Escuela Técnica Superior de Ingeniería Aeronáutica y del Espacio
Universidad Politécnica de Madrid

Politecnico di Torino

Master of Science Thesis

Implementation of a manoeuvres optimisation prototype

AUTHOR: Laura De Filippis
EXPERTISE: Vehículos Espaciales
PROFESSIONAL SUPERVISOR: Isidro Jorreto López
ACADEMIC SUPERVISOR: Manuel Pérez Cortés
ACADEMIC SUPERVISOR: Manuela Battipede

2017-2018

Implementation of a manoeuvres optimisation prototype

by

Laura De Filippis

Higher Technical School of Aeronautical and Space Engineering,
Technical University of Madrid

Politecnico di Torino

Supervisors:	D. Isidro Jorreto López	GMV
	Dr. Manuel Pérez Cortés	ETSIAE
	Prof. Manuela Battipede	POLITO

February-March 2018

"*The most beautiful thing we can experience is the mysterious. It is the source of all true art and all science.*"

Albert Einstein

Contents

List of Figures

List of Tables

Nomenclature

ACRONYMS

focussuite ®	GMV's Flight Dynamics and Operations software solution	
AOP	Argument Of Perigee	deg
Cp	Solar radiation coefficient	
DE	Differential Evolution algorithm	
ECC	Eccentricity	
ETSIAE	Escuela Técnica Superior de Ingeniería Aeronáutica y del Espacio	
FDO	Flight Dynamics and Operations	
GEO	Geostationary Earth Orbit	
GMST	Greenwich Mean Sidereal Time	
GMV	GMV Innovating Solutions	
GODE	GOlem & Differential Evolution	
GOLEM	Geosynchronous Orbits Linearized Equations propagator with Manoeuvres	
GOPA	GOlem & Parametric Analysis	
GUI	Graphical User Interface	
INC	Inclination	deg
ISV	Initial State vector	
LST	Longitudinal Sidereal Angle	deg
PA	Parametric Analysis Optimiser	
POLITO	Politecnico di Torino	
RAAN	Right Ascension of the Ascending Node	deg

SADE	SApo & Differential Evolution	
SAPA	SApo & Parametric Analysis	
SMA	Semi Major Axis	km
SRP	Solar Radiation Pressure	Pa
SST	Semi-Analytic Satellite Theory	
SV	State vector	
TSV	Target State vector	
UPM	Universidad Politecnica de Madrid	
VOP	Variation of parameters	

GREEK SYMBOLS

λ	Longitude	deg
μ	Earth's gravitational parameter	$km^3 \cdot s^{-2}$
Ω	Right Ascension of the Ascending Node	deg
ω	Argument of Perigee	deg
θ	True Anomaly	deg

LATIN SYMBOLS

ΔV	Velocity Increment	$km \cdot s^{-1}$
a	Semi-major axis	s
a_{long}	Longitudinal acceleration	km/s^2
d	Number of elapsed seconds from the start of the current day	s
e	Eccentricity	$-$
e_x	x-component Eccentricity vector	$-$
e_y	y-component Eccentricity vector	$-$
EA	Eccentric Anomaly	deg
i	Inclination	deg
J_2	Second zonal harmonic of the Earth gravitational potential.	$-$
M	Mean Anomaly	deg

R_E	Earth radius	km
S_{body}	Spacecraft's sidereal angle	deg
S_{Sun}	Sun sidereal angle	deg
T_{Earth}	Earth Orbital Period	s
T_{orb}	Orbit Period	s
V_{mean}	Mean velocity	m/s

NOTATION

$\dot{x}$	Derivative of variable x with respect to time

PHYSICS CONSTANTS

μ	Earth's gravitational parameter	398600.4415 km^3/s^2
G	Universal gravitational constant	$6.67 \cdot 10^{-11}$ $m^3/(kg \cdot s^2)$
P_{solar}	Radiation Pressure at 1 AU	0.0456 kN/km^2

Abstract

Optimization is the main process thanks to which it is possible to make something, such as a design, a system, or a decision, as good as possible. In particular in the scientific field of expertise it refers to the mathematical procedures which allow to achieve this perfect result. Specifically optimising means getting the best variables values, among the available ones, evaluated under a set of constraints and according to a specific optimisation objective function.

Orbit propagation concerns the determination of the motion of any sort of body which can be found into space. The motion of a body, in accordance with Newton's laws, can be obtained starting from its initial state, which means its position and velocity in space at known time epoch, and considering forces which act on it. During the years, starting from Kepler to present day scientists, many mathematicians have tried, successfully in most cases, to discover and develop new mathematical, analytic, semi-analytic and numerical techniques involved with bodies' orbital trajectories.

SARROTO, **S**tation **A**cquisition, **R**elocation and **R**e-**O**rbiting **TO**ol, is the project developed during this training. Its main aim is performing an optimisation procedure related to manoeuvre events. This process is computed in order to reach the target final body state at certain epoch considering the best combination of velocity manoeuvre components and its epoch of occurrence.

Acknowledgements

This project meant a lot to me, not only because it gave me the possibility to learn more and more about subjects that often are superficially mentioned during university studying, but also for the big opportunity which has allowed me to achieve: the possibility to became a part of the working world in a field of expertise which I have dreamed for all my life.
The achievement of this goal is something that I could not obtained without some important people which help me in all the possible way.

First of all I have to thanks my professional tutor, Isidro Jorreto López, who helps me not only from the academic point of view, but also he supported me during all the project.

I have to mention Antonio Lozano Lima and all the FDO department of GMV, which as the great family which represents, has expressed friendship in all moment of day life, never let me feel an outsider, and for this reason I am pleased.
A special thanks to my academic tutors, Dr. Manuel Pérez and Prof.ssa Manuela Battipede, which gave me their confidence and possibility to develop this project under their supervision.

I am so grateful to have near me, although not physically, people who love me and express their love not with words but never making me feel alone and understanding how can be difficult to keep together work and personal life. For this, and for their love, thanks with all my heart. A sincere thanks to my 'sisters-in-law', because their presence every day of my life is a confirmation that life can be benevolent in a form that we don't expect.

Last but not least, I have to say thanks to the most important people of my life. To my brother, because even if he is not aware, makes me feel important and protected. To my mother, because never stops taking care of me, and taking any possible risks, leads me to the right choice, without her I would not be here in this moment. To my father, because he is the only one who knows the deepest dreams hidden in my heart, and never stops encouraging me to achieve them, also when it seems impossible.
For your supports and vicinity, thanks.

Dedicated to people I love,
and love me back.

Chapter 1

Introduction

Optimisation and bodies' orbit propagation can be considered two fundamental problems among the scientific community.
In fact, firstly focusing on optimisation, it has to be said that achieving the perfection, in all fields of expertise, has always been one of the main objectives. In particular this research has been conducted in an exhaustive way in all the scientific fields of discovery, from the scientific and engineering ones, to business decision-making and general industry, where the optimisation approach is largely required.

So, this work has born to became part of this scenario, fitting in the context which concerns spacecraft's flight dynamics field of development. The main objective of this project has been the development of a prototype for spacecraft's manoeuvres optimisation. This tool takes the name of SARROTO which stands for *Station Acquisition, Relocation and Re-Orbiting TOol*. It consists in a software fully implemented in Fortran 90. Focusing on its general description, there are two main parts that have to be considered

- the software architecture, which consists in analysing the tool general design from the computational point of view;
- the program interface with users which allows to define the inputs/outputs procedure.

Starting from the first aspect, basically this program implements an orbital propagation followed by an optimisation process which concerns the manoeuvre event. This means that the principal feature which has to be taken into account when executing this tool is the orbit to be propagated which involves manoeuvre events to be optimised.
This software leads to the intent of putting together the propagation and optimisation phases. The introduction of subprograms in the main tool allows to achieve these two purposes. In particular the word **subprograms** refers to specific propagators and optimisers. Therefore SARROTO consists in a collection of these tools. Before starting with its general description, it is necessary to introduce some preliminaries about flight dynamics subject. A generic manoeuvre event can be described through two variables:

– Velocity vector increment, ΔV, which is the definition of manoeuvre itself. It is defined as a set of three components.

– Epoch at which this event happens.

Within this thesis it will be assumed the **RSW** as coordinate system for the satellite's motion along the orbit. Its representation can be seen in Fig. 1.1. In this system velocity is composed of three components: Radial, Along-Track and Cross-Track. The radial component always points from the Earth's centre to the satellite following the radius vector. The along-track is orthogonal to the radius vector and its direction is the one indicated by the velocity vector. The S axis, as appears in Fig 1.1, results aligned with the velocity vector only for circular orbits or at apogee and perigee of elliptical orbits. The cross-track is normal to the orbital plan. More information about orbital manoeuvres can be found in appendix section B.

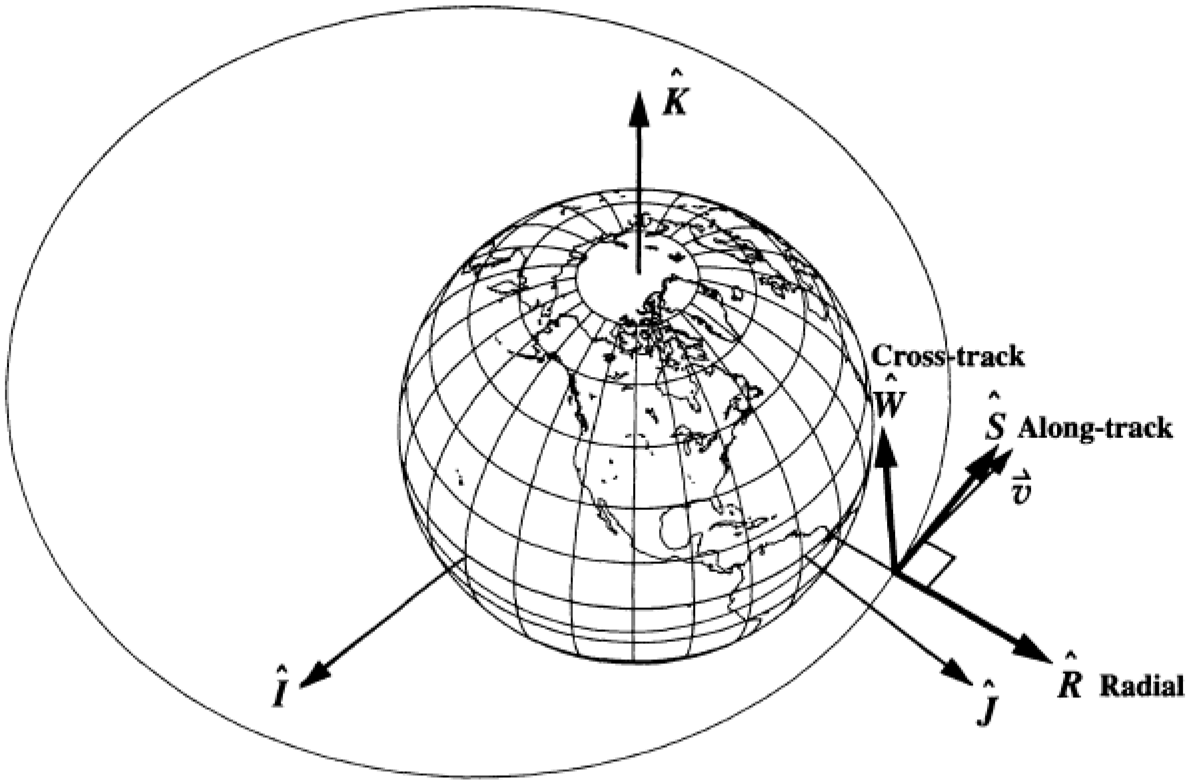

Figure 1.1: RSW satellite coordinate system representation.

The optimisation phase consists in looking for the best value for the previous mentioned variables: the three components of velocity and epoch of each manoeuvre. The optimising working process depends on the selected optimiser. Within the optimisation process it has to be introduced the orbit propagation. Propagating an orbit, in the aerospace field of interest, means predicting the position of one desired body during time. To compute this position at certain epoch it is necessary to know some fundamentals parameters. These ones are the body initial

state, composed of the state vector parameters, and forces which act on it. State vector parameters refer to a wide collection of coordinates system, as illustrated in appendix section A.2.
In SARROTO, users have to introduce this initial state as input. A final epoch of propagation and desired parameters types have to be selected. The set of these last parameters will correspond to the target state vector. The combination of manoeuvre events have to be considered inside this propagation.
Once that first propagation has been computed, optimisation takes place considering the cost function. This last one lies in a mathematical equation which brings together the final state vector obtained with propagation and the target one. The expected result which should come up from this relation is its minimum value, obtained with the best combination of manoeuvre parameters mentioned above. This process will be examined in greater detail in related section 2.2.

Propagation and optimisation are the basis of this software. For this reason their general characteristics will be explained below.
Today it is possible to find a wide variety of optimisation algorithms, in particular in this project it has been implemented the differential evolution optimiser, and the parametric analysis optimiser. The first is one of the evolutionary algorithms created in 1995, while the second is a tool developed during this internship. Their operating method will be illustrated in next chapters. These two optimisers are the main features through which it is possible to compute the best body trajectory among all the considered solutions.

As yet introduced, orbit propagation is what defines the motion of a general body along is trajectory. Before starting scientifically defining what is an orbit propagation it has to be mentioned why this last said aspect is of particularly significance.
Since humans rolled their eyes to the sky, they have been interested in looking up at the stars, planets and other celestial bodies, both for desire of knowledge and for religious reasons. Astronomy, as general field of expertise, has always been one of the most studied and age-old subject.
Theoretical basis at the mathematical astronomy, so as it is known today, can be referred to Kepler followed by Newton and other seventeenth century mathematicians such as Laplace, Legendre, Cowell an others. Their interest involved planetary motion, developing mathematics theories describing their movement.
Today thanks to these scientists initial attempts all these features are known. In addition to the very purely physical interest enriched during years of study, the motion of planets, and of general objects which are into space is pursued due to the satellite business.

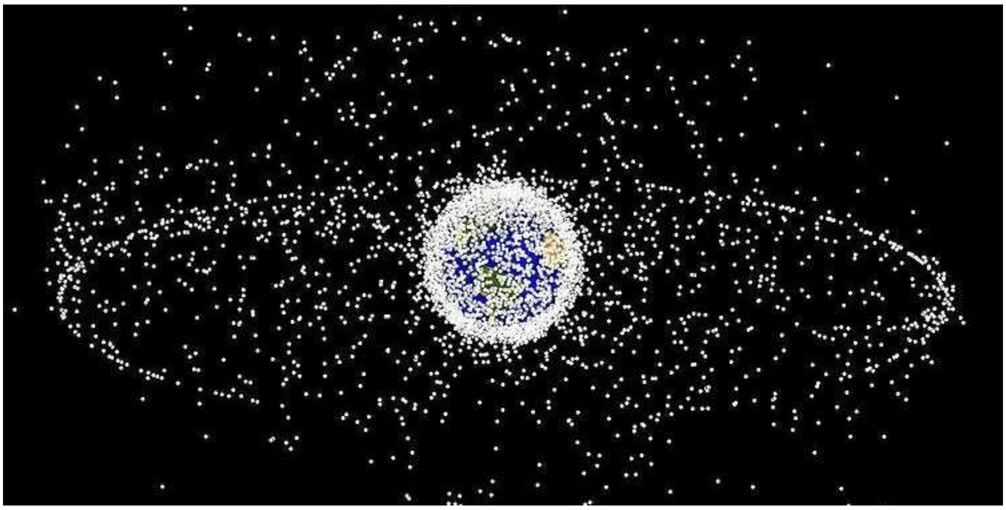

Figure 1.2: Spacecrafts orbiting around Earth.

