Space Mission Analysis

Assignment 2 – due 02.12.2025 11.59pm

The assignment shall be done in groups of 2-3 students. However, each student must upload the code and report individually via ISIS!

Reports must include the plots and some basic explanations in PDF format as well as your MATLAB code.

Alternatively, you can submit a MATLAB live script (.mlx) showing and commenting all plots and calculations instead of the PDF.

Initial Orbit Determination (40p.)

Table 1: Observations for task 1 & 3

Number	Time	r_x	r_y	r_z
1	0.0	3419.8556 km	6019.8260 km	2784.6002 km
2	1min 16.48s	2935.9119 km	6326.1832 km	2660.5958 km
3	2min 33.04s	2434.9520 km	6597.3867 km	2521.5231 km

Table 2: Observations for task 2

Number	r_x	r_y	r_z
1	0 km	0 km	6378.137 km
2	0 km	-4464.696 km	-5102.509 km
3	0 km	5740.323 km	3189.068 km

1. Gauss' solution of Lambert's problem.**(9p.)**
   a. Implement a function for the Gaussian solution for Lambert's problem. (5p.)

      Hint:
      $$x_2 = \frac{4}{3}\left(1 + \frac{6}{5}x_1 + \frac{6}{5}\frac{8}{7}x_1^2 + \frac{6}{5}\frac{8}{7}\frac{10}{9}x_1^3 \ldots\right)$$
      Can be coded as:

```
term = 1;
x2 = term;
nTerms = 4 %number of terms in the series
for i=2:nTerms
    j=2*i+1;
    term=term*x1*(j+1)/j;
    x2=x2+term;
end
```

   b. Using observations 1 and 3 in table 1, calculate $\boldsymbol{v}_1$ and $\boldsymbol{v}_3$ and determine the Keplerian elements for $\boldsymbol{r}_1$ and $\boldsymbol{v}_1$ and for $\boldsymbol{r}_3$ and $\boldsymbol{v}_3$. (2p.)
   c. Propagate the determined orbit using either your Kepler or your Runge-Kutta propagator (or the uploaded solver) and mark the full orbit as well as the connecting segment. Show $\boldsymbol{r}_1$ and $\boldsymbol{r}_3$ in the plots. Include Earth to scale (using a sphere is sufficient). (2p.)

2. Gibbs Method **(6p.)**
    a. Implement the Gibbs method as a function and compute $\boldsymbol{v}_2$ using the observations in table 2. (5p.)

        Test value using observations in table 2: $\boldsymbol{v}_2 = \begin{bmatrix} 0 \\ 5.531148 \\ -5.191806 \end{bmatrix}$ [km/s]

    b. Determine the Keplerian elements based on the observations in table 2, using r_2 and $\boldsymbol{v}_2$. (1p.)

3. Gibbs-Herrick Method **(6p.)**
    a. Implement the Gibbs-Herrick method as a function. Use a similar structure to your Gibbs method function. (5p.)

        Test value using observations in table 1: $\boldsymbol{v}_2 = \begin{bmatrix} -6.4416 \\ 3.7776 \\ -1.7205 \end{bmatrix}$ [km/s]

    b. Determine the Keplerian elements based on the observations in table 1, using r_2 and $\boldsymbol{v}_2$. (1p.)

4. Angles Only Gauss Approach **(13p.)**
    a. Implement the Angles Only Gauss approach with the expansion using the Gibbs and Herrick-Gibbs method. Determine the orbit based on observations 62,70 and 74 given in observation_data.mat. Right ascension and declination are given as topocentric coordinates. Keep in mind that all vectors must be in the same (ECI) system for calculations using the angles only method. Use existing MATLAB functions for coordination transformation. (8p.)
    Hint 1: Choose a tolerance level high enough and/or limit the loop by number of iterations. A reasonable starting point for tolerance levels for this task (!) is 1e-9 and max_iter 1000. It is however always advised to check whether the conditions can be eased to reduce computational burden.
    Hint 2: Keep in mind to use both Gibbs and Gibbs-Herrick depending on the data.
    Hint 3: Use the `roots` function to find the roots of the polynomial and be careful with the order the function treats the polynomials.
    b. Compute the Keplerian elements based on your results from 4a. (1p.)
    c. Evaluate the Kepler elements and their variations using different observation batches (e.g., 10 sets of 3 observations each), that are chosen randomly. Keep in mind the chronological order of observations in each set as well as the chronological order of the sets itself. (4p.)

Your report should include all graphs, captions and some basic (!) explanations of what you did. The report's form, structure, language etc. will be accounted for with **3p.** as well as the code's structure, appropriate comments etc. will be accounted for with **3p.**

I strongly recommend writing your script as lean as possible and to try to avoid repeating code. Use loops and parameter sets to calculate or plot different orbits.

All calculations and simulations must be done using MATLAB with own functions (no usage of libraries or proprietary or open-source functions)!

Exceptions to this are the functions *eci2lla, lla2ecef, eci2ecef* and their reverse versions. Furthermore, functions like *wrapTo360, wrapTo2Pi* and similar as well as *geoplot* are exceptions.