

Orbital Mechanics - Assignment report



**Orbital Mechanics
Master of Science in Space Engineering
Academic Year 2019-2020**

GROUP 1

**Assignment 1: Interplanetary Explorer
Assignment 2: Planetary Explorer Mission**

Name	Person Code	Matricola
Enrico Bassissi	10538061	945090
Lucia Apparenza	10533452	945706
Marco di Trocchio	10545177	945944
John Lane	10683919	928727

Assignment 1: Interplanetary Explorer Mission

1 Abstract

The purpose of the assignment was to design an interplanetary transfer of a spacecraft that minimised the ΔV required to get from Mercury, the starting planet, to Neptune, the target one, in a time window of 40 years. In order to accomplish this, a fly-by around Mars was required, which provided a gravity assist ΔV to the spacecraft. The group achieved the minimum ΔV by means of a triple loop process: the first loop imposed time windows for each stage (departure, fly-by, arrival), based on reasonable analysis of the relative motion of the planets. The triple loop optimisation computation was then repeated two further times using a narrowed time span centred on the result of the previous routine, in order to calculate a more accurate value in computationally efficient manner. As a different approach, a genetic algorithm was then used to provide a solution which could be compared with the one from the triple-loop process.

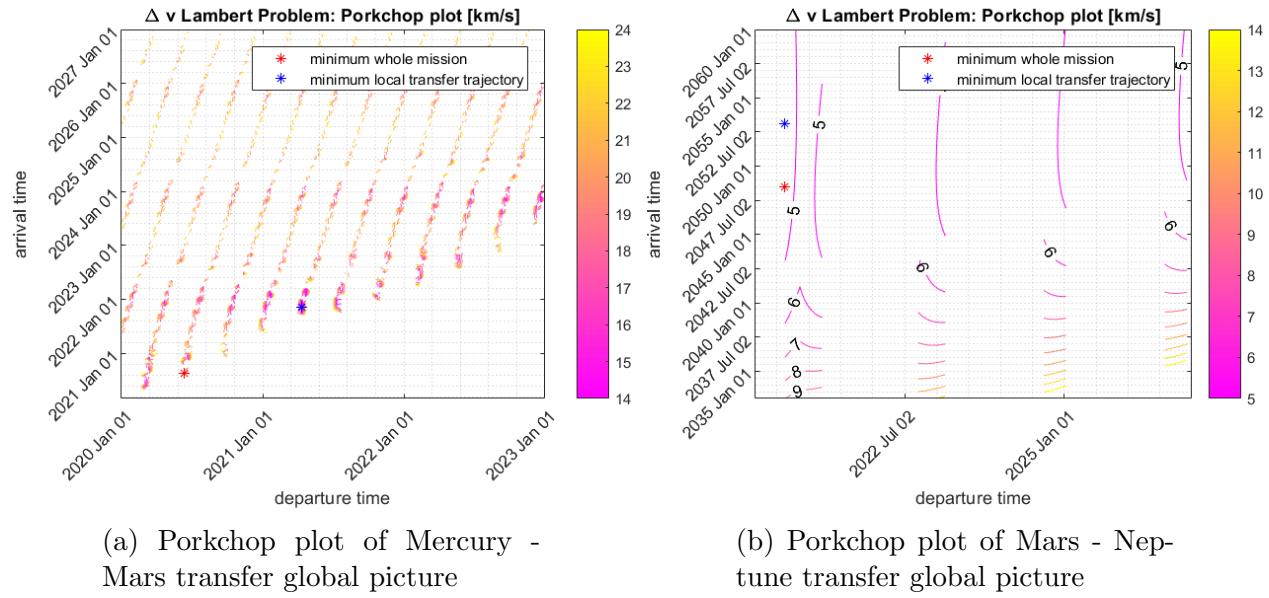
2 Design Process

2.1 Time window selection

The time windows were chosen based on both the synodic periods and the relative orbit motions of the planets, combined with a trial and error process of porkchop plot interpretation to find a time when the position of the planets was such that energy requirements were lowest.

2.1.1 Departure from Mercury

The synodic period was a useful tool in selecting which time spans to allocate for each section of the journey. This is because the synodic period covers the window of all possible relative alignments of one planet with respect to each other, estimating the orbits as circular this was calculated as 100.88 days for Mercury-Mars. This set the lower bound of the time span between earliest and latest departure. But as the orbits are not circular, the trajectories from one synodic period to another may vary. For this reason, also the porkchop plot was used to adjust the window. Starting from a generic Lambert problem between Mercury and Mars, the graph shows that the pattern is actually repeated after one synodic period and in each one a local minimum can be found. Even if it does not represent the optimal value of the whole minimizing process (compare the blue marker in fig. 1), with (the red marker in 1a).



(a) Porkchop plot of Mercury - Mars transfer global picture

(b) Porkchop plot of Mars - Neptune transfer global picture

2.1.2 Flyby window

For the choice of the window of the flyby manoeuvre, the minimum transfer parameters between Mercury and Mars were calculated: in terms of time, the minimum was obtained using the code *parabolic_transfer.m*, which computes the parabolic transfer time starting from Mercury until reaching Mars.

Since the orbit of Mars is not perfectly circular, a method to calculate the range of parabolic times of flight was composed. The maximum was set as the time from Mercury to the circle of radius equal to the apoapsis of Mars, the minimum was the time to the circle of radius equal to the periapsis of Mars. This provided a range between 56.7 days and 72.87 days for the minimum parabolic transfer time, dependent on the departure date found by the Lambert solver. In order to integrate this result and to verify it, another function: *window_check.m*, was used. This takes into account the upper bound constraint over the ΔV , by using the Lambert solver which returns false if the value is above a limiting ΔV , which is based of current technological limitations. In line with the previous results, the minimum transfer time would be 58.09 days, thus the earliest date for Mars flyby was set two months later the first available launch date. In terms of ΔV cost, the minimum can be estimated through a Hohmann transfer between the two heliocentric orbits considered circular. The function *homann_simil_flyby_cases.m*, which went through all the possible combinations, was used in a post-process phase in order to verify the consistency of the results.

The process previously described was then repeated for the Mars-Neptune transfer, again imposing a limiting ΔV for the departure from Mars inside the *window_check.m* function, so that the minimum time of transfer was found to be around 6 years. This gave a minimum date of arrival of 2033/1/1, and the rest of the time window was allocated towards Neptune.

2.2 Time window discretisation

The next step was to discretise the chosen time windows. The process of discretisation was a trade off between choosing enough points such that all the various meta-geometries of the orbit could be selected and trialled, while at the same time selecting few enough points such that the calculation could be carried out in a reasonable time with the computational power available. The level of discretisation was based on the orbital period of each of the planet. Mercury had the shortest orbit so it was necessary to discretise into small time periods between points. Mars has a longer orbit so the time period could be extended, but at the same time we must consider that it is the flyby planet. Therefore, its change of position sensibly affects the geometry of the hyperbola that will be needed to connect the two interplanetary legs; for these reasons the middle window requires more evaluation points. The number of points on Neptune, after a trial and error process, was chosen to be kept low in order to save computational cost

	Global Picture		Subroutine 1		Subroutine 2	
	Time (d)	Angle (deg)	Time (d)	Angle (deg)	Time(d)	Angle (deg)
Mercury	10.15	41.13	1.29	5.37	.16	.46
Mars	14.11	7.44	1.27	1.04	.16	.14
Neptune	80.83	.49	20.54	.12	5.09	.03

Table 1: Discretisation of time steps in time (days) and angle (degrees) for each subroutine

3 Calculation of Transfer Arcs

3.1 Lambert Solver

The next step was to use the Lambert solver function to compute and analyse all the possible transfer arcs for each of the time windows. Within this evaluation, any of the values of ΔV 's that were outside of physically acceptable or feasible bounds were discarded in order to make the code more efficient. The functions *lambert_solver_flyby_"custom".m* returned the ΔV associated to each trajectory for each time in the Mercury arc combined with each time in the Mars arc, then the result was plotted in a porkchop plot. The same process was repeated for the Mars to Neptune transfer arc and this provided an incoming and outgoing Lambert arc for each instant of time for the global window.