It has to be taken into account that, with the advance of technology, the space has begun to be populated not only with spacecraft but also with their ”dead components”, referring to what is nowadays called space debris. These last are part of the monitoring activity which is performed in order to avoid possible collision between spacecraft and objects. Fig. 1.3 shows the increment along time of the objects orbiting around Earth.

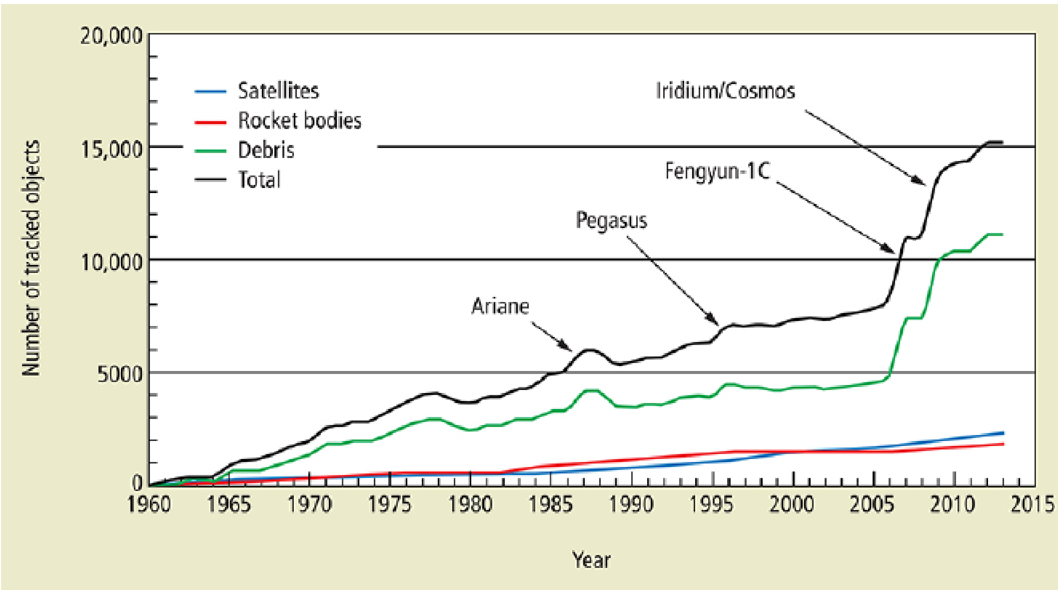

Figure 1.3: Number of objects orbiting around Earth.

Propagation is involved in this context since it plays a fundamental role in calculating these objects orbit so as evaluating collision risk could be possible.
Once approached to the pragmatic aspect of the aerospace business related to propagating orbital bodies, it is possible to deal with its mathematical and theoretical aspect. The objective of propagators developed during past years is computing the best possible solution in terms of bodies future positions and velocity. The efficiency of a propagator depends on the way it is implemented. In particular during this project readers will face three types of propagators:

- **Analytic**: this type of propagator is considering a low fidelity one, only based on mathematical formulae. Despite their low efficiency, computationally these ones are the fastest and can provide a first approximate solution. **GOLEM** is the geostationary analytic propagator entirely developed during this internship.
- **Semi-Analytic**: this propagators family is in a middle way between the completely analytic and the numerical one. They are based on analytic formulae but, in addition, incorporate some numerical techniques to improve results precision. **SAPO** is the semi-analytic propagator developed in recent years in GMV's FDO section.
- **Numerical**: it consists in a propagator which numerically integrates the equation of motion. **PROPAG** is the one developed in GMV's FDO section.

A general overview has been presented in this introduction in order to make the reader more comfortable within the project thesis presentation. All these aspects which have been mentioned will be better explained in next chapters and in some cases in greater detail in dedicated appendix sections.

1.1 Thesis overview

This thesis can be divided in five major parts.

1. The first part is dedicated to mathematical concepts used to develop this project. This section includes discussion about orbit propagation theories.
2. The second part illustrates the main programme which has been developed during this internship, the SARROTO programme. Its accurate description, general architecture and code development can be found in chapter 3.
3. The third is dedicated to results and is described in chapter 4. The main aim of this major third part will be to show SARROTO working process.
4. The next one is dedicated to SARROTO future improvement and final general remarks about this project.
5. Last part, the appendix section, illustrates demonstrations, explanation of parameters, variables, and general mathematics expressions related to orbit. These features are presented in order to give all the readers the information to understand all the procedures and techniques used in this work.

This project has born with the intention to develop a code which could be used with different orbit and conditions. This goal will be achieved thanks to a modular design, which allows using different propagators and optimisers, as well as a highly flexible configuration to make SARROTO suitable for many different missions. So, the work done during this internship is only the beginning of a harmful development. As it will be discussed in chapter 5, many improvements are planned. These ones will enrich this first SARROTO phase of development.

1.2 Technological facilities and software

In order to come up with this project it has been necessary to use specific computer software, such as the ones involved in coding and numerical computation.

In particular *Fortran 90*® is the programming language which has been used to develop the coding part of the project. It is still in operation in aerospace industry even today thanks to its high reliability, and specifically used in FDO department in GMV company. A better description about this coding language is presented in appendix section H.

The implementation of this software has been carried out through LINUX operating system.

SARROTO has been integrated into ***focussuite***® which is the FDO software created in GMV for flight dynamic satellites mission analysis and operations. The ***focussuite*** environment integrates different tools. Some of them have been used within this work. One of these features is the ***focussuite*** graphical interface. This tool, **Graphical User Interface** abbreviated to the acronym GUI, comes up from the necessity to give the user the chance to interact with general program routines. In fact, as it should appear clearer in next chapter, this interface allows client to make appropriate choices about cases he would like to perform.
With respect to graphs and plots, they have been created using an other of these tools, named ORBCOMP, developed in FDO Section. Further information related to ***focussuite*** are mentioned in the appropriate appendix section C.

Block diagrams and flow charts have been composed with ***Visio***®. Bar diagrams have been created with ***Excel***®.

Chapter 2

Mathematical underlying theory

2.1 Propagation Theory

Propagation, together to optimisation, form the basis of this project. Nowadays a wide selection of methods of propagation exists, which differ in the mathematical framework used to compute final result. In particular in this project the focus is on the ones based on two-body problem and semi-analytic theory. Different approaches are used, always starting from the two body problem. As mentioned in chapter 1, and later explained in Sec. 3.1, different types of propagators have been used in this project. Two of them have been introduced into SARROTO. A third one has simply been used to make comparison with the first two.

1. The first propagator introduced is **GOLEM**. This propagator has been fully developed during this internship. It is considered quasi-analytic. Indeed it makes usage of analytic equations but at the same time it involves numerical techniques for dealing manoeuvre events. Its mathematical background will be discussed in section 2.1.2. Its architecture can be found in dedicated section 3.1.1.1 of chapter 3. Its graphical interface is explained in section 3.2.4.1.

2. The second one which has been introduced into SARROTO is **SAPO**. It is considered semi-analytic since perturbations effect are involved in two-body problem equation. Differently from the previous mentioned propagator, SAPO has been developed in past years in FDO and fully integrated during this internship in SARROTO.
   Theories behind this propagator will be discussed in sec. 3.1.1.2 and a more accurate description about is mathematical background can be found in appendix F. Its structure and placement inside ***focussuite*** environment can be found respectively in sections 3.1.1.2 and 3.2.4.2.

3. The third and last one one is the propagator belonging to the group of the numerical one, **PROPAG**. It uses numerical techniques to get final solution. It is a propagator software of high reliability, developed in GMV and used in important organisations such as ESA, European space agency, and for commercial satellites operators such as Eutelsat, Eumelsat, Hispasat. Interested readers could find more information about this propagator in

appendix section E

It has to be said that what it is considered important in a propagator development is the computational speed and the precision and accuracy obtained in the final solution. GOLEM is the faster between the mentioned one, but it is PROPAG which leads to the better solution. SAPO is a middle ground between them: expected results are obtained with good precision in not so long computational time.

2.1.1 SAPO's mathematical theory

Firstly it has to be mentioned that SAPO mathematical approach refers to the semi-analytic satellite theory as described in [5]. First general overview will be treated in this paragraph, leaving the mathematical experts and adventurer readers to give a look to appendix F, where the exact mathematical steps have been faced.

Before discussing about this theory it is necessary to focus on the main definition of mean and osculating elements. These elements are used to describe the orbit of any astronomical body. The difference between these two vector of elements is the way of considering perturbations.

- The osculating elements are used to describe the satellite's motion along an unperturbed orbit if perturbationsinstantaneously disappear. In absence of perturbations these elements would not change.
- A perturbed orbit can be described through the mean elements since perturbations always occur within a real orbit. They can be defined as the set of osculating elements to which secular and periodic perturbations have been included.

The starting point is the two-body problem equation to whom disturbing functions have to be introduced. In particular the equations presented below, Eq.2.1, represents the Cartesian equation of motion, and as it will be discussed in the appendix section F, it is a first representation which has to be translated in terms of averaging generalised equations.

$$\ddot{r} = \frac{-\mu}{r^3}\vec{r} + \vec{q} + \nabla\mathscr{R} \tag{2.1}$$

where $\vec{q}$ is the contribute to acceleration due to not-conservative forces, while the other one $\nabla\mathscr{R}$ is the contribution of the conservative disturbing forces. In order to solve this last one it can be possible referring to the **VOP**, the Variation of Parameters method, which consists in a mathematical method used to solve inhomogeneous linear ordinary differential equations. As it can be found in [14], a good way to solve this problem is to isolate the short periodic disturbances from the long-periodic and secular contributions in order to propagate the mean element rates.It has to be reminded that short and long perturbations refer to the same group which is the periodic perturbation contribution.
On the other hand the secular perturbation have to be introduced in a separate group. An accurate

description of these perturbed phenomena can be found in appendix section G.
Eq. 2.1 can be rewritten using the solution of the undisturbed problem which can be expresses as

$$\vec{r} = \vec{x}(a,e,i,\Omega,\omega,M,t) = \vec{x}(\vec{c},t) \tag{2.2}$$

$$\vec{v} = \dot{\vec{x}}(a,e,i,\Omega,\omega,M,t) = \dot{\vec{x}}(\vec{c},t) \tag{2.3}$$

where with $\vec{c}$ represent a single vector in which are regrouping all the classical elements. Finally the no perturbed and perturbed two-body equations result to be

$$\ddot{\vec{x}} + \frac{\mu\vec{x}(\vec{c},t)}{x(\vec{c},t)} = 0 \tag{2.4}$$

$$\ddot{\vec{x}} + \frac{\mu\vec{x}(\vec{c},t)}{x(\vec{c},t)} = a_{\vec{pert}} \tag{2.5}$$

In the perturbed problem, differently from the unperturbed one, it has to be considered that the orbital elements, defined in vector $\vec{c}$, have to change in time. The variation of these six orbital elements can be summarised in a differential equation of first order

$$\frac{d\vec{c}}{dt} = f(\vec{c},t) \tag{2.6}$$

In order to solve this it is necessary to proceed distinguishing between conservative and not conservative forces The following parameters rate of change are obtained for the separated cases:

- Contribution to variation due to conservative forces is known as ***Lagrange planetary equation***, differential equations, better explained in [14]

$$\begin{aligned}
\frac{da}{dt} &= \frac{2}{na}\frac{\partial R}{\partial M} \\
\frac{de}{dt} &= \frac{1-e^2}{na^2e}\frac{\partial R}{\partial M} - \frac{\sqrt{1-e^2}}{na^2e}\frac{\partial R}{\partial \omega} \\
\frac{di}{dt} &= \frac{1}{na^2\sqrt{1-e^2}\sin i}\left(\cos i\frac{\partial R}{\partial \omega} - \frac{\partial R}{\partial \Omega}\right) \\
\frac{d\omega}{dt} &= \frac{\sqrt{1-e^2}}{na^2e}\frac{\partial R}{\partial e} - \frac{\cot i}{na^2\sqrt{1-e^2}}\frac{\partial R}{\partial i} \\
\frac{d\Omega}{dt} &= \frac{1}{na^2\sqrt{1-e^2}\sin i}\frac{\partial R}{\partial i} \\
\frac{de}{dt} &= \frac{1-e^2}{na^2e}\frac{\partial R}{\partial e} - \frac{2}{na}\frac{\partial R}{\partial a}
\end{aligned} \tag{2.7}$$

- Contribution due to no-conservative forces, which can be found in [14], and Lagrange VOP equations are

$$
\begin{aligned}
\frac{da}{dt} &= \frac{2}{n\sqrt{1-e^2}}\left\{e\sin f F_R + \frac{a(1-e^2)}{r}F_S\right\} \\
\frac{de}{dt} &= \frac{\sqrt{1-e^2}}{na}\left\{\sin f F_R + \left(\cos f + \frac{e+\cos f}{1+e\cos f}\right)F_S\right\} \\
\frac{di}{dt} &= \frac{r\cos u}{na^2\sqrt{1-e^2}}F_W \\
\frac{d\omega}{dt} &= \frac{\sqrt{1-e^2}}{nae}\left\{-\cos f F_R + \sin f\left(1+\frac{r}{a(1-e^2)F_S}\right)\right\} + \\
&\quad - \frac{r\cot i\sin(\omega+f)}{\sqrt{\mu a(1-e^2)}}F_W \\
\frac{d\Omega}{dt} &= \frac{r\sin f+\omega}{na^2\sqrt{1-e^2}\sin i}F_W \\
\frac{dM}{dt} &= \frac{1}{na^2e}[(a(1-e^2)\cos f - 2er)F_R - (a(1-e^2)+r)\sin f F_S]
\end{aligned}
\tag{2.8}
$$

where the components F_S, F_R, F_W represent the specific components force perpendicular to the radius vector, along this last one and normal to the orbital plan, respectively.

2.1.1.1 Semi-analytic satellite theory and its limitations

The mathematical approach for the development of this propagator makes use of the equinoctial elements representation. This coordinate system has been preferred to others in order to avoid the singularities which could appear if $i \to 0$ or $e \to 0$. In addition, for definition, the semi-analytic theory makes use of the mean elements, differently from the numerical propagator which uses the osculating ones.

This theory can be used when short and long term in perturbations group can be separated. Indeed, this method is based on averaging procedure applied to differential equations of motion over a fast-moving angular variable. Once these averaged equations have been calculated, the objective is to use these ones to predict the motion of the slowly varying elements. For this reason this method does not apply very well to perturbations which do not have averaging operator that fulfils the required properties. Among these non-averageable variables can be listed

- Atmospheric drag on an asymmetric vehicle
- Solar radiation pressure on an asymmetric vehicle

- Continuous thrust, with some cases exception, and impulsive thrust. In particular in this second case, it is possible to consider its effect adding the ΔV components of manoeuvre to osculating elements just converted in position and velocity components. Then, once perturbation has been taken into account, reconverting the obtained result into equinoctial elements in order to obtain the osculating and subsequently the mean elements.

So, summarising the main points, SAPO has been developed on the complete semi-analytic satellite's theory introduced by [5] and considers most of the perturbations which act on the satellite.

2.1.2 GOLEM's mathematical theory

GOLEM, which stands for Geosynchronous Orbits Linearized Equations propagator with Manoeuvres, is a geosynchronous analytic propagator entirely developed during this internship.
Its main aim is discovering and obtaining a first faster solution instead of using previous propagators. SAPO, PROPAG, for example, although lead to more precise solutions, take up a greater amount of time for computation, which is especially relevant when propagation is mixed with optimisation.
Actually, it has to be remarked that some shades of semi-analytic features have been introduced, as explained in next lines.

