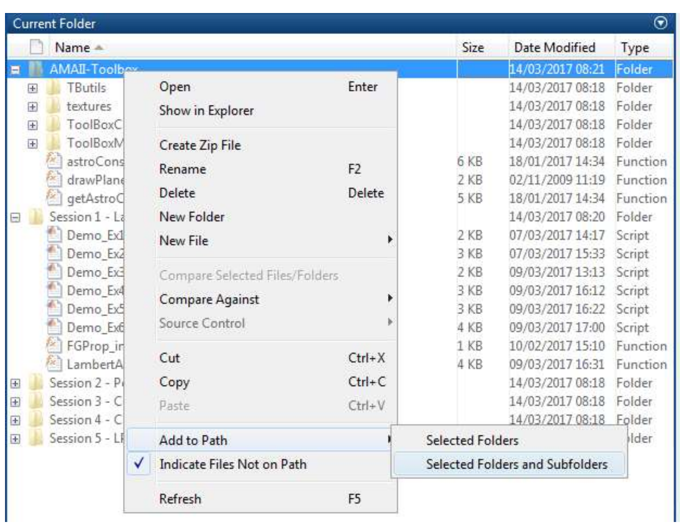
The following bullet points provide some basic hints on how completing the tasks of this course.

- 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.



- Remember the rule: Never ever duplicate code sections! When another choice can be made: Choose always to group into a common function any code that repeats with only small differences due to different input parameters.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

Session 1 Lambert Arc Matlab Guide PART B

Exercise 3. “Shoot” the minimum energy orbit for a Δt equivalent to the real ExoMars TGO transfer (Δt= 215 days) and compute the $d\mathbf{x}$ error of the final position state.

1. Implement algorithm in slide 48 to complete Exercise 3. Also use the code you have written to solve the minimum energy orbit to compute the departure velocity from Earth.

2. Propagate for the required Δt using F and G coefficients and compute the difference between Mars at the arrival date and your position.

Exercise 4. Use the STM to correct the previous shooting and come up with a better approximation of the velocity at departure.

1. Resume from script implementing Exercise 3, save it as Ex4_session1.m

2. Implement the state transition matrix algorithm as described in slide 55.

Help: See code for the implementation (note nomenclature may vary slightly from the slides)

```
% This section reproduces the pseudo-code in slide 35.
%-----------------------------------------------------------------------
% Relevant State Transition Matrix
C=sma*sqrt(sma^3/muSun)*(3*sin(DE_f)-(2+cos(DE_f))*DE_f)...
    -DT*sma*(1-cos(DE_f));

dr=rf-r1;
%r2dot
% Gdot
Gdot=1-sma/norm(rf)*(1-cos(DE_f));
r2dot=1/G*(-r1+Gdot*rf);

dv=r2dot-r1dot;

drf_drdotzero=r1_n/muSun*(1-F)*(dr'*r1dot - dv'*r1)+...
              C/muSun*r2dot'*r1dot+G*eye(3);
```

3. Compute matrix $\frac{\partial \mathbf{r}_f}{\partial \dot{\mathbf{r}}_0}$ using the initial conditions $\left(\mathbf{r}_0 \quad \dot{\mathbf{r}}_0\right)$ as given from the minimum energy orbit.

4. Compute the correction to apply to $\dot{\mathbf{r}}_0$ as:

$$\Delta\dot{\mathbf{r}}_{Correction}\left(t_1\right)=\left(\frac{\partial \mathbf{r}_f}{\partial \dot{\mathbf{r}}_0}\right)^{-1}\left(\mathbf{r}_{Mars}-\mathbf{r}_f\right)$$

Or in MATLAB code:

```
Dr1dot=inv(drf_drdotzero)*(rf-r2)';

r1dot_new=r1dot-Dr1dot';
```

5. Perform the correction to the initial velocity as $\dot{\mathbf{r}}_0 + \Delta\dot{\mathbf{r}}_{Correction}$ and propagate the conditions $\left(\mathbf{r}_0 \quad \dot{\mathbf{r}}_0 + \Delta\dot{\mathbf{r}}_{Correction}\right)$ for Δt= 215 days.

6. Is the final distance between $\mathbf{r}_f$ and the position of Mars on 15/10/2016 smaller than in Exercise 3?

Exercise 5. Iterate the differential correction method until convergence to an error of below 1m.

1. Resume from script implementing Exercise 4, save it as Ex5_session1.m

2. At the end of the first “shooting” (i.e. propagation), create a new variable name Error such as:

```
Error=norm(rf-r2)
```

3. Create a *while-loop* after the first shooting (i.e. previous to the first correction) on which the expression that limits/stops the execution of the loop is:

```
while Error>1e-3 % 1m...
```

4. Run the script and, if done correctly, should converge in four loops.