Algorithm 1: Triple Loop Lambert Solver

Input: Time Windows and Discretization Point
Output: Minimum ΔV for the Mission

```

for Each Mercury Departure Date do
    for Each Mars Flyby Date do
        if Mercury Departure Time > Mars Arrival Time then return
        else Calculate  $\Delta V$  for the First Transfer Arc
        if Technologically not Achievable then return
        else
            for Each Neptune Arrival Date do
                if Mars Departure Time > Neptune Arrival Time then return
                else Calculate  $\Delta V$  for the Second Transfer Arc
                if Technologically not Achievable then return
                else
                    Do the Flyby Manoeuvre on Mars
                    if Radius at Periapsis or Turn Angle not Valid then
                        return
                    else Store Value in Matrix
    
```

Find the minimum in this 3D matrix and its corresponding Time of Flight

3.2 Powered gravity assist calculation

The powered gravity assist was calculated using the *PGA.m* function. The role of this function was to find the artificial Δv needed to propel the spacecraft to the velocity required by the outgoing heliocentric Lambert leg to perform the interplanetary trajectory required by that iteration of the loop. It does so by separating the contribution of natural and artificial velocity increase; this verified, as expected, that the acceleration due to Mars is very little (less than 1 km/s), owing to the dimensions of the planet itself and to its distance from the target planet. Now, the amount that came from the power system on the spacecraft had to be included in the final ΔV cost calculated. In addition, this function sets the constraints that the hyperbolic path must respect in order to avoid crashing onto the surface of planet or even pass inside its atmosphere, which would affect spacecraft trajectory. Mars atmosphere taken to extend to 200 km over its surface[8][9], and this set the radius of perigee limit of hyperbola (and therefore the aiming radius), from which the flyby path could be fully characterized in order to have the desired outgoing velocity. All of these calculations were made following the patched conic method, which consists of linking the spacecraft parameters from the outside by zooming inside the sphere of influence of the planet.

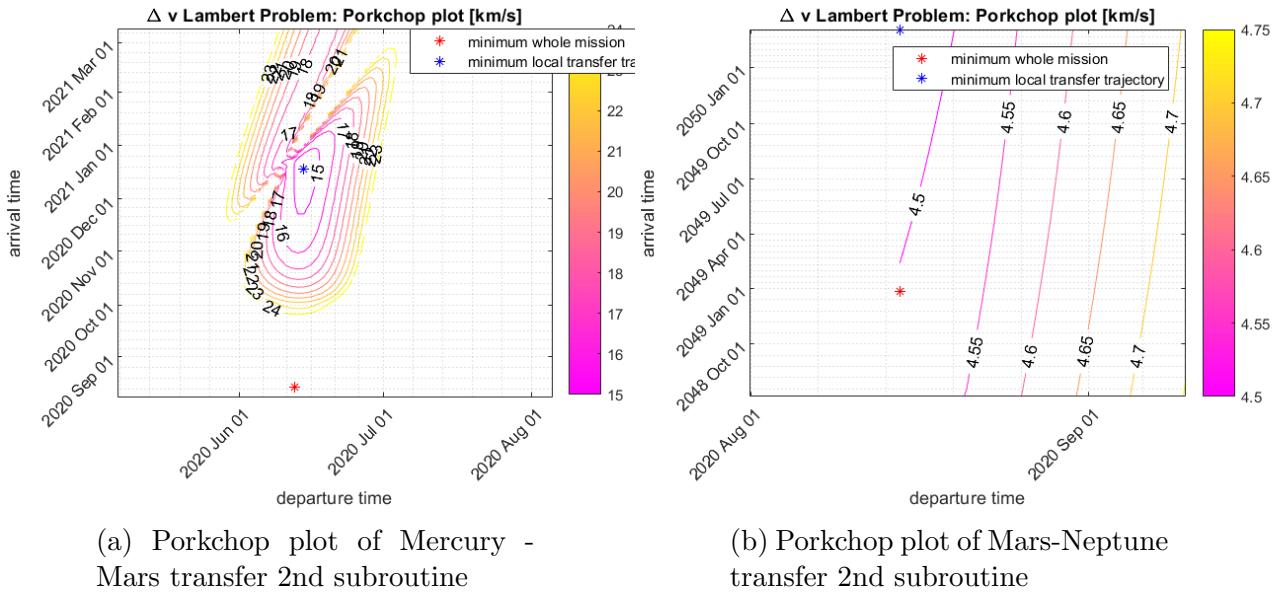
3.3 Calculation of the minimum value

The ΔV values were calculated for each possible combination and were stored in a 3-dimensional matrix where each dimension represented the time windows of departure, flyby, arrival. Each i,j,k element in the ΔV matrix corresponded to a value in the matrix of position and velocity vectors of the planets. The minimum value in the ΔV matrix was found, and the corresponding radial and temporal coordinates were then traced back in the position and

velocity matrices at which this minimum value occurred.

4 Matrix Refinement

After finding the mission trajectory at which the ΔV was minimised, it was hypothesized that this value gave the approximate time at which the ΔV was at an absolute minimum, but that further refinement of the time windows would lead to a more accurate answer. This is because the previous discretisation allowed for large time gaps between events occurring, so it provided just a global overview of the behaviour of the ΔV . Therefore, the next step was to use our function *ref_param.m* which sets the time obtained from the first loop as a midpoint for the new time window of the first subroutine, divided into units that were a certain number of times smaller: now there will be a matrix of the same size as the previous one but over a smaller time span, providing evaluation points closer to each other. The refinement parameter was selected as 8 for Mercury, meaning the time between each iteration was one eighth of the previous routine, 6 for Mars and 4 for Neptune. Neptune has a much longer orbit compared to Mars and Mercury so choosing too small a time gap meant its position would hardly change in relation to Mars and Mercury. Again from this process a minimum ΔV was obtained, from which the same refining routine was repeated, reducing the windows. The graph below shows the porkchop plot from the second subroutine:



5 Triple Loop Results

5.1 Time windows

The results of triple-loop are shown below showing progressively narrowed time windows for Mercury, Mars and Neptune.

		Earliest departure	Latest departure
Mercury	Global picture	2020/1/1 00:00:00	2023/1/1 00:00:00
	Subroutine 1	2020/4/3 20:53:20	2020/8/17 14:43:25
	Subroutine 2	2020/6/4 3:26:21	2020/6/20 22:55:57
Mars	Global picture	2020/3/1 00:00:00	2027/1/1 00:00:00
	Subroutine 1	2020/4/27 10:13:05	2021/3/11 00:48:10
	Subroutine 2	2020/7/31 21:23:28	2020/9/9 21:58:39
Neptune	Global picture	2033/1/1 00:00:00	2060/1/1 00:00:00
	Subroutine 1	2045/2/12 7:49:10	2051/10/24 12:47:43
	Subroutine 2	2048/7/6 14:09:39	2050/3/4 22:7:35

Table 2: Time windows for Mercury, Mars and Neptune during triple-loop

5.2 Optimal dates

The result of the triple loop process found an optimum time for each launch, flyby and arrival. The results each subroutine are displayed in the table below.