At first, a fully analytic version had been developed, with equations which will be shown later in the paragraph. During its implementation, it has become clear that it would not have been efficient considering manoeuvres events. Indeed in this first version, the manoeuvre effects are included in the equations through the presence of the along track velocity component.

At a later stage, the way of considering manoeuvres has been improved by the use of a numerical procedure step. This last one essentially consists in adding the increment of velocity vector components of manoeuvre to the state vector which results at manoeuvre epoch. The exact computation procedure will be explained in section 3.1.1.1.

GOLEM has been developed starting from the Kepler's Equation. This equation appears for the first time in the *Astronomiae Copernicanae*, the first astronomy textbook based on the Copernican model written by Kepler himself, which for this reason gives the name to the equation. Thanks to this equation, it is possible to calculate time at which a certain position is reached.

The main parameters used for solving the planetary orbit motion which can be distinguished into this equation are the mean anomaly and the true anomaly. In particular the true anomaly, indicated with θ, represents the position of a planet or any other celestial object in its orbit around the Sun, or other celestial object taken as reference. The true anomaly is the angle between the direction to the perifocus and the direction to the planet, as seen from the focus of the orbit. The Mean anomaly can be calculated as follows, where M corresponds to this last said, while E is the Eccentric Anomaly

$$M(t) = E(t) - e \cdot \sin E(t) \tag{2.9}$$

In order to evaluate this value, since this Eq. 2.9 is a transcendental equation, it is necessary to refer to the classical orbit element as it follows

$$E = 2 \cdot \arctan \sqrt{\frac{1-e}{1+e}} \cdot \tan \frac{\theta}{2} \tag{2.10}$$

Eccentric anomaly can be solved through Eq. 2.10 only if orbit elements e, θ are known. If not, given M, it is necessary to use numerical analysis or expansions series to solve **E** starting from ¡Eq. 2.9 .
Once defined these two equations, it is possible tracing back to the complete list of equations used to implement this propagator.
It will be illustrated how perturbations and manoeuvre events are introduced into this analytic propagator.
In GOLEM perturbations taken into account are the effects of Non-spherical Earth and solar radiation pressure. These phenomena lead to changes in orbital elements values. Manoeuvres, by definition, induce variation in the elements values.

- One of the non-spherical Earth effect perturbing a spacecraft is considered through the second zonal harmonic coefficient J_2. Mathematical equations which introduce this term are the ones related with change of rate of orbital elements: Mean Anomaly, Arg. of Perigee and Right Ascension.

$$\delta M = J_2 \cdot \frac{sec_{day}}{T_{orb}} \cdot \frac{(2 - 3\sin^2(i))}{(a^2 \cdot (1-e^2)^2)} \rightarrow \text{Mean Anomaly rate computation} \tag{2.11}$$

$$\delta \omega = J_2 \cdot \frac{sec_{day}}{T_{orb}} \cdot \frac{(4 - 5\sin^2(i))}{(a^2(1-e^2)^2)} \rightarrow \text{AoP rate computation} \tag{2.12}$$

$$\delta \Omega = -2 \cdot J_2 \cdot \frac{sec_{day}}{T_{orb}} \cdot \frac{\cos(i))}{(a^2(1-e^2)^2)} \rightarrow \text{RAAN rate computation} \tag{2.13}$$

Into the effects of the non-spherical Earth the longitudinal drift has to be included. In particular this longitudinal variation which affects the spacecraft's semi-major axis is due to the equatorial ellipticity of Earth. This deviation can be calculated considering the tesseral harmonics coefficient J_{22}. Longitudinal value and related acceleration can be calculated for a GEO orbit using some simplifications. Please refer to [2]. The deviation in semi major axis due to longitudinal acceleration is calculated as

$$\delta a = -2 \cdot \left(a_{long} \cdot (\Delta T \cdot sec_{day}) - \Delta V_{along}\right) \cdot \frac{T_{rot}}{2\pi} \tag{2.14}$$

where the variable a_{long} represents this mentioned acceleration.

- Solar radiation pressure acceleration for a satellite in GEO Orbit only affects the two components of the eccentricity vector.
  Its effect on spacecraft are difficult to predict since surfaces reflectively and properties in space change from values established in laboratory. However for GEO S/C solar radiation effects can be described through the *effective cross-section to mass ratio* parameter σ which depends on the spacecraft mass and geometry.
  The variation of eccentricity vector direction depends on the sidereal angle of the Sun. It can be obtained trough the two equations represented below.

$$e_x(t) = e_{x_0} - \frac{1}{2}\frac{3P}{2V} \cdot \sigma \cdot (\sin S_{Sun}) \cdot \delta T + \delta e_x \tag{2.15}$$

$$e_y(t) = e_{y_0} + \frac{1}{2}\frac{3P}{2V} \cdot \sigma \cdot (\cos S_{Sun}) \cdot \delta T + \delta e_y \tag{2.16}$$

  In the above equations S_{Sun} indicates the Sun sidereal angle. The last term on the right of each equation, respectively δe_x and δe_y, defines the eccentricity vector rate of change due to manoeuvre events, as it will be explained in next related paragraph. The term $\frac{3P_{solar}}{2V}$ is equal to

$$\frac{3P}{2V} = \frac{3}{2} \cdot \frac{4.56 \cdot 10^{-6}}{3.075 \cdot 10^{3}} \cdot 86400 = 0.00019 \;\; kg/m^2/day$$

  where $V = 3.075\, km/s$ is the spacecraft's orbital velocity in GEO and $P = 4.56 \cdot 10^{-6}\, N/m^2$ the solar pressure.

- As can be found in appendix section B, depending on the manoeuvre event type, elements could vary their value or remain unchanged. As mentioned at the beginning of this section, two method of considering manoeuvre effect have been taken into account in developing GOLEM.

  1. The first method developed consisted in taking into account the manoeuvre contribution analytically. Indeed the increment of velocity ΔV had been introduced inside equation. The manoeuvre type allowed with this simplification was the East-West manoeuvre. The along-track velocity was the only component among the three to be considered in propagating orbit. This type of manoeuvre affects the eccentricity and the semi major axis, inducing their rate of change, as it appears in Eq. 2.17 and in Eq. 2.18 and 2.19

$$\delta a = -2 \cdot \left(a_{long} \cdot (\Delta T \cdot sec_{day}) - \Delta V_{along}\right) \cdot \frac{T_{rot}}{2\pi} \tag{2.17}$$

$$\delta e_x = 2 \cdot \frac{\Delta V_{along}}{V_{mean}} \cdot \cos S_{body} \tag{2.18}$$

$$\delta e_y = 2 \cdot \frac{\Delta V_{along}}{V_{mean}} \cdot \sin S_{body} \tag{2.19}$$

Eq. 2.18 and 2.19 describes the eccentricity vector components rate of change. The variable S_{body} defines the spacecraft's sidereal angle. In these equations V_{mean} represent the mean velocity, which for orbits with small eccentricity (i.e. the circular orbits) can be calculated as

$$V_{mean} = a \cdot \frac{2\pi}{T_{Earth}} \tag{2.20}$$

where **a** is the semi-major axis, T_{Earth} defines the Earth orbital period that for a satellite in GEO orbit is equal to $T_{Earth} = 86164.090530\,s$

2. In order to improve the way of dealing with manoeuvre events, a different approach is considered in the second method. A numerical technique is used rather than the analytic formulation. This numerical approach consists in adding the increment of velocity to the satellite's state vector reached at the manoeuvre epoch. This process could be described as

$$\frac{d\vec{x}_i}{dt} = \frac{d\vec{x}_0}{dt} + \Delta V \tag{2.21}$$

With this new numerical approach not only along-track, but also the radial and the cross components will be included in the propagation.
The analytic contribute due to ΔV_{along} introduced into Eq. 2.17 disappears. The eccentricity rate of change defined in Eq. 2.18 and 2.19 takes zero value. So, the equation 2.17 related to SMA rate of change and the ones related to eccentricity vector Eq. 2.15, 2.16 become

$$\delta a = -2 \cdot \left(a_{long} \cdot \Delta T \cdot sec_{day}\right) \cdot \frac{T_{rot}}{2\pi} \tag{2.22}$$

$$e_x(t) = e_{x_0} - \frac{1}{2}\frac{3P}{2V} \cdot \sigma \cdot (\sin S_{Sun}) \cdot \delta T \tag{2.23}$$

$$e_y(t) = e_{y_0} + \frac{1}{2}\frac{3P}{2V} \cdot \sigma \cdot (\cos S_{Sun}) \cdot \delta T \tag{2.24}$$

2.2 Cost Function Mathematical definition

As mentioned in the introduction of this project, optimisation can be computed thanks to a mathematical equation which relates the final and the target state vector. About these last two state vectors, recall that the first is obtained through propagation, the second is defined by users. The value obtained from this mathematical equation is called *Cost Function*. The main aim of this function is allowing reducing not only fuel consumption but also achieving the desired final orbit position. The expected result for an optimum solution is the lowest value that this variable can assume.
This function can be represented as

$$Cost\ Function = Constraints + \sum \Delta V \tag{2.25}$$

With *Constraints* on refers to the following statement

$$\begin{cases} Constraints = \sum Weight_i \cdot \left(\frac{Target\ SV_i - Final\ SV_i}{Tolerance_i}\right)^2 & \text{if } ||Final\ SV_i - Target\ SV_i|| > Tolerance_i \\ Constraints = 0 & \text{if } ||Final\ SV_i - Target\ SV_i|| < Tolerance_i \end{cases} \tag{2.26}$$

where the subscript *i* indicates the considered target element. The variables *weight* and *tolerance* have to be introduced as inputs. These last two are related to the target state vector elements that client would like to obtain as output. The weight is a measure of the element related importance in achieving the best solution. The tolerance indicates the maximum deviation without penalty between the final state vector and the target one. It has to be pointed out that the check process in Eq. 2.26 has to be done for all the elements selected for the target state vector. As it can be seen in Eq.2.26, the deviation from target state vector appears in a squared term. This is because a solution which moves away from the target is more significant than one which is near to this last.

This function works in such a way that the priority is ensuring that through optimisation the target state vector is achieved. On the other hand fuel consumption is the other parameter which has to be optimised. This optimisation can be computed trying minimising the ΔV since these two are correlated. The fuel consumption becomes significant to the cost function computation when the target takes zero value. When this last is not verified, the cost function results negatively affected by the *Constraints* contribute in the sum in Eq. 2.25. Indeed, as mentioned at first, the minimum value of cost function variable is the one which leads to the best optimised solution.

Chapter 3

SARROTO

In this chapter it will be presented the SARROTO program. Its description is divided in two part:

- SARROTO General Architecture
- SARROTO Users Interface

3.1 Software Architecture

SARROTO, which stands for **S**tation **A**cquisition, **R**elocation and **R**e-**O**rbiting **TO**ol, is a project born from the idea to create a manoeuvre's optimisation software. This SW will be able to use different type of propagators and optimisers, in order to achieve the client's necessity and requests.

Since this program has been launched during internship started in July 2017, only few optimizers and propagators have been introduced. Some of them preexisting, others have been created during SARROTO development as mentioned in chapter 1. The main characteristic of SARROTO is its general coding structure and its high flexibility. This allows implementations of various GMV developed tools.
The main characteristics of this project can be resumed as follows.

- SARROTO is structured in such a way that it is possible to choose, through the focussuite interface, discussed in section 1.2 and in appendix C , the solution which best fits the client requests.
- It has been fully written with **Fortran 90**. In particular it has been divided in different modules and subroutines. This modular structure allows to compute specific objective in different programs and a easier software future enlargement.
- Subroutines and modules developed for other programs and written in Fortran 90 have been utilised. Among them, coordinates and reference frame conversion, manoeuvres handling and state vector allocation modules can be mentioned.

- Possibility to choose, not only different propagators and optimizers, but also the following features
  - A great variety of target state vector elements to obtain as final result after computation. These elements can be selected from a list which contains a wide selection of coordinates types. Associated to these elements both their weight inside computation and tolerance for the wished results can be found.
  - Fixed or variable number of manoeuvres. In the first case client defines number of manoeuvre presents in the propagation. The second option corresponds to the maximum number of manoeuvre between which looking for the optimum solution.
  - Numbers of parameters to be optimised or to be fixed. These ones are the ΔV components of velocity (i.e. cross, along-track and radial) and manoeuvre epoch. It has to be pointed out that client can choose which of these would like to be optimised.

SARROTO, as said before, is structured in different modules. Thanks to its *block structure* it is possible to choose different options for optimisation cases. At the top of the main program there is the **Manager**, which is the block code that runs all the high-level modules related to the acquisition of data, computation and representation of results. It can be defined as the module which runs the main programme.

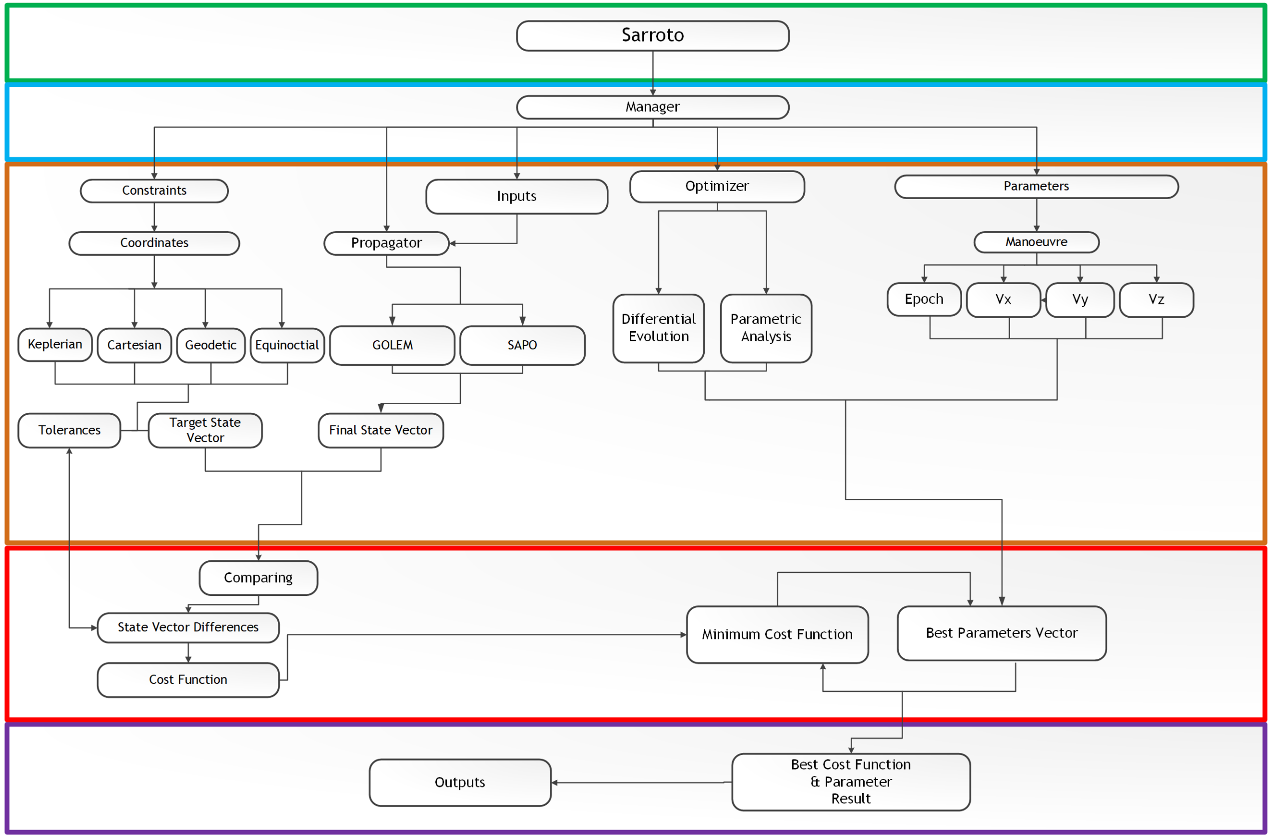

Figure 3.1: Block diagram for SARROTO working tool.