	Mercury	Mars	Neptune
Global picture	2020/6/11 08:53:20	2020/8/17 6:54:54	2048/6/28 22:49:10
Subroutine 1	2020/6/12 15:03:14	2020/8/14 17:53:57	2049/5/8 22:49:10
Subroutine 2	2020/6/13 12:57:05	2020/8/14 17:53:57	2048/12/26 19:00:48

Table 3: Optimum arrival time from each subroutine

5.3 Minimum ΔV

	Global picture	Subroutine 1	Subroutine 2
Minimum ΔV [km/s]	29.9915	27.9332	27.6672

Table 4: Minimum ΔV for each triple-loop iteration

	Global picture	Subroutine 1	Subroutine 2
ΔV Mercury-Mars [km/s]	22.67	23.28	23.13
ΔV Mars-Neptune [km/s]	4.57	4.524	4.535

Table 5: Optimum arrival time from each subroutine

5.4 Gravity assist data

	PGA ΔV [km/s]	Natural ΔV [km/s]	$r_{periapsis}$ [km]	Delta [deg]	SOI time [h]
Global picture	2.755	0.156888	3723.24	1.67514	10.8556
Subroutine 1	0.130276	0.6741	3717.37	1.59093	10.5788
Subroutine 2	0.0014477	0.785178	3721.4	1.56518	10.498

Table 6: Data of gravity assist manoeuvre

5.5 Final trajectories

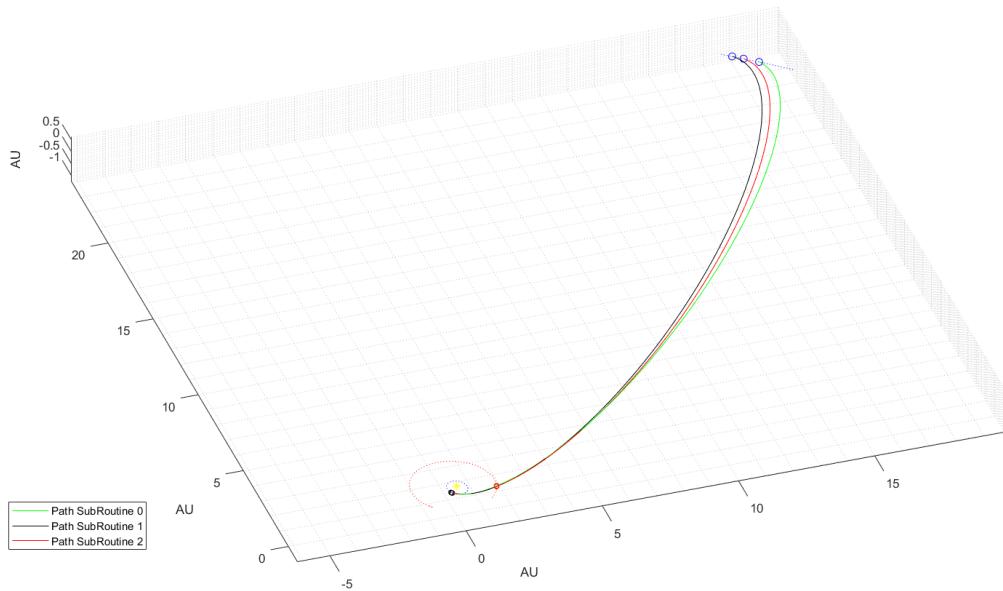


Figure 3: Final trajectory for each iteration of triple loop

6 Genetic Algorithm Design Process

The genetic algorithm was again initiated by imposing time windows for launch, fly-by and arrival of the interplanetary transfer. These values set the lower and upper boundary from which the function takes the random times used to calculate the ΔV and minimize it.

The physical constraint was also set, in the form of linear inequality constraints to specify the restriction that it must respect when choosing the evaluating time values of each transfer leg. This eliminated the possibility of an arrival time before a departure time:

$$T_{mars} - T_{mercury} > 2 \text{ months} \quad T_{neptune} - T_{mars} > 6 \text{ years} \quad (1)$$

The ΔV 's are calculated using the *ff-ME2NE.m* fitness function which again calls on the *uplanet.m* function and the Lambert solver to work out the total ΔV for the randomly selected times.

The other important aspect of initializing the genetic algorithm was choosing the right options. The results can be affected by choosing different operators therefore, choosing a good set of operators for the problem is often done better by experimentation.

Through a series of iterations changing *Population Size*, *Max Generations* and *Crossover-Fraction* with the function *ga_loop_options.m* it appeared that the only significant constraints in choosing them were represented by a low disparity between the first two and a value of *CrossoverFraction* = 0.8. The Crossover fraction is a way to stochastically generate new solutions from an existing population.

Even the literature on this topic often disagrees among sources; in one paper by R. Storn [7] for the P.S., it is argued that *PopulationSize* = $10 \times D$, where D stands for the dimension of the problem. In this case $D = 3$, so it was decided to increase the number *P.S.* = 30, remaining in the same order of magnitude: i.e *PopulationSize* = 80, and set a maximum number of generations still of the same order of magnitude, but bigger in order to give the code more possibilities to converge properly with the given tolerances, i.e *MaxGenerations* = 150.

6.1 Method 1

As a first approach, the group decided to run the algorithm over the whole windows, with less attention to details, in order to avoid making the code computationally heavy as done in the triple loop part and without any human deterministic control.

In this case the result of the genetic algorithm was interesting, since it provided a value that was quite different from the one obtained by the triple loop. The higher value may be due to the fact that it was run over a larger window and therefore has less chance of locating the minimum.

A hybrid function begins from the point where the genetic algorithm stops. To further improve the solution (or when the genetic algorithm reaches the *MaxGenerations*) the group used the *fmincon.m* optimization solver of Matlab to perform a faster and more efficient local search.

6.2 Method 2

Being non-deterministic, each time the genetic algorithm was launched, it yielded a different value. Even running the code several times, the result was always greater in value with respect to the one of the triple loop. To improve that, a sort of triple loop process was again used, but with lower computational time and load required. Using the same options as before, the previous time windows were divided in smaller ones, in order to let the genetic algorithm focus on smaller time spans and thus find the best solution without running the risk of getting stuck in a local minimum. To set these time spans the synodic periods of the planets were exploited as in the triple-loop: the lower and upper bounds set the limit date of each synodic period inside the previous global time windows, progressively growing. For Neptune instead, the initial time window was arbitrarily divided into 3 sub-windows in order to let the analysis be more adaptive for the last transfer. Doing so, 15 spans resulted in the Mercury window, 5 spans in the Mars one and 3 spans in Neptune.

7 GA Results

The two methods applied for the genetic algorithm yielded very different results:

	$\Delta v_{tot} [km/s]$	$\Delta v_{pga} [km/s]$	$\Delta v_{nat} [km/s]$	$r_{peri} [km]$	$\delta [\text{deg}]$	SOI time [h]
Method 1	32.1645	6.72239	0.159653	3672.43	1.82429	11.1837
Method 2	28.3537	4.26924	0.142488	3687.34	1.71069	10.8961

Table 7: Results of the Genetic Algorithm

	Departure Mercury	PGA Mars	Arrival Neptune
Method 1	2021/3/6 8:28:24	2024/12/29 22:25:50	2048/4/30 11:42:25
Method 2	2020/6/15 9:18:6	2026/11/26 12:23:45	2048/4/19 13:50:36

Table 8: Optimum arrival time from GA

The first plot of 4 shows the best and mean values of the population in every generation and where the plot identifies the best value found by *ga* when it stops. The second plot shows the solution improvement given by the gradient and the point of evaluation that represent the *mjd2000* of departure, flyby and arrival at that minimum found: it can be seen that using a hybrid function improves the accuracy of the solution.

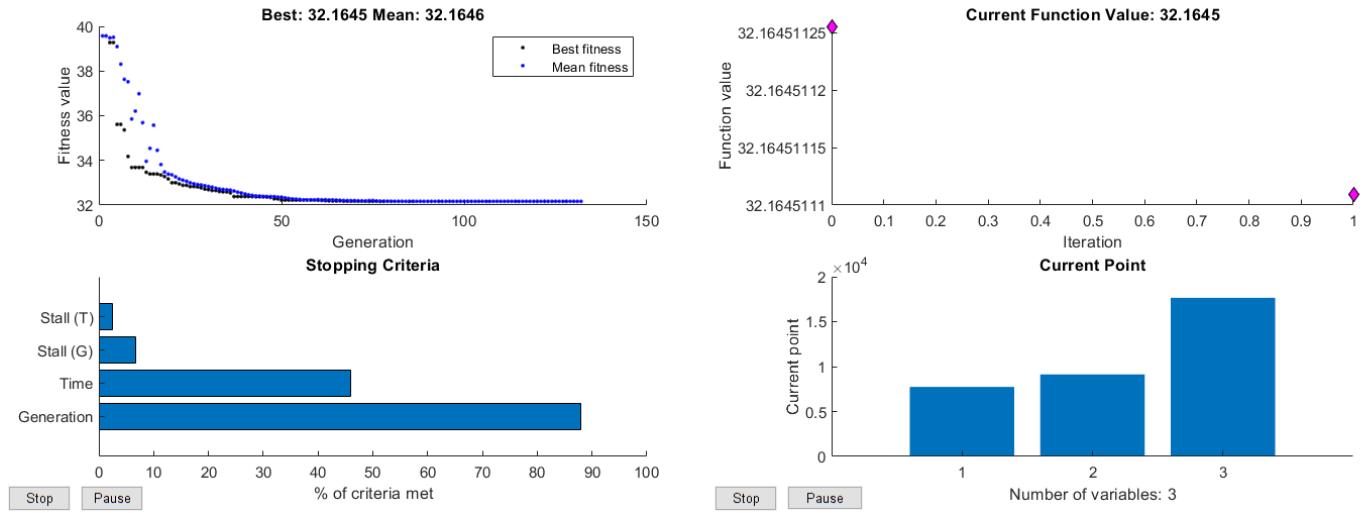
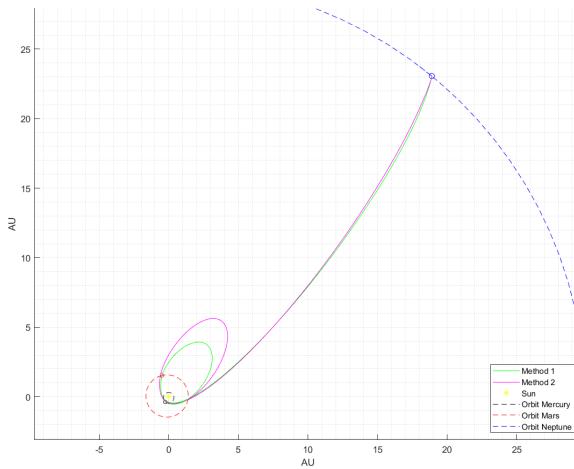
Figure 4: Convergence of GA and $fmincon$, Method 1

Figure 5: Trajectory achieved through Genetic Algorithm Method 1 and 2

8 Conclusion

The ΔV obtained from the grid search method produced the lowest value which was chosen as the optimum. Even after running the genetic algorithm multiple times it did not produce a ΔV as low as the triple-loop. This was because the grid method takes into account the synodic periods of the planets and confines launching to occur only in times when the alignment of the planets favours minimising the ΔV . Within these time windows, where the minimum was likely to occur, all the possible combinations were then tested, and then further analysed

at a more discrete level, meaning it was highly likely that a minimum would be found as the time windows were progressively narrowed.

The genetic method values were chosen at random over the entire time windows. Therefore it did not search the time windows as meticulously within the favourable regions as the triple-loop. It was interesting to see that when both methods were combined, i.e. the group imposed time windows based on synodic periods and then ran the genetic algorithm within these windows the results came very close to the optimum ΔV from the triple loop.

It was interesting to observe that there is little change in the ΔV in the trip loop from the second to third iteration owing to the fact the spacecraft had already found the optimum Mercury-Mars transit and this was the part of the mission that consumed most of the energy (approximately 80%). Mercury is closer to the sun, therefore the gravitational pull is much stronger in the Mercury-Mars transfer which requires more fuel to overcome.

Further improvements to the final value ΔV could have been made by incorporating multiple fly-by's such as the Sun and Jupiter to exploit the natural gravity boost, but this mission was confined to Mars which at the end did not provide a large natural ΔV boost. Increasing the time windows above the given bounds until the planets were aligned more favourably could also have allowed for a reduction in ΔV .

This problem provided an interesting comparison between non-deterministic methods of interplanetary transfer optimisation, such as genetic algorithms, and standard grid search refinement. It is widely accepted in the space community that heuristic methods are the best choice for optimisation of interplanetary transfers [2]. The difference is that most space missions involve many design variables i.e. maximum payload, minimum ΔV , minimum time, thus requiring multi-objective optimisation. In our problem the only objective was to minimise the ΔV , therefore a more theory based approach is favoured.

Assignment 2: Planetary Explorer Mission

9 Abstract

The second assignment involved the analysis of the effect of orbital perturbations on the motion of an Earth orbiting satellite: the team was requested to consider only the effect of Lunar and J2 zonal harmonics. The perturbations on each Keplerian parameter was plotted in both short and long time spans. The frequency analysis was then carried out in order to find the main harmonics associated with the perturbing effects. Furthermore, the motion of the spacecraft was compared with real data of a satellite chosen by the group among the ones with similar characteristics. Chinasat 7 evolution was shown to exhibit a similar behaviour in some but not all orbital parameters, with deviations due to the omission of other perturbations.

10 Nominal Orbit Propagation

The mission starts with the satellite in its nominal orbit defined in **ECI**¹ frame by the Keplerian parameters shown below. The parameters, a , e , and i were provided, while Ω , ω and θ were selected by the group with values chosen to ease the process of analysis with Chinasat 7 later on.

	a [km]	e [-]	i [deg]	Ω [deg]	ω [deg]	θ [deg]
Nominal Orbit	38329	0.1398	24.4561	35.7116	258.9600	0

Table 9: Initial Orbital Parameters

The Keplerian parameters were first converted to Cartesian parameters using the *sv_from_coe* function and then the *ode113*² numerical integrator to propagate the orbit data over time and show the unperturbed satellite path.

11 Ground Track

The groundtrack is the projection of a satellite's orbit over the Earth's surface. At each time the corresponding ground track point is defined by its latitude and longitude relative to the rotating Earth and can be plotted as the trace left on the planet's surface by the line connecting the centre of the Earth and the satellite as it travels its orbit. Since the satellite

¹ECI: Earth Centered Inertial frame

²*ode113*: Non-Stiff Ordinary Differential Equation solver derived from Forward Euler Method

and Earth have different velocities, the groundtrack will be subjected to a westward shift in term of longitude for each period, given by the formula: $\Delta\lambda = T\omega_E$.