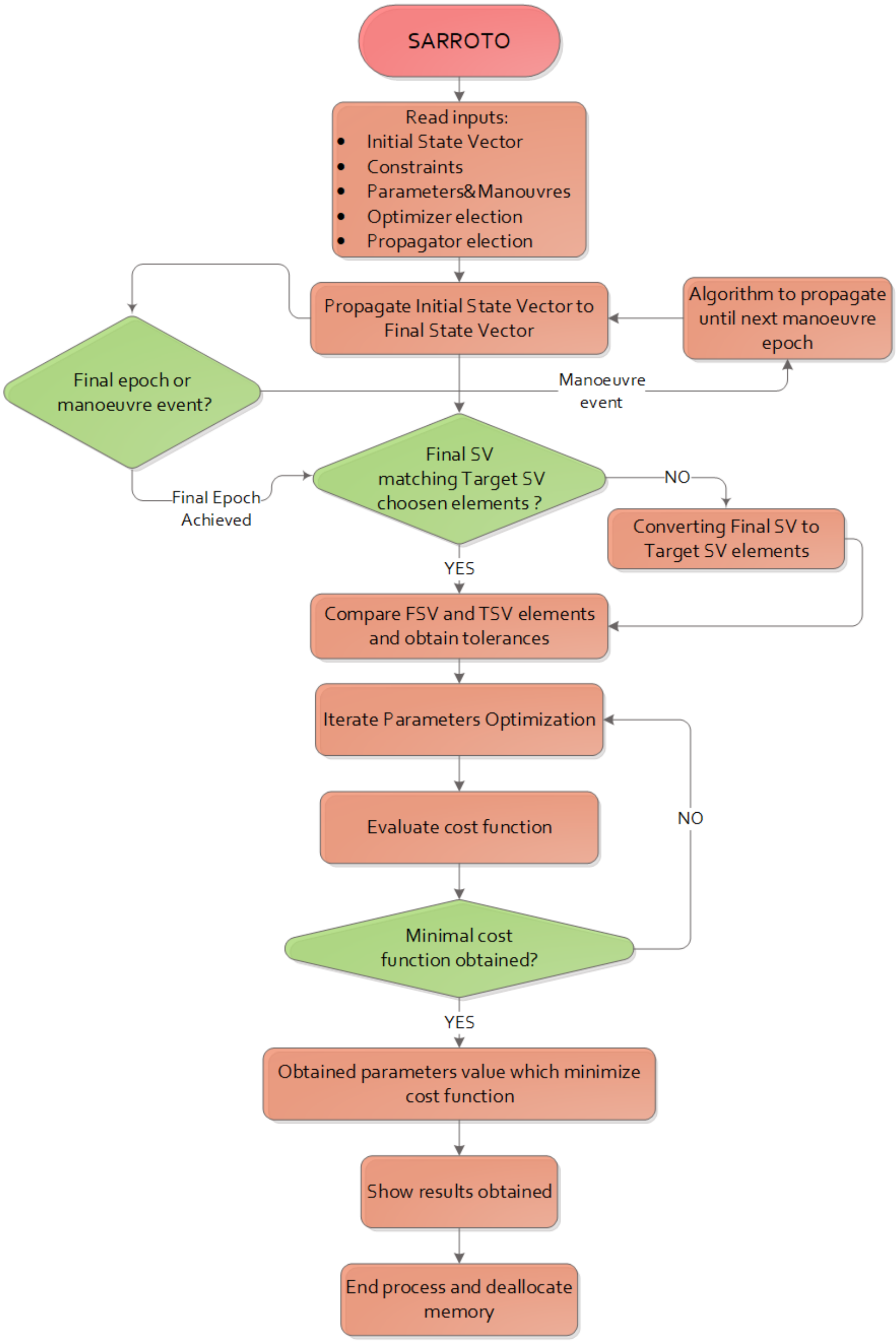

Figure 3.2: SARROTO working flowchart.

In Fig. 3.1 is represented a block diagram which shows SARROTO general architecture. Basically the general program execution can be described as follows:

1. Some initial variables have to be introduced as inputs. These ones are the initial state vector, defined in keplerian coordinates and the related starting epoch. The optimiser has to be picked between the available ones.

2. Then the target state vector has to be selected. This selection can be made choosing, in a list of elements, the desired combination of variables that the client would like to obtain as output. As previously mentioned, weight and tolerance values are associated to each element.

3. The third step concerns the parameters choice. With parameters on refers to variable which can be optimised. Actually *parameters* refers to four quantities: three ΔV increment of velocity components, respectively radial, along track and cross-track, with the addiction of the manoeuvre epoch value. This last corresponds to time when manoeuvre will be performed. These four components are part of manoeuvre data vector: each manoeuvre has to be defined through these four variables. In appendix section B readers can found orbital manoeuvre description.
   The number of manoeuvre events has to be introduced as input. Different optimization options can be selected.
   Users can select through the interface panel mentioned in section 1.2, the number and type of abovementioned parameters which have to be kept fixed or to be optimised. For each optimising parameter selected, user has to introduce the searching interval in which the value to be optimised has to be looked for. The searching interval will be defined by means of lower and upper bounds. When fixed option is chosen, a specific value for the considered parameter has to be given.
   Furthermore, it has been introduced the option of optimising the manoeuvres number. In this case the maximum number of allowed manoeuvres during the propagation has to be introduced. The optimised result will be found computing propagation with number of manoeuvre in ascending order, until getting to the N-maximum allowed number previously defined. Manoeuvre epoch is forced to be one of the parameter to be optimised. Increment of velocity components can be optimised or kept fixed. The searching interval will be the same for all the tested manoeuvre both for epoch and ΔV.

4. Made these first choices, next step is defining the propagators options. As anticipated, two are the propagators introduced at this point in SARROTO: SAPO, a semi-analytic propagator developed during past FDO project and GOLEM, a completely analytic propagator, subsequently updated with some numerical features
   Their options can be different depending on the selected propagator.

   – In GOLEM time step for propagation and physical coefficient which consider solar radiation perturbation effects have to be introduced.
   – In SAPO in addition to the ones mentioned in GOLEM, various perturbations options can be selected.

5. Once these propagators options have been selected it is possible to choose the optimiser. Due to the early phase of development of this project, just two optimizers have been introduced:
   - Differential Evolution Optimiser, used in previous FDO project. It has been developed in 1995 by two mathematicians, as mentioned in [8]. Its complete explanation can be found in appendix section D and in section 3.1.2.1.
   - Parametric Analysis Optimiser, developed during internship. Its working method is illustrated in dedicated section 3.1.2.2.
6. The FORTRAN code is developed in such a way that at the end of the propagation and optimisation phases, the best solution will be shown in the focussuite output window. In particular the expected results are the final state vector obtained with the best optimised parameters solution and the optimum manoeuvres data.

In Fig. 3.2 it is represented the diagram which shows how the programs works.
In next paragraphs it will be explained the propagation and optimisation phases once that inputs have been defined.

3.1.1 Propagation Method

Input parameters needed for orbit propagation are different according on the selected propagator. Common input variables for both the two above mentioned propagators are the initial state vector with the related epoch and the target epoch. Manoeuvre list may be introduced if required. In addition to these parameters, each propagator can be configured with its specific option.

3.1.1.1 GOLEM Propagation Process

In addition to the previous mentioned variables, GOLEM needs some specific parameters to be defined. These parameters are the time step for moving inside the propagation time interval and the coefficient which describes perturbation due to solar radiation pressure. The propagation works in this way:

- First of all, number of steps between initial and final epoch are calculated considering the time step defined as input.
- Propagation starts taking into account initial variables defined as inputs. These variables are the initial state vector and the target epoch. Since the initial state vector is introduced as input, only some orbital elements value has to be calculated at first step. In particular elements' rate of change due to perturbations has to be evaluated.
- From second steps on, all parameters defined in state vector and the ones necessary to propagation, such as eccentric anomaly, mean anomaly and others, have to be calculated. The calculation of element at considered epoch makes usage of element value at previous step and its related rate of change.

- Some variables, such as Sun Ephemeris values, Greenwich Mean Sidereal Time, are calculated thanks to functions just existed in **focussuite** library. Their computation is done simply calling these preexisting functions.
- It has to be said that it is assumed that there will not be manoeuvre at first propagation (i.e this requirement is obtained through Fortran code flags).

As mentioned in section 2.1.2, two versions of this propagator have been developed. Readers can found their explanation in following paragraphs.

3.1.1.1.1 Initial Version At first GOLEM had been born as a geosynchronous completely analytic propagator. Its propagation worked in such a way that it began at initial epoch, defined in the main option ***focussuite*** panel, and finished at target epoch, introduced as input. In this first representation just one propagation for epoch interval took place: manoeuvre effects were considered inside equations, as explained in sec. 2.1.2. This manoeuvre effect can be synthesised simply considering the ΔV components, actually just the along track component, while other components will be skipped (i.e. obtained adding *flag rule* inside the code).
A timeline representation of what happens during GOLEM orbit propagation with manoeuvre events can be found in Fig. 3.3.

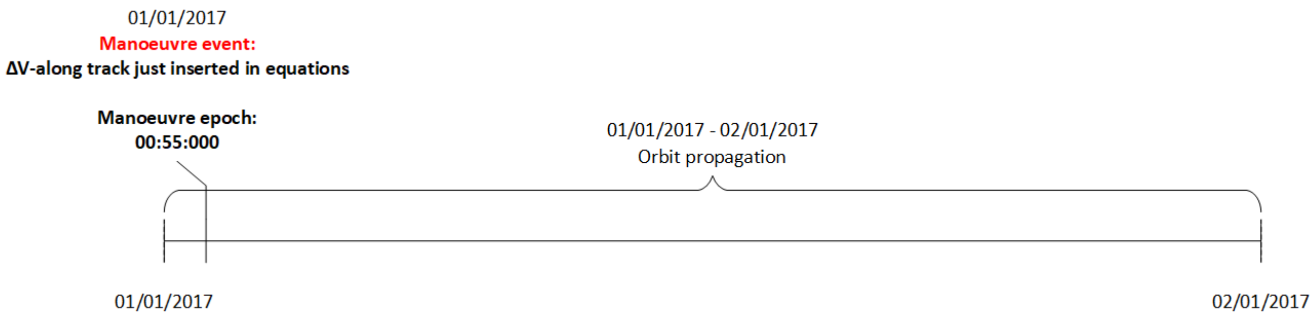

Figure 3.3: Timeline orbit propagation with manoeuvre for first GOLEM version.

3.1.1.1.2 Ultimately propagator Differently from the first version of GOLEM propagator, this last developed one can not be more considered completely analytic. This difference is due to the new way of considering manoeuvre event inside orbit propagation. The manoeuvres are no more taken in account analytically. In this ultimate version the ΔV components are fully considered, and their contribution is added to the state vector reached at manoeuvre epoch once converted in cartesian coordinate system. Please remind that initial state vector as to be introduced in keplerian element, as mentioned in section 3.1
For this reason there is not just one propagation, but the number of propagated orbit depends on the number of manoeuvres. Their number has to be introduced as inputs, both for fixed selection of optimised case.
Propagation starts at initial epoch and ends at first manoeuvre epoch, that can be known or calculated from optimisation. Then once that this first final state vector has been obtained, all

three velocity components are converted into cartesian coordinate system and added to the state vector velocity components obtained from propagation. Propagation follows until next epoch manoeuvre, or in case there are not, until target state vector epoch.
A simply timeline representation of propagated tool can be seen in Fig. 3.4

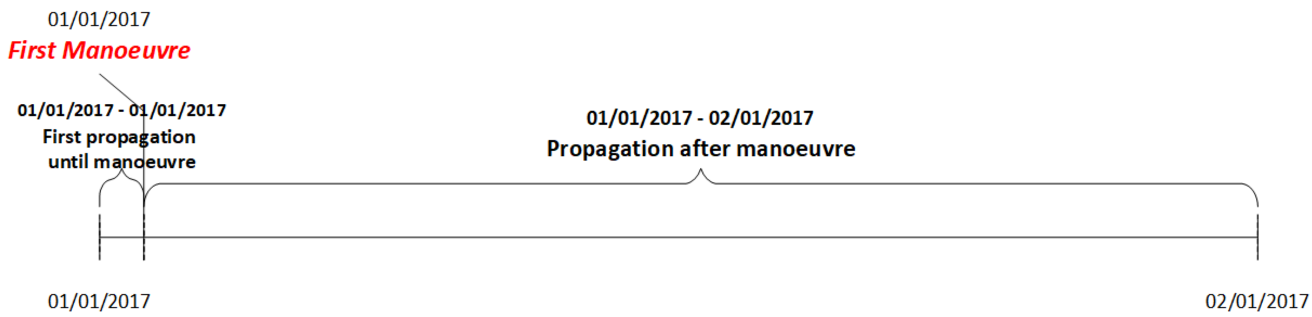

Figure 3.4: Timeline orbit propagation with manoeuvre for second GOLEM version

3.1.1.2 SAPO Propagation Process

SAPO, one of the GMV's semi-analytical propagator, is a software developed since 2014. The same features listed for GOLEM propagators are used for this propagator. Its main characteristics can be summarised in the list present below:

- Structured in modules which in their turn are divided into subroutines or functions, purposely developed for SAPO or belonging to other software libraries.
- Fully written in FORTRAN 90.
- High flexibility which allows the software itself to be adapted to other programs, such as it happens for SARROTO software.

SAPO working steps can be synthesised as follows

- Initialisation of variables and parameters needing for computation, coming up from client election, defined in the ***focussuite*** panel.
- Integration is performed considering presence of manoeuvre and orbital perturbations.
- Output results processed according with selected panels options.

A flow diagram showing how SAPO works can be found in Fig. 3.5.