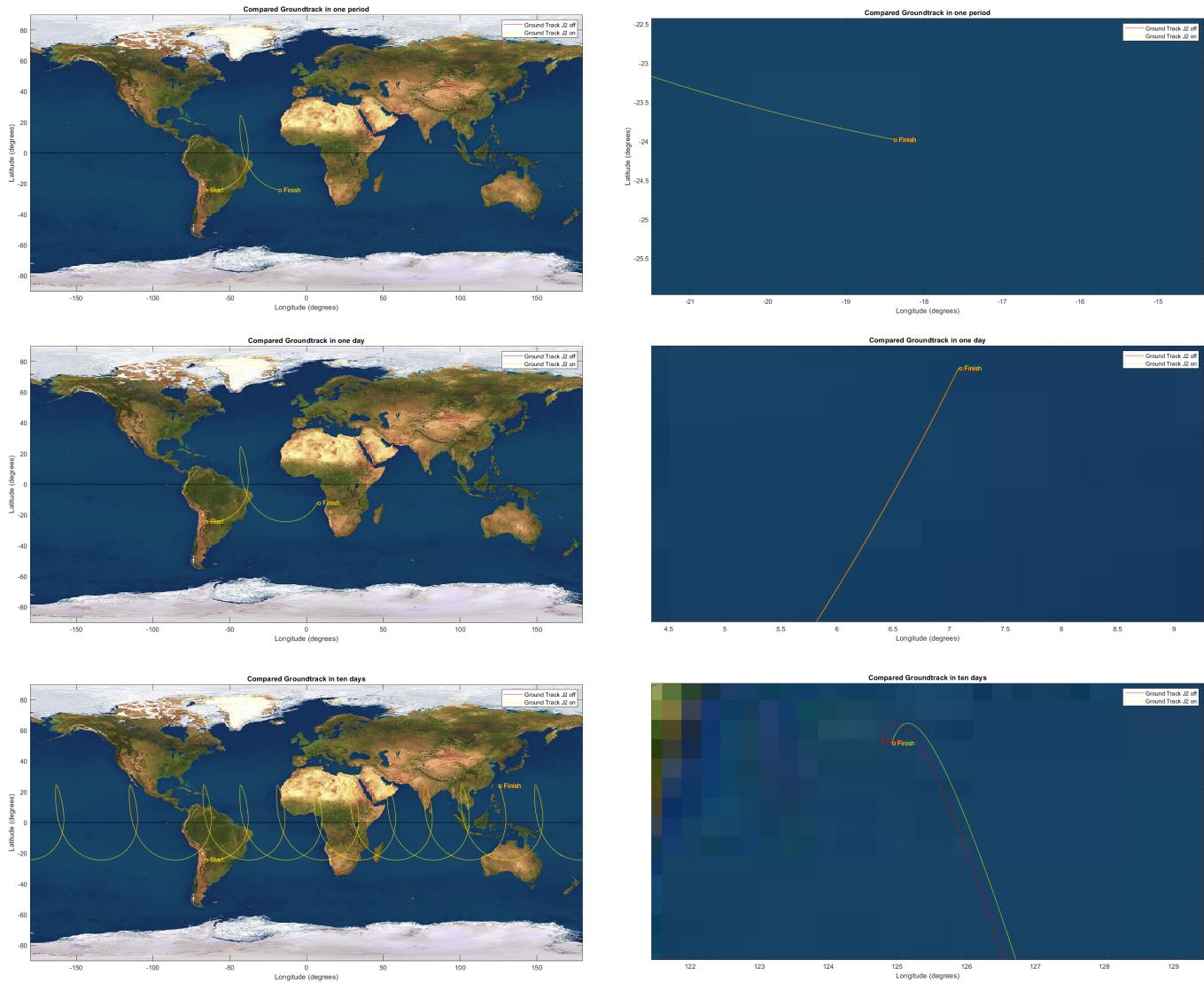


Figure 6: Ground Track for 1 Orbit, 1 Day, 10 Days

In this case the $k : m$ ratio was given as **6:5**, meaning that after six revolutions of the Earth, the satellite would undergo five full revolutions and begin at the same starting point again, given by the equation:

$$k\Delta\lambda = m2\pi \rightarrow \frac{T}{T_E} = \frac{m}{k} \quad (2)$$

From (2), the period of the new orbit was found and using formula (3), also the new value of the semi-major axis can be computed:

$$T = \frac{2\pi m}{\omega_E k} = 2\pi \sqrt{\frac{a^3}{\mu_E}} \quad (3)$$

Therefore, repeating ground tracks can be obtained by choosing an orbit with a certain suitable period, since T only depends on the semi-major axis a , it was required to modify a . It can be noticed that the groundtrack with modified a , in the unperturbed orbit, repeats itself exactly as there is no effect altering the orbit from one period to another (red line figure 7b). In the case of perturbed motion due to the J_2 effect, the perturbing acceleration modifies the trajectory so that each orbit groundtrack is different from the previous one, by a shift of longitude at each revolution due to $\dot{\Omega}(a, e, i)$, $\dot{\omega}(a, e, i)$, $M_0(a, e, i)$, $n(a)$. In this case the drift over one day has been calculated as $\Delta\lambda = 0.0114^\circ$.

The semi-major axis of the J_2 perturbed orbit was then modified (table 10) in order to achieve a repeating orbit, by solving the implicit function of the secular regression of the ascending node.

$$\frac{m}{k} = \frac{\omega_E - \dot{\Omega}}{n + \dot{\omega} + M_0} \quad (4)$$

The resulting groundtrack is the yellow line in figure 7a.

	Nominal	Modified a (J_2 off)	Modified a (J_2 on)
a (km)	38329	37338.43	37338.02

Table 10: Semi-major axes values for nominal and modified orbits

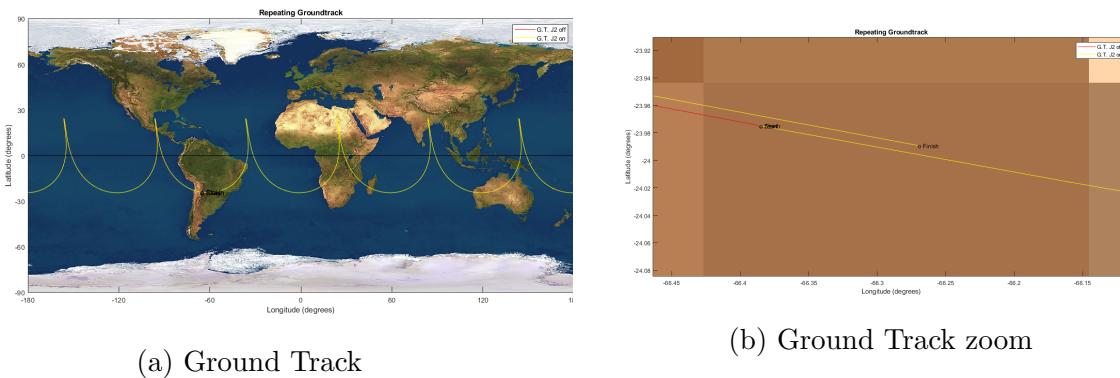


Figure 7: Repeating GroundTrack for unperturbed and perturbed case

Zooming close to the junction point of the groundtracks, even with the new semi-major axis value, the initial and final points are not overlapping. The reason is due to the fact that the J_2 model implies the solving of an implicit function and returns a semi-major axis value

strictly related to one precise situation: as time passes by, the error accumulates and the modified a is no longer suitable to attain a repeating ground track. Obtaining a repeating ground track requires modification in an iterative way.

The difference observed was a demonstration of the fact that, if not accounted for, the perturbations cause the real orbit to become different from the calculated one, ultimately making the information about its theoretical position useless after a long time.

12 Orbit Propagation

12.1 Propagation Methods and Interpretation of the Results

The perturbations here taken into account are the J2 (second zonal harmonics), related to the oblateness of Earth and Lunar third body effect which is due to the mutual gravitational attraction/interaction among the masses of the satellite, Moon and Earth. These effect can be considered as a summation of different perturbing accelerations added to the main gravitational acceleration:

$$\vec{r} = -\mu \frac{\vec{r}}{r^3} + \sum_{p=1}^N \vec{a}_p \quad (5)$$

where \vec{r} and \vec{r} are respectively the position and acceleration related to the satellite, \vec{a}_p are the N accelerations due to the perturbing effect, in the case illustrated N=2 (\vec{a}_m , \vec{a}_{J_2}).