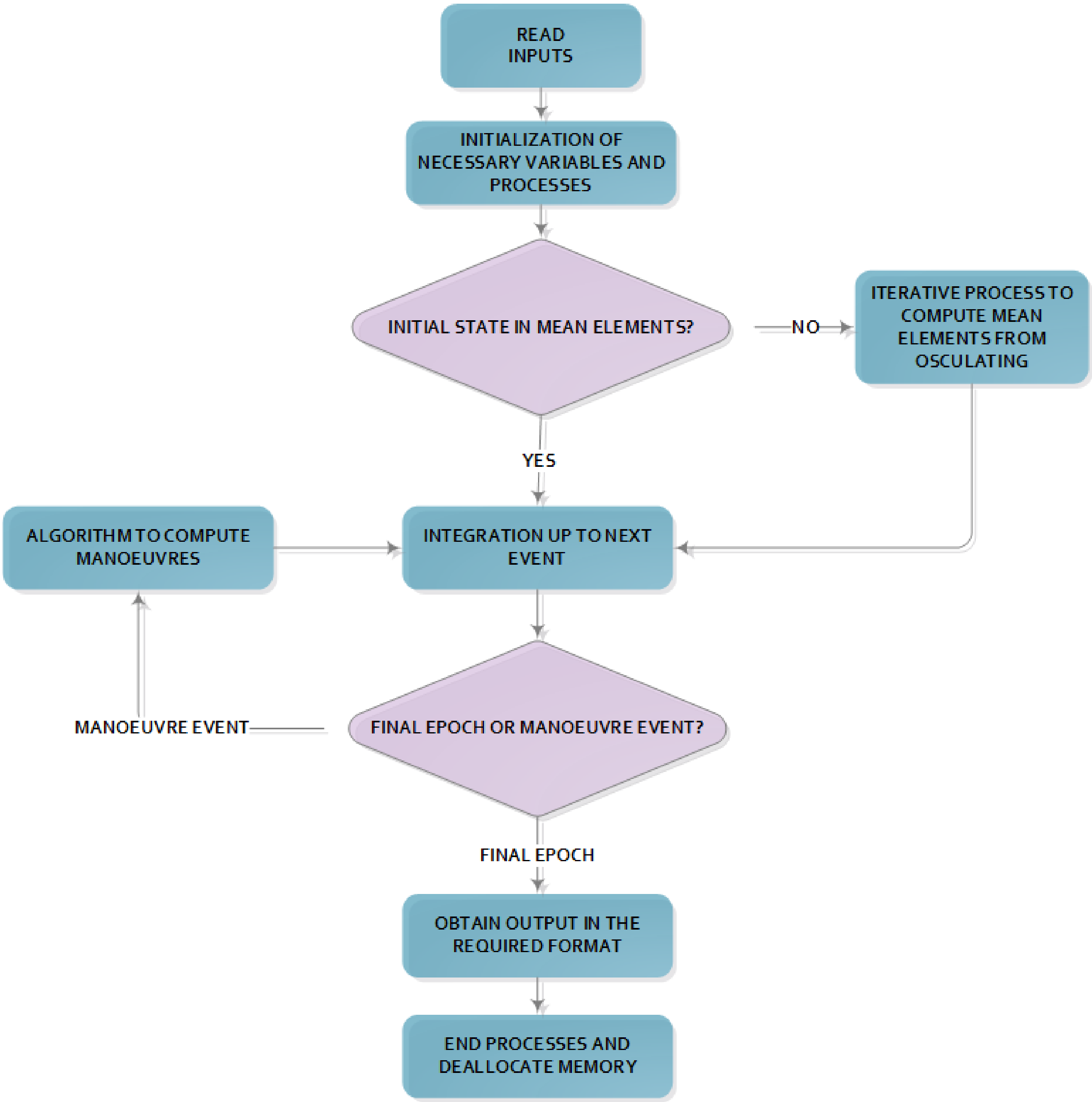

Figure 3.5: Flow diagram illustrating SAPO propagator at works.

As said before, SAPO, in the same way of GOLEM, has a modular structure, especially consisting in placing lower modules at the bottom of the architecture design, allowing to avoid problems like circular dependencies, which can generate failures in compiling Fortran. Relations between modules are illustrated in Fig. 3.6.

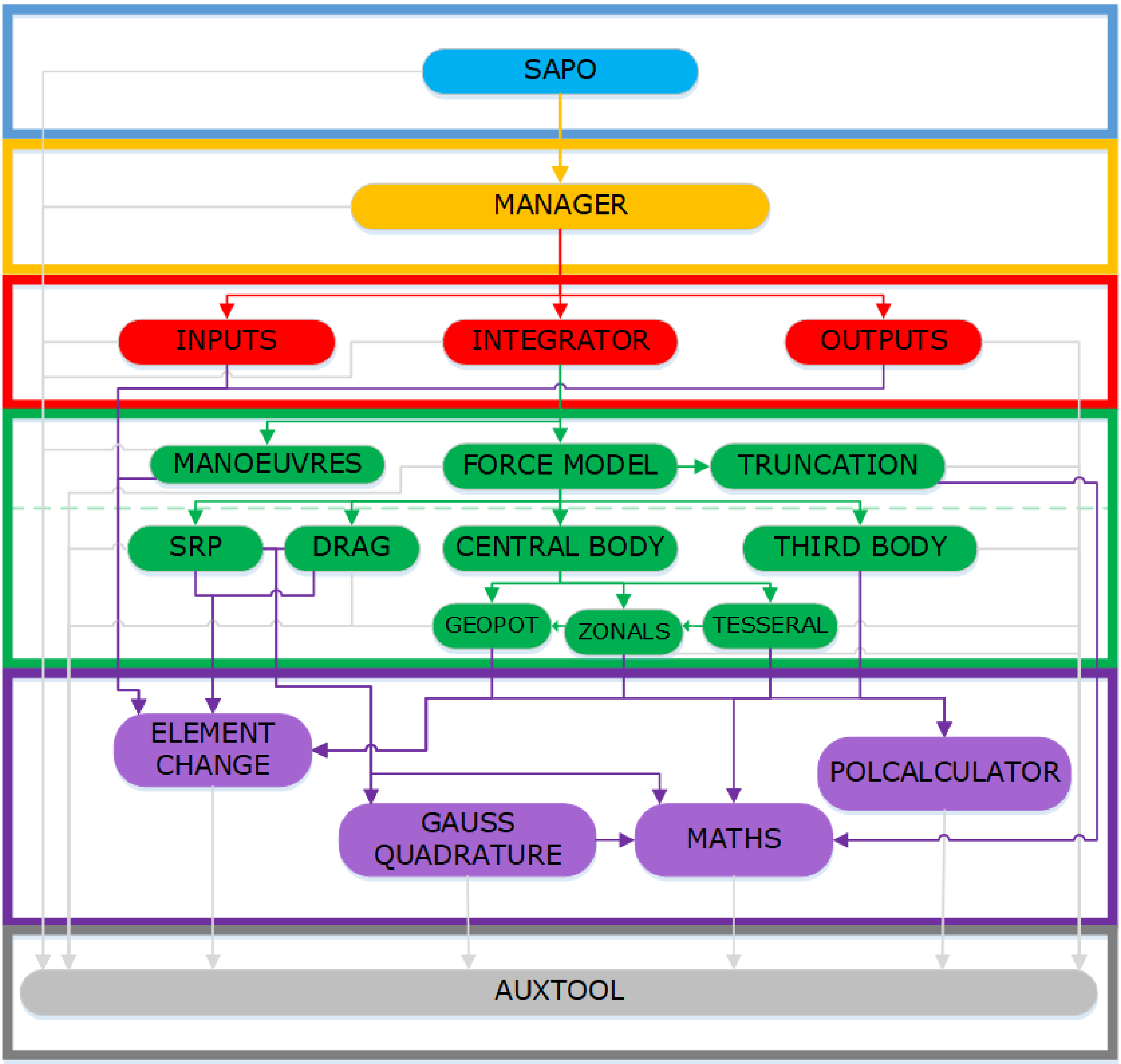

Figure 3.6: Architecture division of SAPO propagator.

To this general SAPO configuration, some changes have been made. In fact this propagator was born to be a single software, to be executed alone. Thanks to the flexibility obtained with its design, it has been possible to introduce this propagator into SARROTO program.

The SARROTO call to SAPO software happens thanks to the introduction of a new module. This new module has been developed as general as possible, so as to be introduced in the existed and future works, such as it happens with SARROTO. Due to its purpose this new module has been called **Interface**. It allows the main program to use the propagator without destroying its initial configuration. A flow diagram showing how this call works is represented in Fig. 3.7.

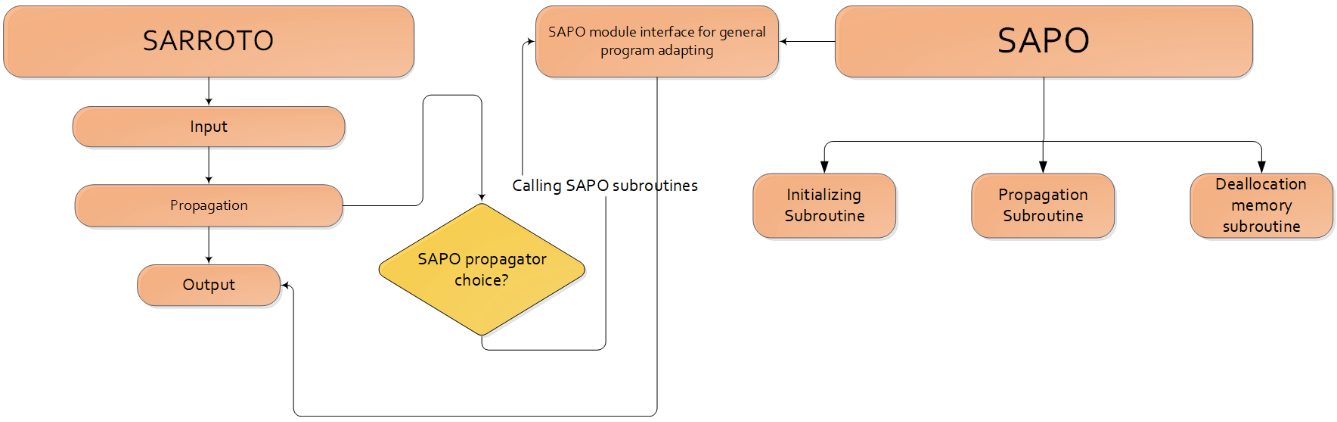

Figure 3.7: SAPO's call from SARROTO main program.

As said previously, other modules have been added in this internship to first SAPO version, due to the necessity to adapt this propagator to SARROTO, and to other general programs when needing. On the other hand the usage of some modules in SAPO have been hidden. In particular modules related to state vector election and manoeuvres handling have been overtaken. Indeed in the standalone use of this propagator these last ones have their own select data panels. These information are collected through the SARROTO main option panel.
So in next rows all modules will be explained, underlining the ones which are hidden because overlapped with the ones relevant for client first options election.

- **SAPO**: general module which is responsible for the manager module call
- **Manager**: it contains the subroutine and modules related to input data collection and dedicated to the integration phase; the ones related to collect data are overlapped and substituted with SARROTO ones
- **Integrator**: in this modules numerical integration of differential equations, presented in Eq. 3.1, are performed. These modules are not subject of study during this internship and for this reason there will be not fully explained. It has to be said that SAPO, differently from GOLEM, works with equinoctial coordinates, presented in appendix section A.2.3. For this reason, even if input coordinates system can be selected, a transformation will always be necessary to pass from one coordinate type to other. Different types of integrator can be used, more specifically these algorithms of integration which can be implemented

are Runge-Kutta 4 (fixed step size), and Dormand-Prince 45 and 78 (Variable Step Size).

$$\frac{da_i}{dt} = n(a)\delta_{i6} + \sum_{\alpha} A_{i,\alpha}(a,h,k,p,q,t) \tag{3.1}$$

- **Inputs**: modules related to data acquisition; initial data overlapped with SARROTO ones, but information about the satellite whose orbit has to be propagated can be found in this section.
- **Outputs**: these modules are hidden. At the end of SARROTO execution just information about optimised manoeuvre data and state vector will be displayed.
- **Force Model, Truncation**: the first module manages forces, the second one is about the limit of expansion according to tolerance defined by user. This last one is a numerical technique introduced in SAPO in order to increase propagator efficiency. It has not been used within this work.
- **Manoeuvres**: in this module all the function related to manoeuvre information are performed. Manoeuvres information is an input for SAPO. Users can pick between considering these ones as impulsive or long.
- **SRP, Drag, Central Body, Third Body**: general modules necessary to compute mathematical accelerations for obtaining data about all perturbations types, such as solar pressure, drag, central body, third body.
- **PolCalculator, Element Change, Gauss Quadrature, Maths**: module containing generic mathematical function.
- **Auxtool**: this is part of GMV generic modules and subroutine, there is a wide selection of functions, from transformation coordinates system, to manoeuvre handling and others.

3.1.1.2.1 Manoeuvre implementation Both impulsive or long manoeuvre can be implemented in SAPO propagator. A clear representation on the timeline of what happens during propagation when long manoeuvre is performed can be seen in Fig. 3.8. The manoeuvre is introduced in the general equation in the following way

$$\frac{da}{dt} = \vec{F}_{Kepler} + \vec{F}_{Gauss} + \vec{F}_{Lagrange} + \vec{F}_{manouvre} \tag{3.2}$$

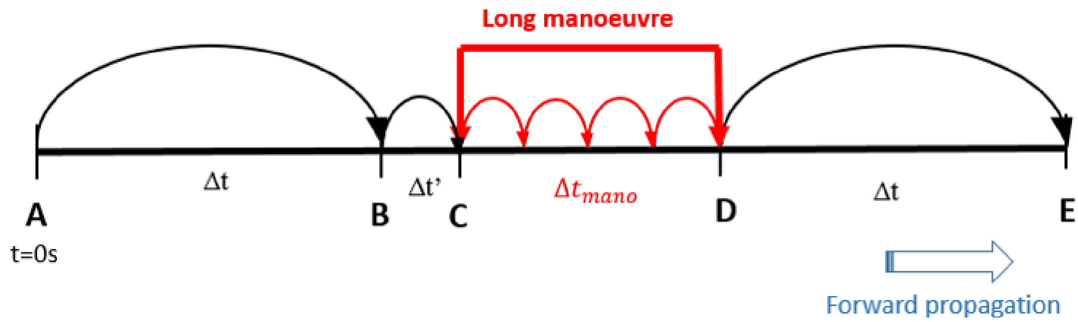

Figure 3.8: SAPO long manoeuvre effect on timeline propagation.

Different approach depending on the impulsive or long manoeuvre has been mentioned. In case of impulsive manoeuvre the mathematical model which is considered is the one of an infinite force which is applied during an infinitely short time. In this case there are two possible approaches. In the first one it is necessary to compute transformation in Cartesian coordinate system. At the same way discussed for GOLEM manoeuvre effects in section 2.1.2 and 3.1.1.1, the ΔV has to be added to the state vector obtained at this epoch

$$\frac{dx_{next}}{dt} = \frac{dx_{prev}}{dt} + V_{components} \tag{3.3}$$

A representation of what happens is found in Fig.3.9.

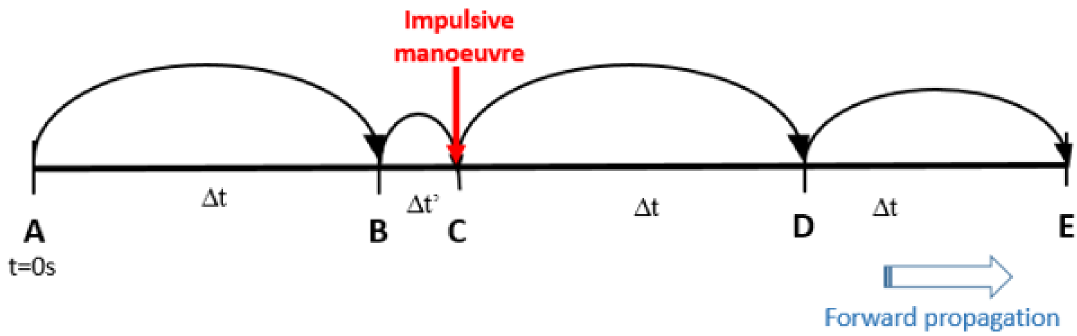

Figure 3.9: SAPO impulsive manoeuvre effect on timeline propagation: first approach.

The second approach is performed considering the same equation mentioned for the long manoeuvre through Eq. 3.2, but simply considering a short duration of this manoeuvre, as it can be seen in Fig. 3.10.