The orbit was propagated over time through *ode113* solvers in two different ways:

- Cartesian coordinates using perturbing acceleration:

$$\begin{cases} \vec{a}_m = \mu_m \left(\frac{\vec{r}}{r^3} - \frac{\vec{R}_m}{R_m^3} \right) \\ \vec{a}_{J_2} = \frac{3}{2} \frac{J_2 \mu R_e^2}{r^4} \left[\frac{x}{r} \left(5 \frac{z^2}{r^2} - 1 \right) \vec{i} + \frac{y}{r} \left(5 \frac{z^2}{r^2} - 1 \right) \vec{j} + \frac{z}{r} \left(5 \frac{z^2}{r^2} - 3 \right) \vec{k} \right] \end{cases} \quad (6)$$

$$\{\ddot{r}, \dot{r}\} \implies \{a_{(t)}, e_{(t)}, i_{(t)}, \Omega_{(t)}, \omega_{(t)}, \theta_{(t)}\} \quad (7)$$

- Keplerian parameters in Gauss planetary equations:

$$\vec{a}_p^{rsrw} = [\hat{r} | \hat{s} | \hat{w}]^T \vec{a}_p^{XYZ} \quad (8)$$

$$\begin{cases} \dot{a}(t) = \sum_{p=1}^N \left(2 \sqrt{\frac{a^3}{\mu_E(1-e^2)}} e \sin \theta a_p \hat{r} + (1 + e \cos \theta) a_p \hat{s} \right) \\ \vdots \\ \dot{\theta}(t) = (...) \end{cases} \quad (9)$$

The lowest value of J2 acceleration was in the order of 10^{-9} (see figure 9 position [2,3] for the norm value), so it had to be assured that the tolerance was below this value. The chosen tolerances assigned to the ode solver were 10^{-13} for **Relative Tolerance** and 10^{-14} for **Absolute Tolerance**. The result obtained (figure 8) showed that there was infinitesimal difference in terms of accuracy between the two methods, which could be attributed to the accumulation of numerical errors over time (figure 8 position [2,3]).

Period of Observation	a [km]	e [-]	i [deg]	Ω [deg]	ω [deg]
30 days	4.3×10^{-8}	4.5×10^{-13}	6.8×10^{-14}	2.3×10^{-13}	9×10^{-12}
One year	1.5×10^{-6}	2.4×10^{-11}	1.9×10^{-12}	1.2×10^{-11}	1.1×10^{-10}

Table 11: Absolute Differences between Cartesian and Gauss Methods

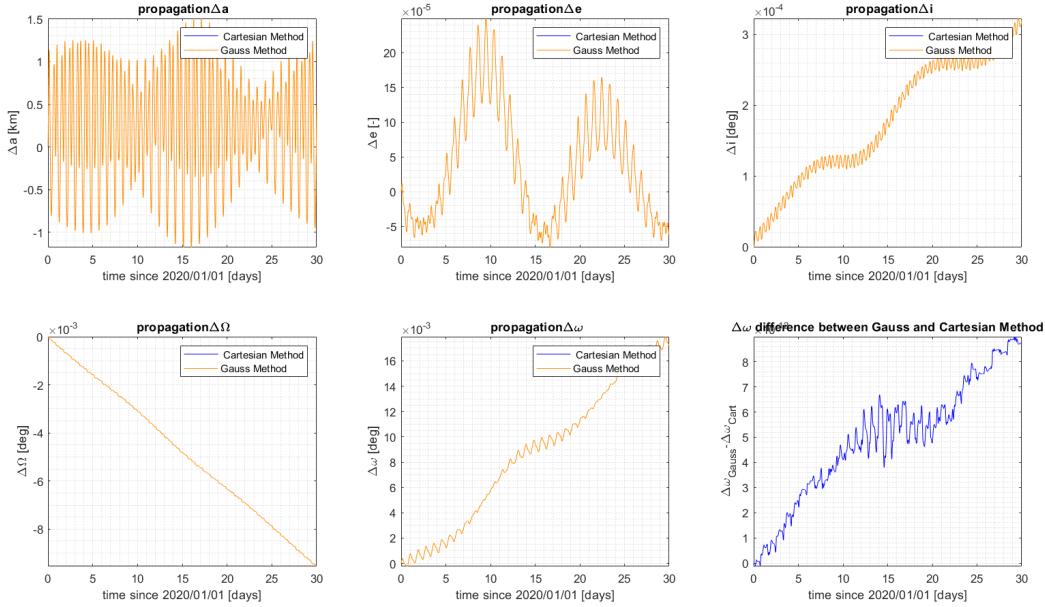


Figure 8: Deviation from Nominal Values of Gauss and Cartesian coordinates over 30 days

12.2 Time analysis of orbital parameters

It was chosen to observe the orbital parameters for time periods of 30 days and ten years and then compare them with respect to the ones of the nominal orbit to demonstrate the effects of the perturbations. The first choice came from the fact that the orbital period of the moon is about 27.3 days so the entire evolution of its effect can be observed over one month. On the other hand, ten years represents the typical lifetime of an operative satellite, therefore it represents the range of interest for most applications. The main perturbations observed were:

- **Short Term** oscillation for each orbital element that repeats approximately every orbital period
- **Long Term** oscillation that repeats in a very long period much higher than satellite's one
- **Secular Variation** which are non-periodic variation of orbital parameter

Short term variations have little influence on orbital parameters variations, since they repeat themselves with a very high frequency with respect to the characteristic period, thus the satellite orbit evolution follows the trend given by long and secular effects for this reason the evolution of θ is of no interest.

In figure 9, the effects of the single perturbations taken into account separately upon each orbital element is shown and then reported in Table 12 in terms of periodic and secular effects.

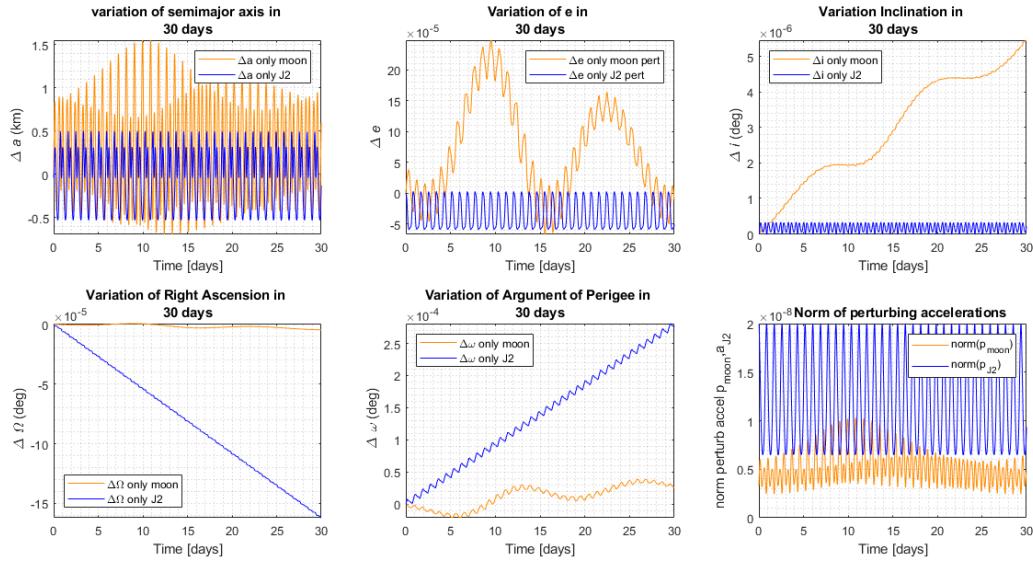


Figure 9: Evolution of orbital parameters (30 days)

The main long term variations of orbital parameters were due to the Lunar cycle and repeated themselves every semi-lunar period (approximately 14 days). The value for the perturbing acceleration also exhibits a maximum and a minimum when the Moon comes closest and farthest to the satellite during the semi-cycle.

It can be noticed from the evolution graphs (figure 9, position [1,1]), that the semi-major axis mean value remains almost constant, as expected, because both J_2 and the Moon are conservative perturbations, so the energy of the satellite's orbit does not change.

Keplerian Parameter	J2 effect	Moon's Third Body effect
Semimajor axis (a)	Periodic	Periodic
Eccentricity (e)	Periodic	Periodic and Secular*
Inclination (i)	Periodic	Periodic and Secular*
RAAN (Ω)	Periodic and Secular	Periodic and Secular*
Argument of Perigee (ω)	Periodic and Secular	Periodic and Secular*

Table 12: Effect of Moon and J_2 on orbital parameters

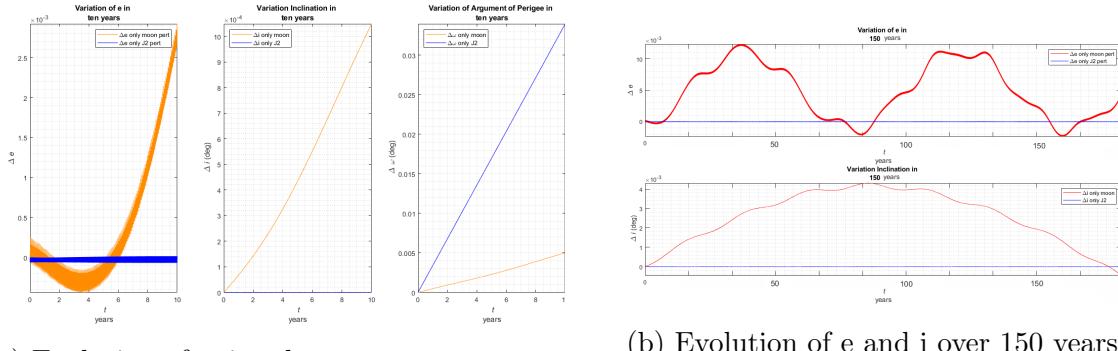
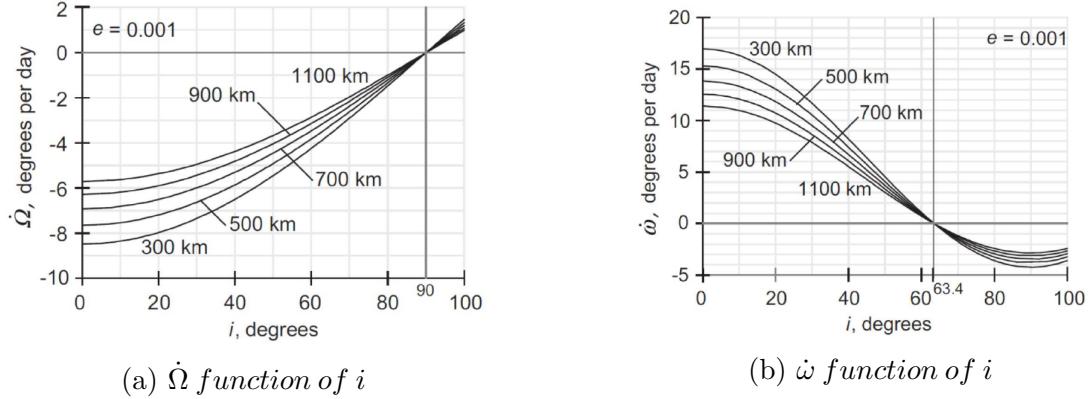
(a) Evolution of e , i and ω over ten years(b) Evolution of e and i over 150 years

Figure 10: Long Term effects of Moon and J2 on orbital parameters

In figure 10a it is shown that there is a long periodic variation in e repeating in a time span longer than ten years, while i and ω appear secular. In figure 10b a more accurate study of the long term evolution of orbital parameter over 150 years is displayed. This long term change is due to the precession of Moon's inclination with respect to the ecliptic plane, which moves in a retrograde direction with a period of 18.6 years.

The right ascension of the ascending node decreases accordingly to what was expected because the orbit has an inclination of less than 90° . This implies that it is prograde and therefore the nodal regression is negative (figure 11a). The anomaly of the perigee is increasing as expected because the inclination of the satellite's orbit is less than 63.4° , and therefore $\dot{\omega}$ is positive (figure 11b).

Figure 11: $\dot{\Omega}$ and $\dot{\omega}$ function of inclination at varying altitude and fixed eccentricity

12.3 Computational Cost Comparison Between the Methods

The two methods were indistinguishable from each other in terms of results, so a further analysis about their computational load and time had been carried out.

The reference parameter chosen was the time needed to compute the integration over 500 orbits, changing the relative tolerance applied inside the ode integration itself. This was an

interesting comparison because it displayed that the Gauss method was significantly better in all cases: in fact the red curve in figure 12 always lies under the other one. This was attributed to the fact that the *ode113* is an adaptive step integrator, so it changes the steps whenever it considers it more useful. The main advantage of Gauss Planetary Equations is that they are propagating the single parameter, which has a very slight variation during the time evolution, and rarely reaches the value zero, which represents the most challenging task for the calculator. In those exceptional cases, such that Ω is not defined because $e \rightarrow 0$ or $i \rightarrow 0$, the nonsingular equinoctial orbit elements were employed to avoid singularities. Instead the Cartesian integration Method used the Cartesian x , y , z coordinates of the satellite, which have greater variation in time, often passing through zero.

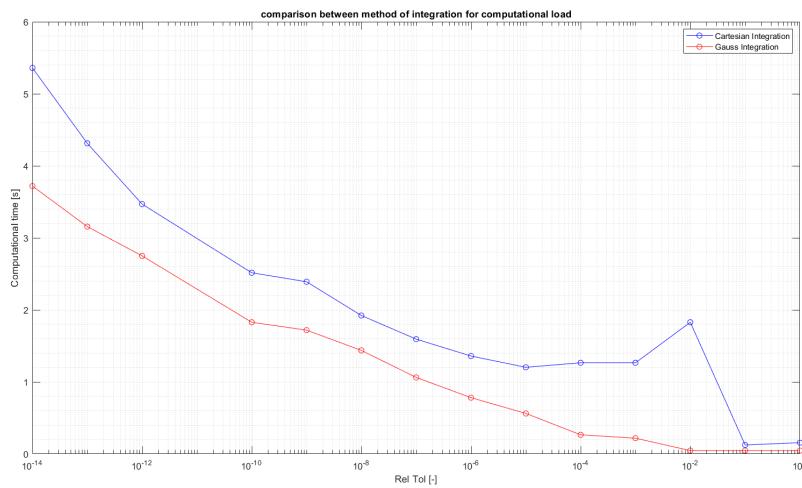


Figure 12: Computational cost comparison between Cartesian and Gauss propagation method varying the tolerances required

13 Spectral Analysis and Filtering of High Frequencies

The spectral analysis involved using the propagated parameters in the time domain and transforming them into the frequency domain in order to better identify patterns or characteristics from the noisy data. A frequency filter is used to purify the signal from the high frequency perturbations of the values obtained, in order to better grasp only the main trends of the data in the plots.

This was achieved by applying a cut off frequency to the data: applying a low cut-off frequency meant the higher frequency perturbations were cut, so only the oscillations with the bigger periods were conserved. Different cut-off frequencies were used, according to the results of the spectral analysis, where the main perturbations were registered, to provide the main periodical perturbation applied to the spacecraft.

The main low frequency behaviour were interpreted as linked to the period of the moon $T_{moon} = 2.360 \times 10^6 [s]$ thus a characteristic frequency of $f_{moon} = 4.236 \times 10^{-7} [Hz]$, and a

series of very close frequency peaks with a magnitude of circa $f_{sc} = 1.3 \times 10^{-5}[\text{Hz}]$ linked to the orbital period of the satellite $T_{sc} = 7.468 \times 10^4[\text{s}]$ and to the periodic interaction between the spacecraft and the moon on their revolutions around the Earth in a sort of synodic period of $T_{sp} = 7.693 \times 10^4[\text{s}]$.

Another check that was done was regarding which perturbation affected each parameter most, so that the most significant cut off frequency for each parameter was used.