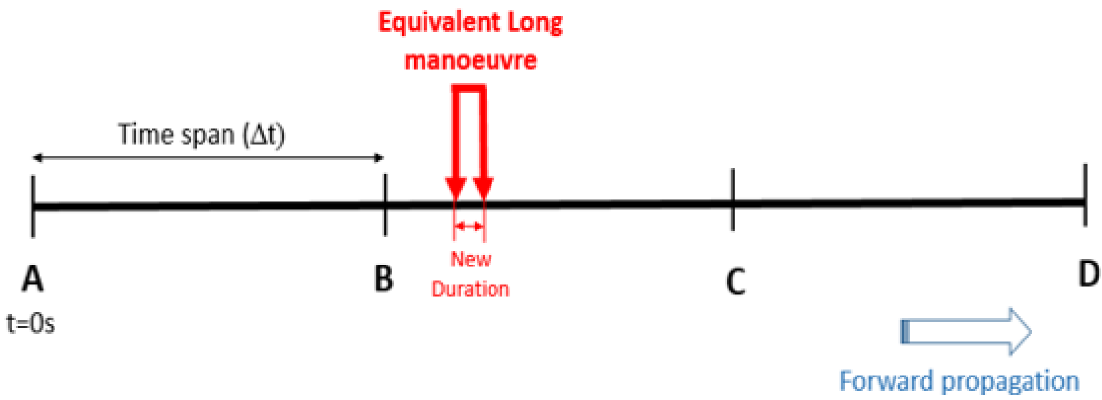

Figure 3.10: SAPO impulsive manoeuvre effect on timeline propagation: second approach.

3.1.1.3 SAPO vs GOLEM

Both propagators have been used to perform different analysis cases. Differences between the two propagators are summarised in next lines.
Due to the quasi-analytic definition of GOLEM, it can be considered just a first method to investigate a propagation. GOLEM, as its name suggests, can be used only for geosynchronous orbits or orbits near to this one. Differently from SAPO, in GOLEM propagator not all perturbations are considered and for this reason it can not be executed when accurate results would like to be obtained. In addiction, some orbit cases, with singularities due to Keplerian coordinates usage into equations, can not be performed.
Instead, focusing on SAPO perturbation, Equinoctial coordinates have to be used because they represent the best way to describing perturbations effects and are not affected by orbit singularities. More details about these elements can be found in appendix section A.2.3. Indeed the presence of forces are the reason why movement can not be considered purely Keplerian. In the true semi-analytic theory perturbations effects have to be taken into account. These ones are the same mentioned in SAPO modules section related to *Perturbations*. Not all the perturbations have the same importance, which means that their magnitude effects depends on the proximity to the Earth, more specifically on the height of the ground, i.e. aerodynamic resistance affects more when closer to ground, on the other hand solar radiation pressure is more significant when the height of the orbit is greater.
Total list of perturbations is presented below. Once all perturbations have been presented separate list for the ones included in SAPO and GOLEM will be drawn up.

- Not-spherical representation of Earth, due to its oblateness caused by rotation around the polar axis. The effect of this perturbation is described through the zonal harmonics coefficients for the Earth's gravitational field.
- Forces caused by the celestial body attraction which act on the satellite, particularly powerful are the ones cause by Sun and Moon, these ones become more important when distance from Earth increases.

- Atmospheric force, which among all, represents the most important non-gravitational perturbations disturbing low altitude satellites. This force can not be defined precisely at all due to some problems in obtaining very precise value of density and wind in upper atmosphere, or considering the effects of neutral gas and charged particles which act on the diverse satellite surfaces. In geosynchronous orbits its contribution can be neglected since no appreciable atmosphere (i.e. gas molecules) is present.

- Solar radiation pressure perturbation. This force, which is caused by absorption and reflection of photons, is considered utilising the apposite numerical constant.

Among these ones, the ones allowed in GOLEM are:

- Solar radiation pressure. The numerical constant which describes this perturbation includes satellite's mass and area. It is introduced into equations as a single value.

- Non-spherical Earth. It is considered through the second zonal harmonic J_2 and through longitudinal acceleration which depends on J_{22}. These values are taken into account inside GOLEM analytic equations as explained in section 2.1.2.

On the other hand in SAPO the perturbation taken into account are:

- Solar radiation pressure.

- Non-spherical Earth. This effect is taken into account by means of zonal harmonics until the 36th degree and order.

- Sun and Moon gravity perturbations.

- Atmospheric drag.

Other difference between the two propagators is the way to consider the manoeuvre. In fact in GOLEM just the impulsive manoeuvre is executed. In case of long one, it will be converted to impulsive. When manoeuvre event happens velocity components are added to state vector at this epoch in order to obtain the new one after manoeuvre. On the other hand in SAPO, it is possible to select manoeuvre type, if impulsive or long, and the code is developed in different way depending on user selection.
Differently from SAPO, which has a wide number of options to be defined, GOLEM features to be selected just consist in the time interval step size and coefficient for solar radiation pressure perturbation.

3.1.2 Optimization Procedure

The general architecture of the two mentioned optimizers will be discussed in next sections.

3.1.2.1 Differential Analysis Optimiser

This module code have been introduced in GMV's FDO during past project, but got from external resource, as it can be appreciated in [19].
In this section a concise description of this optimizer will be given, but interested readers could find a detailed outline in the appendix section D.
The Differential evolution is an algorithm used in order to solve optimisation problems which concern continuous domains. Its main characteristics which make it interesting from the computational point of view are its robustness, speed in obtain good results and its ease way of use. Its methodology is based on a research performed among NP population variables of **D** dimensional parameter vectors for each generation. This research is computed in order to find the best solution. It implements mathematical formulation in order to generate new solution at each analysed iteration.
Parameters which have to be considered in this optimizer are:

- Population size, which consists in the numbers of parameters vector for each generation among which searching the best vector. This value can be chosen randomly.
- Crossover factor is the constant values which defines the crossover step in the main procedure.
- Strategy is the implemented method. In this Differential evolution code six strategies have been implemented.
- Number of iterations corresponds to the generations number which have to be analysed in order to achieve the solution.

3.1.2.2 Parametric Analysis Optimiser

Differently from the previous, this optimizer has been completely developed and coding in Fortran during this internship. As its name suggests, the inherent module code works with the *parameters* variables which have been defined in previous section. Let recall that these variables are the ones which have to be optimised. The main features of this optimiser, is the ability to generate all the valid combinations of parameters and have them tested through the cost function to choose the one which best fits the desired solution.
The codification method has a purely logic aspect which means that there are absolutely not mathematical equations to consider. In fact logical steps which have been used to perform this optimiser are listed below

1. The first phase described in the module consists on acquiring as inputs the parameters which have to be used for parametric computation. These parameters are:
   - the number of parameters to be optimised, making distinction between epoch or general ΔV values
   - the lower and upper value of searching interval, which correspond to bounds within which the best optimised value is looked for

- the step size value necessary for moving inside the previously defined interval

2. Secondly, all combinations are computed considering each single value in the elected interval for all parameters, moving from the lower to the upper limit with the step size thickness

3. It is necessary to pay attention that, in case of manoeuvre epochs to optimise, it is not allowed to have two manoeuvres at the same time. In addition to that, if multiple manoeuvres are required to optimise, not only this first requirement has to be checked but also that the epochs have to be consecutive in time. It also has to be verified that ΔV module is not null for any manoeuvre, both the fixed and optimised ones, since it would have no sense speaking about a manoeuvre with zero ΔV value. If considered combination presents one of these features it will be skipped and next combination will be evaluated.

4. Once combination has been obtained and checked thanks to logical step, the parameter which has to be analysed is the cost function. In fact each combination vector is introduced as input to cost function calculation module, in which this value is calculated. This last one in turn is given as input to the parametric analysis module which evaluates if the one obtained with next combination is lower than the previous. If this statement is verified new cost function value is stored together with the related combination vector.

5. Once all combinations have been tested, the cost function value and the vector which contains the best optimised parameter is stored, in this way these values will be given as final result.

A graphical representation of combination process can be seen in box 3.4. On the left they are shown searching intervals for each parameter to be optimised. Total number of parameters to be optimised can go from 1 to N. In the represented case it is assumed to have three elements to be optimised. These last are indicated with *x, y, z*. The searching interval is delimited by the lower and upper values given as input. Number of elements inside each vector depends on the numerical step size given as input in order to move along the interval. On the right they are represented the combinations obtained after PA execution.

$$\begin{array}{lllllll} & Lower\ ..\ Upper & & & & \\ 1 & [\,x_1\ \ x_2\ \ x_3\,] & & [\,x_1\ \ y_1\ \ z_1\,] & [\,x_2\ \ y_1\ \ z_1\,] & [\,x_3\ \ y_1\ \ z_1\,] \\ 2 & [\,y_1\ \ y_2\,] & \Longrightarrow & [\,x_1\ \ y_1\ \ z_2\,] & [\,x_2\ \ y_1\ \ z_2\,] & [\,x_3\ \ y_1\ \ z_2\,] \\ .. & .. & & [\,x_1\ \ y_2\ \ z_1\,] & [\,x_2\ \ y_2\ \ z_1\,] & [\,x_3\ \ y_2\ \ z_1\,] \\ N & [z_1\ \ z_2] & & [\,x_1\ \ y_2\ \ z_2\,] & [\,x_2\ \ y_2\ \ z_2\,] & [\,x_3\ \ y_2\ \ z_2\,] \end{array} \tag{3.4}$$

In Fig. 3.11 it is represented the working method used to compute combinations vectors. Squares box on the left represent the searching interval for each parameter to be optimised. On the right combinations which result at the end of PA computation. In particular arrows which connect the box indicate how these combinations are made.

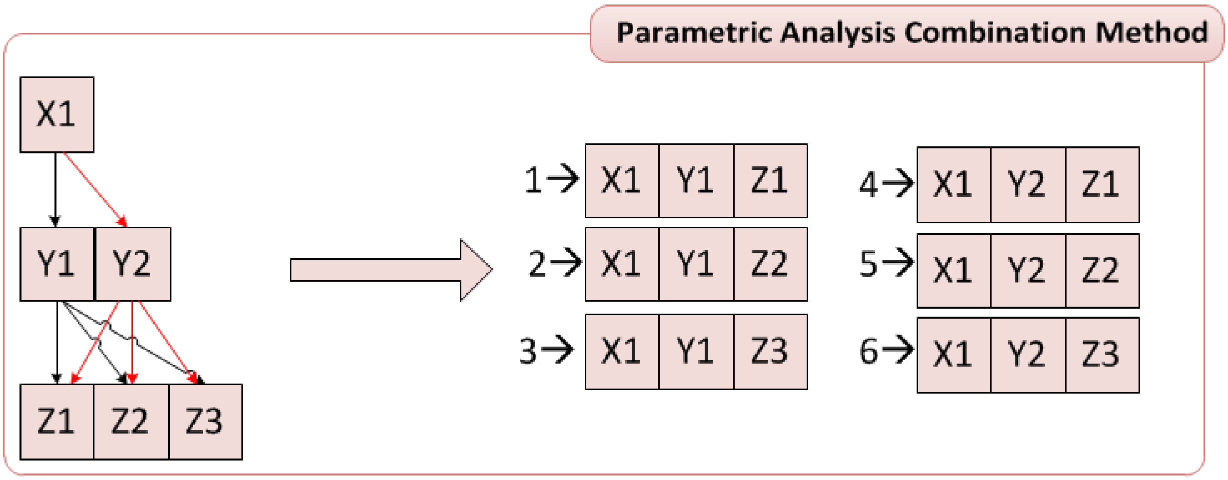

Figure 3.11: Parametric analysis combination method.

A flow diagram which represents what happens during this optimiser call can be found in Fig. 3.12.

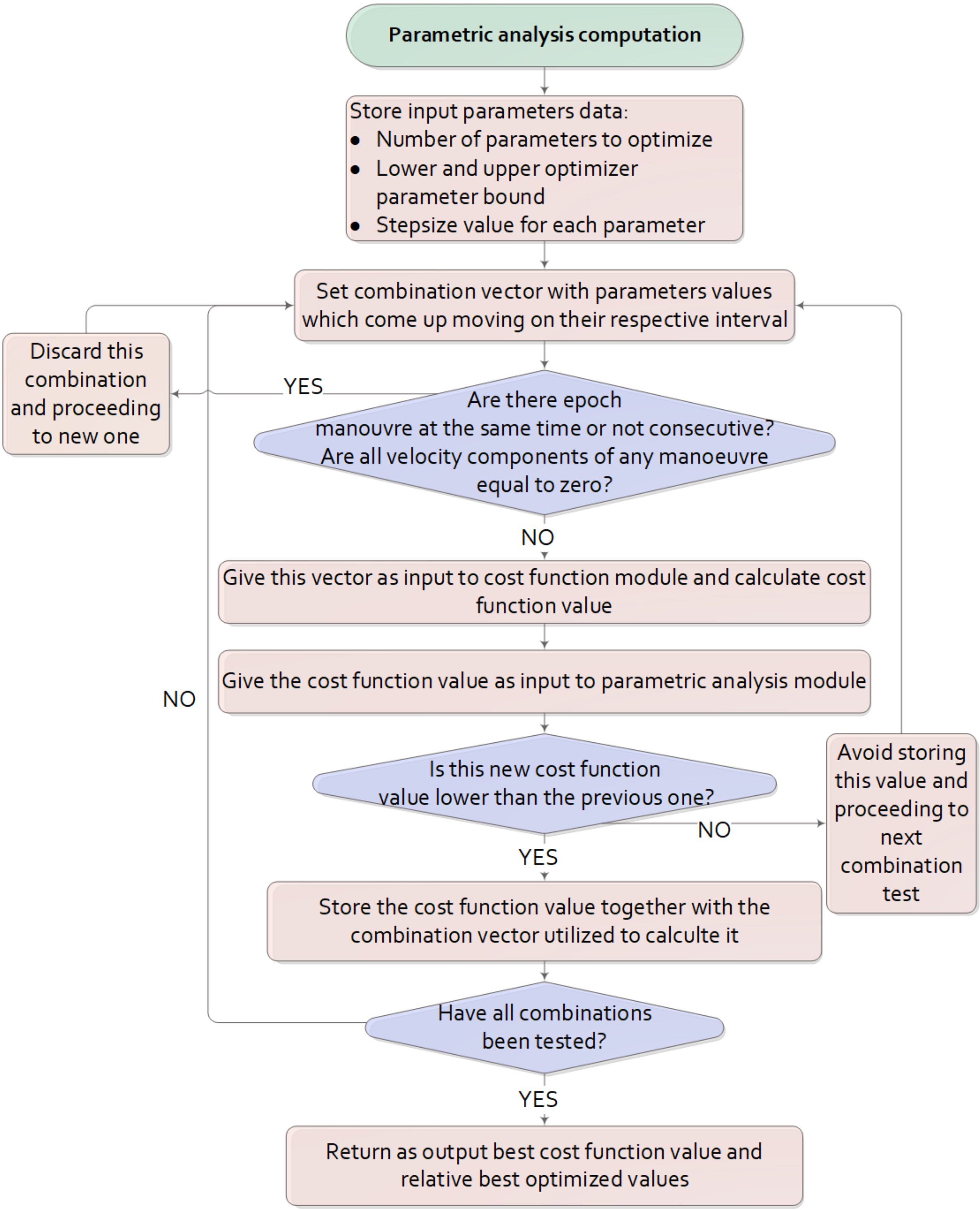

Figure 3.12: Parametric analysis working module flow diagram.

3.1.2.3 Differential Evolution vs Parametric Analysis

The Differential Evolution optimiser has a working method that is completely mathematical which means that utilises equations, and it has a statistical way of working. On the other hand the Parametric analysis one has a logical approach.
Between these two it is preferred the second one because more features can be put in practise. With features in this case on refers to the possibility to select the same time interval for searching the best optimised values, possibility which can not be considered for the differential analysis where the research interval has to be defined for each manoeuvre which user would like to optimise.
In addition, differently from the DE, in the parametric analysis it is possible to choose whether optimising the number of manoeuvres, and in this case selecting the research time interval which has to be the same for all the manoeuvres which client wants to test. In DE this option is not implemented since this this tool can work only with a predefined number of optimised parameter. On the other hand the big difference, which promotes the differential evolution is the time of computation for coming up with the best result. In fact the greatest disadvantage associated to the PA is the time which it takes to test all the possible combinations, since it has to create each combination and check if it is the best. In the chapter 4 related to result, computational type spent for DE and PA combination with both propagators will be discussed.

3.2 Program Interface

As mentioned in previous section, SARROTO is integrated in ***focussuite***, FDO tool for flight dynamics solution. This tool is the interface thanks to which it is possible to select the solution ones would like to obtain. In Fig. 3.13 it is possible to see a representation of the tool interface, as it is in reality.

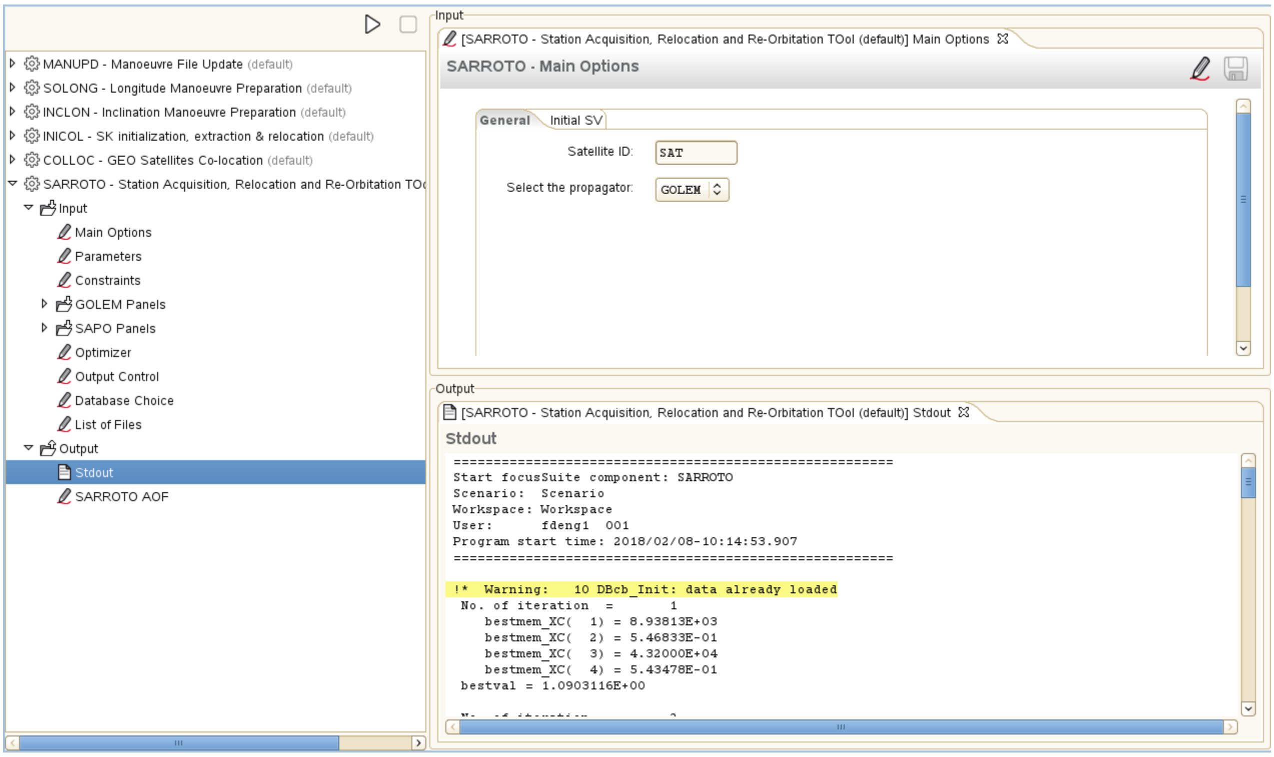

Figure 3.13: ***focussuite*** interface.

In Fig. 3.14 it is possible to see how SARROTO appears in focussuite tool. The main features which appear in the panel list are the following

- **Main Option**: input variables are introduced in this tab.
- **Parameters**: in this panel optimisation options will be present. The number and type of parameters to be optimised will be chosen at this point.
- **Constraints**: target state vector elements, their weight and tolerance values have to introduced here. Their choice is left up to users.
- **GOLEM Panels**: options related to GOLEM configuration.
- **SAPO Panels**: options related to SAPO propagator can be found in this tab.
- **Optimiser**: optimizer option are presented just for DE selection. PA does not need inputs option to be executed.
- **Output Control, Database Choice**: in these panels general option about output format can be found.
- **List of files**: collection of files where output information have to be written, has to be introduced here.

- **Stdout**: It represents the standard output panel, where results are shown at the end of computation.

As it can bee seen in Fig. 3.14 at the end of the list are present some requests related to output files which the client would like to obtain as result. In next paragraphs it will be illustrated each section specifically.

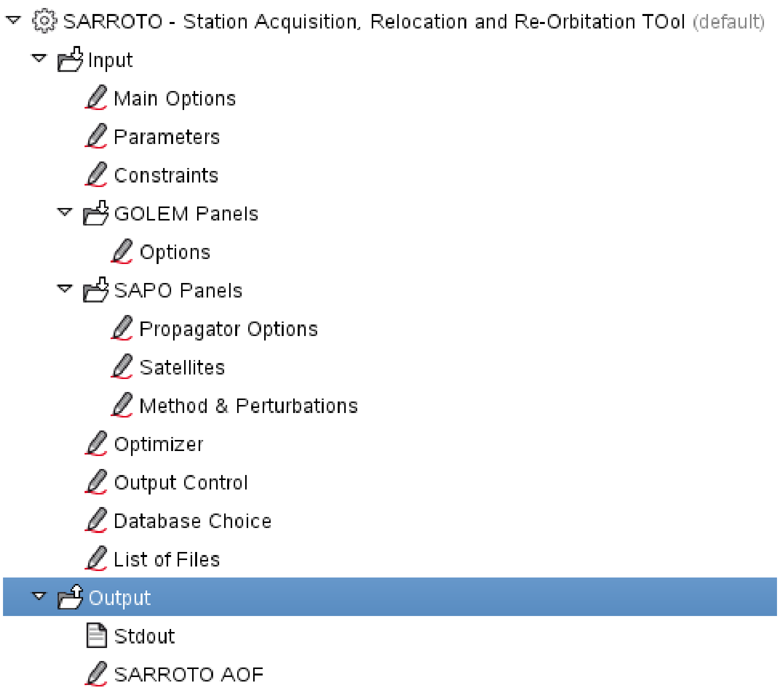

Figure 3.14: SARROTO working list in ***focussuite*** FDO tool.

3.2.1 Main Options Panel

In this tool section main options features are selected. As it can be seen in Fig. 3.15 and 3.16, fundamental inputs refers to this panel. In the first panel, referring on Fig. 3.15, satellite name, defined as Satellite ID, and propagator choice are inserted.

The Fig. 3.16 shows the input state vector panel. In fact state vector always is defined in classical orbit coordinates, the Keplerian ones, and as it can be seen, all the six components are present. In addition to these ones, at the top of the list, the epoch which indicates the begin of propagation is present. The second option, the one presents in the rectangle box, allows to choose between getting the state vector from panel, which means activating the followings parameters boxes, or reading and getting the state vector from out files, which will be introduced in the *List of File* option present in Fig. 3.14. Differently from what concerns the target state vector elements, the initial ones only can be introduced in keplerian coordinates. There is not a specific reason why it should be in this way, in fact as it will be illustrated in future work chapter 5, this is a proposed future improvement.

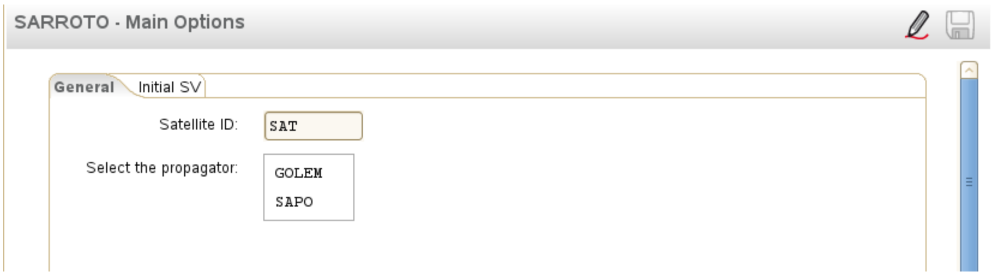

Figure 3.15: ***focussuite*** Main Option panel.

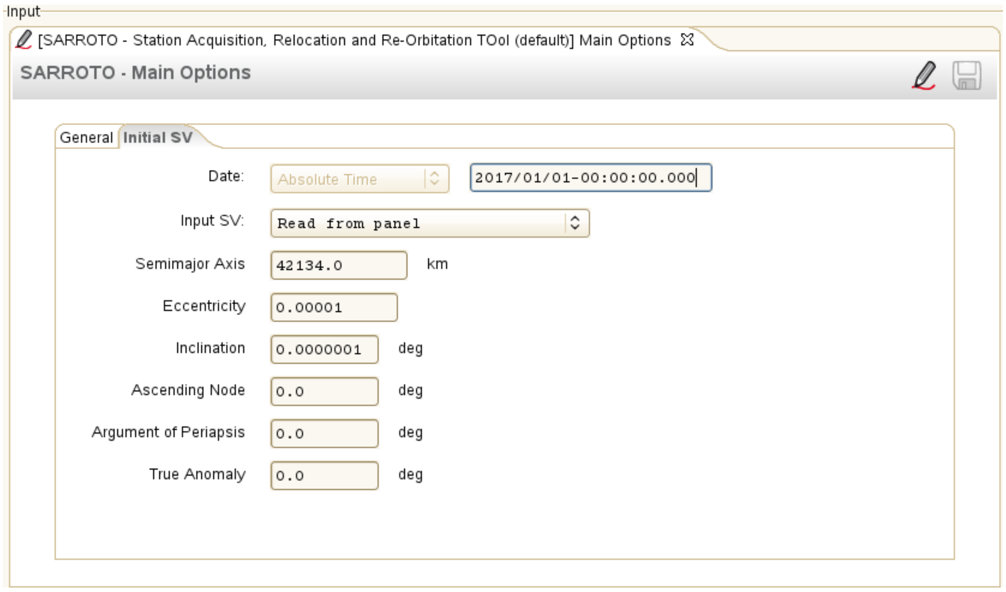

Figure 3.16: ***focussuite*** State Vector choice panel.

3.2.2 Parameters

With the word parameters one refers to variable which can be optimised, as explained in section 3.1. The general panel interface can be seen in Fig. 3.17. In this panel users have to pick between keeping the number of manoeuvre fixed or to be optimised. For each choice a different tab has to be considered.

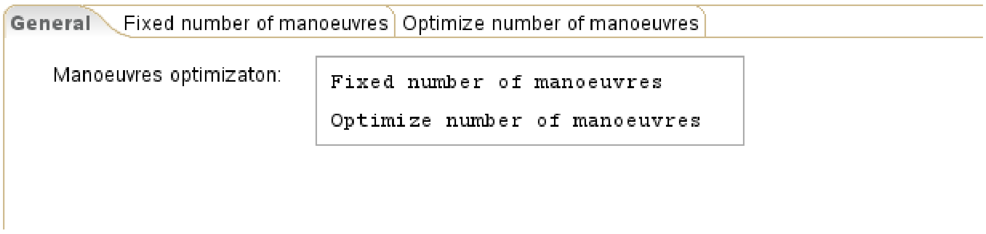

Figure 3.17: ***focussuite*** Parameters general panel.

- **Fixed number of manoeuvres**
  Once that fixed number of manoeuvres has been selected, all the rectangular boxes in this interface will activate. As the Fig. 3.18 shows, the first value which has to be inserted is the number of manoeuvre that client would like to perform during his satellite's propagation orbit. As it can be seen, there is a list of parameters which have to be introduced. For each manoeuvre the abovementioned four parameters have to be defined. It is possible to choose the optimisation method for the manoeuvre epoch: it can be introduced a common searching interval for all the manoeuvres, or each one can be associated to its own interval.

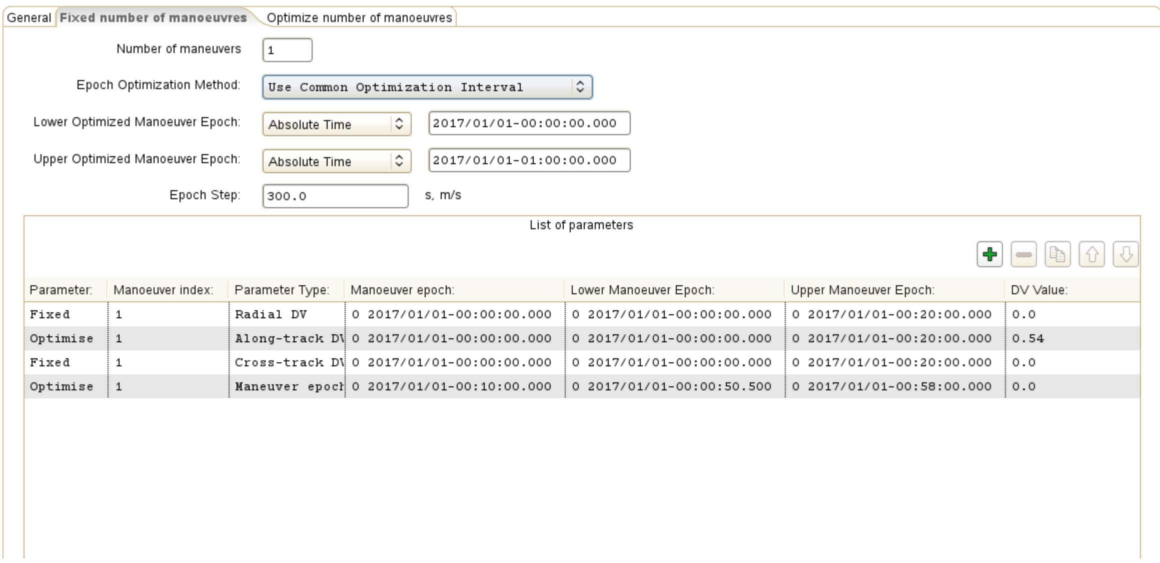

Figure 3.18: ***focussuite*** panel for fixed number of manoeuvre.

- **Optimized number of manoeuvres**
  In case of this selection has been made, the user has to define the maximum numbers of manoeuvres among which he would like to find the best solution (i.e. if the number of permitted manoeuvre is 3, the results could be 1, 2, or maximum three manoeuvre as best result).

A common searching interval for each parameter type is present, defined by lower and upper bound. Epoch must be optimised. On the other hand components of velocity may be optimised or kept fixed to a defined value. This configuration can be seen in Fig. 3.19.