The only parameter which shows a slightly different frequency domain is the semi-major axis that has no secular or periodic effects, and it shows a lot of low frequency noise.

a	e	i	Ω	ω
0.3×10^{-5}	1×10^{-5}	2.5×10^{-5}	1×10^{-5}	1×10^{-5}

Table 13: Cut off frequency chosen for the parameters [Hz]

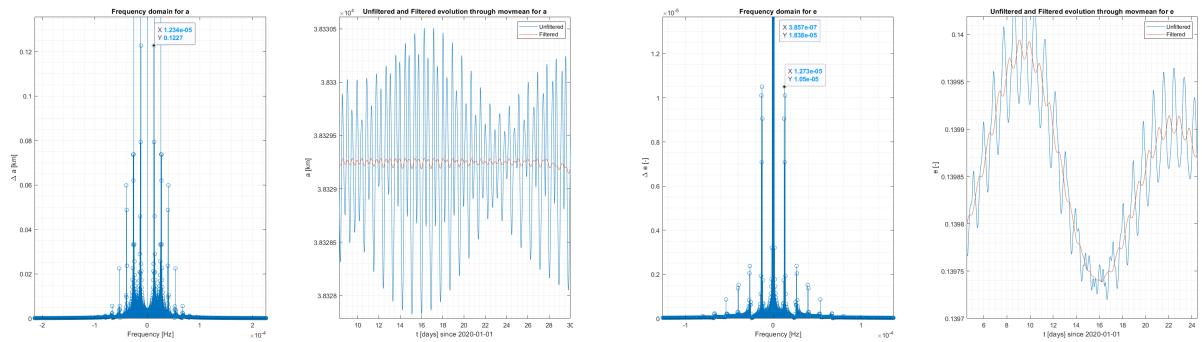


Figure 13: fft and filter of a, e

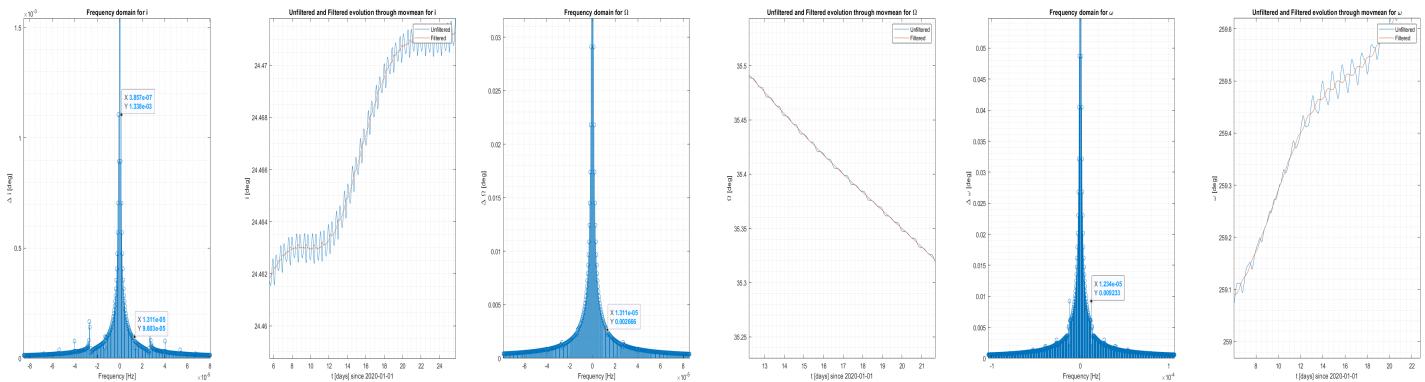


Figure 14: fft and filter of i, Ω, ω

14 Comparison with Real Data

To validate the consistency of the propagation method with respect to a real mission the team was requested to find a real satellite in the same orbital region and evaluate its motion with respect to the predicted one. **Chinasat 7**³ represented a valid choice due to the fact that it had a similar inclination i and semi-major axis a . Through the propagation of its orbit parameters using the two models illustrated before (Cartesian and Gauss) it was possible to compare them with real data obtained from **NASA/JPL's HORIZONS** data propagator. The results shown that the model is comparable in terms of $\Delta\Omega$ and $\Delta\omega$ which achieved a quasi-perfect match with real data, while for the other orbital parameters some deviations can be noticed, caused by the fact that we are neglecting the other main orbital perturbation such as the Solar Third Body Perturbation and the **SRP** (Solar Radiation Pressure) effect. We can assume the effect of aerodynamic drag negligible due to the fact that the *Chinasat 7* has a very large semi-majoraxis so never reaches the region of influence of that perturbing effect. In particular, Δi in the model gave an acceptable result since the major influence on that parameter is the Lunar effect which had been accounted for. Can be expected differences for e in a long term perturbation due to the solar radiation pressure.

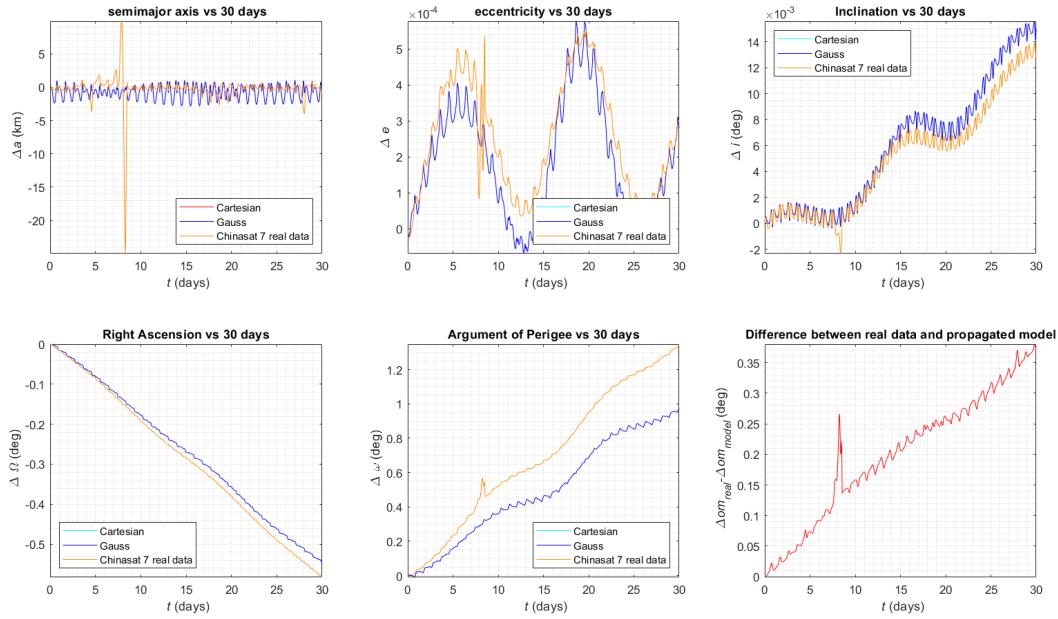


Figure 15: Comparison between Cartesian and Gauss propagation method of Chinasat 7 and real data

³Chinasat 7: telecommunications satellite built for the China Telcom Broadcast Satellite Corporation

15 Bibliography

References

- [1] Howard D. Curtis, H.D. *Orbital Mechanics For Engineering Students*. Butterworth - Heinemann, Daytona Beach, Florida, 2014.
- [2] Cage, P and Kroo, I and Braun. *Interplanetary trajectory optimization using a genetic algorithm, Astrodynamics Conference, 1994*
- [3] Kozai, Y. *On the Effects of the Sun and the Moon Upon the Motions of a Close Earth Satellite*. SAO Special Report 22, Smithsonian Institution Astrophysical Observatory, 1959.
- [4] Richard H Battin. *An Introduction to the Mathematics and Methods of Astrodynamics*. American Institute of Aeronautics and Astronautics, 1999.
- [5] Nasa/JPL's HORIZONS.<https://ssd.jpl.nasa.gov/horizons>.
- [6] David A Vallado. *Fundamentals of astrodynamics and applications*. Springer Science & Business Media, 2001.
- [7] R Storn. *On the usage of differential evolution for function optimization*. Biennial Conf. of the North American Fuzzy Information Processing Society, 1996.
- [8] Robert E Smith. *Space Environment Criteria Guidelines for Use in Space Vehicle Development, pag. 45*. NASA George C Marshall Space Flight Center, Technical Memorandum X 53273, 1965.
- [9] Douglas Isbell, Program Management 1998 Mars Missions (Mars Climate Orbiter), NASA Headquarters, Washington, DC.
https://www.nasa.gov/mission_pages/mars/missions/index-past.html
https://mars.nasa.gov/internal_resources/818