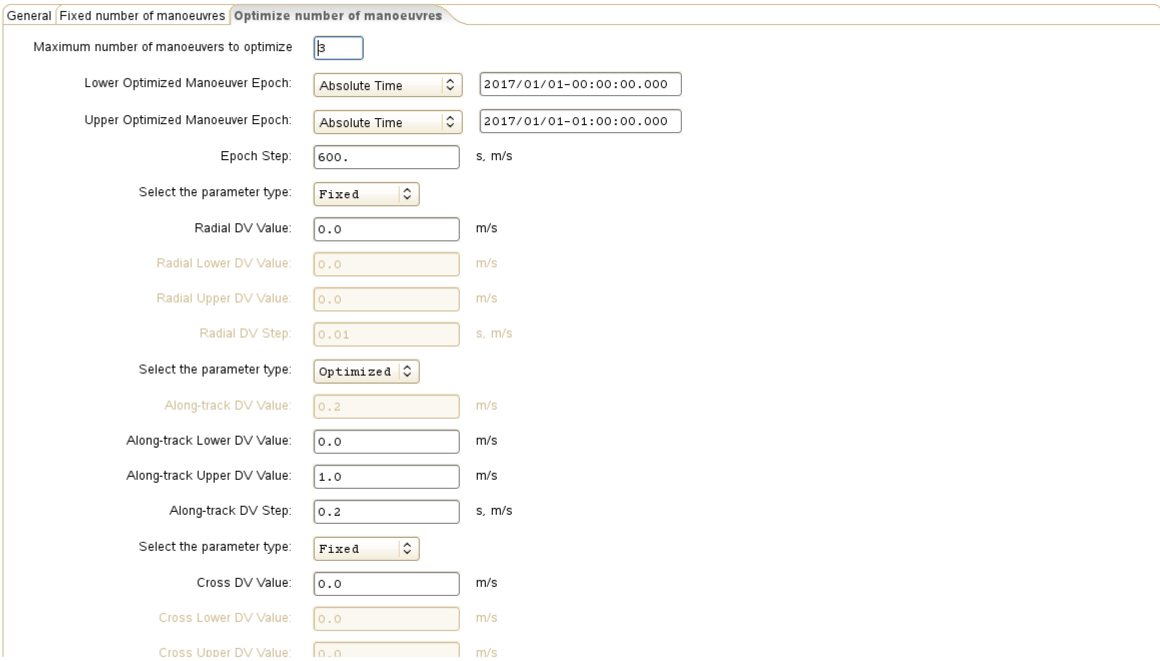

Figure 3.19: ***focussuite*** panel for optimised number of manoeuvre choice.

3.2.3 Constraints

Panels related to this input section refers to target state vector election and corresponding thresholds. As said in the general overview, it is possible to choose the combination of state vector parameters in a complete list. In this last all the elements coming from different coordinates system, in particular Keplerian, Cartesian, Geodetic and Equinoctial ones are present. In Fig. 3.20 it is possible to see how the panel appears in the ***focussuite*** interface. In this list it is possible to introduce N-numbers of parameters and successively select only the ones of interest. Please note that the executing program will come up with error message in case that redundant parameter is introduced.

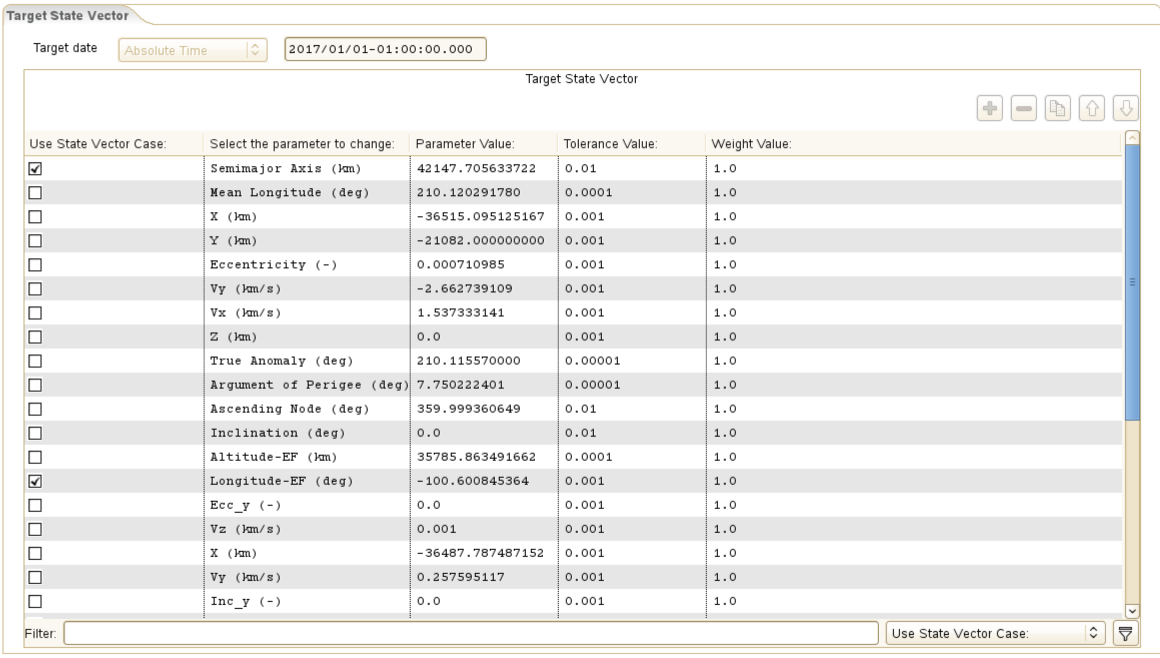

Figure 3.20: ***focussuite*** Constraints choice panel.

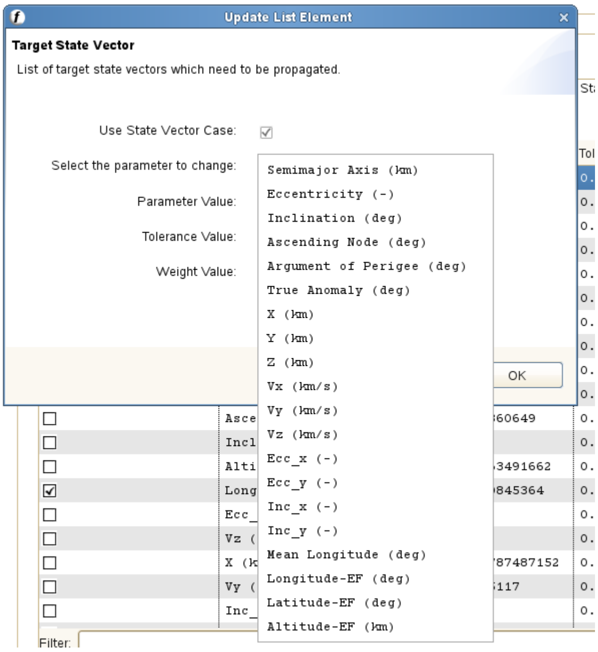

Figure 3.21: Target State Vector list of parameters.

3.2.4 Propagators

3.2.4.1 GOLEM Panels

In GOLEM panel the only two parameters which have to be introduced are the time step for moving along the epoch interval and the solar radiation pressure coefficient. As discussed in 3.1.1.3, this last coefficient already includes the satellite's mass and area contribution. These values have to be defined by user. In order to be executed this propagator must receive as inputs the initial state vector with related epoch and the target epoch. These information are introduced in panels related to Main option and Constraint as it has been possible to note in Fig. 3.16 and 3.20

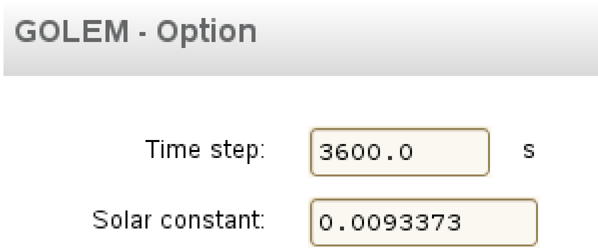

Figure 3.22: ***focussuite*** GOLEM propagator main option panel.

3.2.4.2 SAPO Panels

The SAPO version adapted to SARROTO presents a reduced number of panels with respect to the SAPO standalone tool. This because most of the inputs are introduced in the main program rather than in the propagator configuration. Fig.3.23 shows this focussuite panel for input data acquisition. On the other hand step size, as it happens with GOLEM, can be defined by client as represented in Fig. 3.24

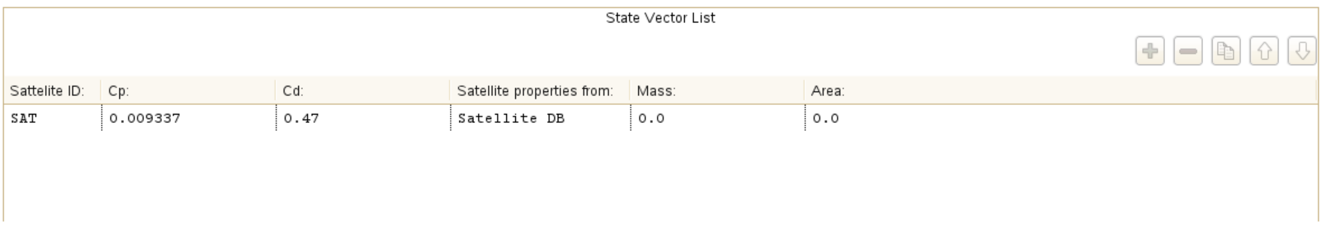

Figure 3.23: SARROTO's SAPO satellite election section panel.

Figure 3.24: SARROTO's SAPO main option section panel.

Manoeuvre data can be inserted in apposite space list as it is shown in Fig. 3.25. In particular as it is possible to notice the only input data which can be selected is whether considering manoeuvre as long or impulsive.

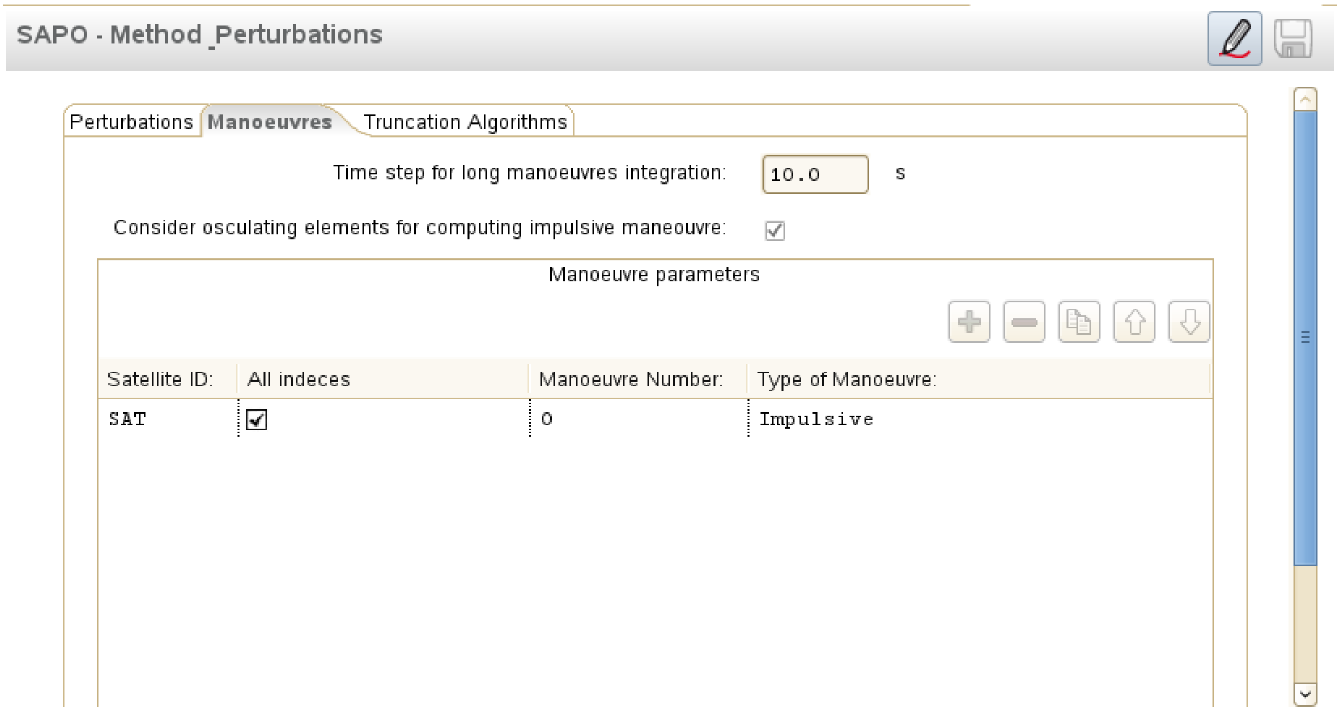

Figure 3.25: SARROTO's SAPO manouvre section panel.

Information about perturbation can be selected in panels which can be found in SARROTO's SAPO section, as it can be seen in Fig. 3.26.

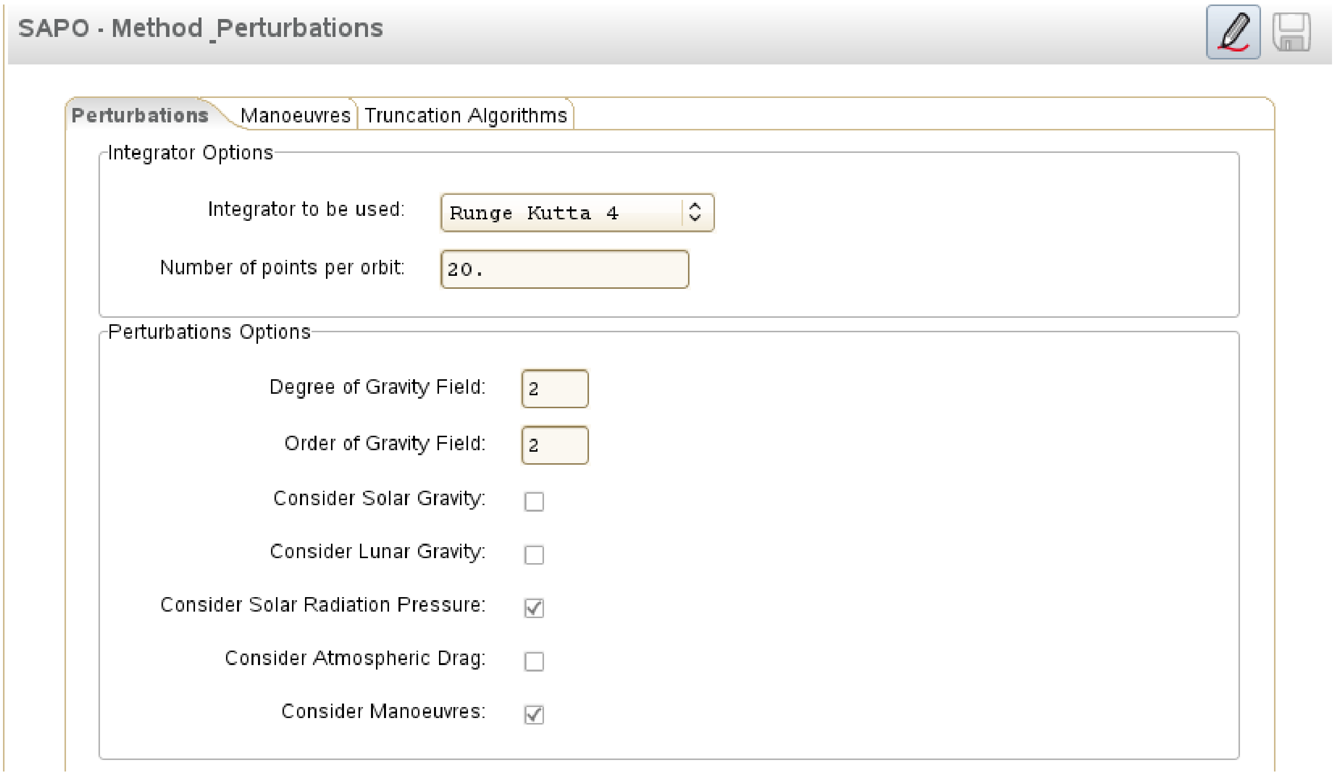

Figure 3.26: SARROTO's SAPO perturbation section panel.

3.2.5 Optimiser Interface

3.2.5.1 Differential Evolution Panels

The parameters defined in 3.1.2.1 have to be introduced in panels as input values. They can be defined by users according to requirements. In Fig. 3.27 the DE input panels is represented.

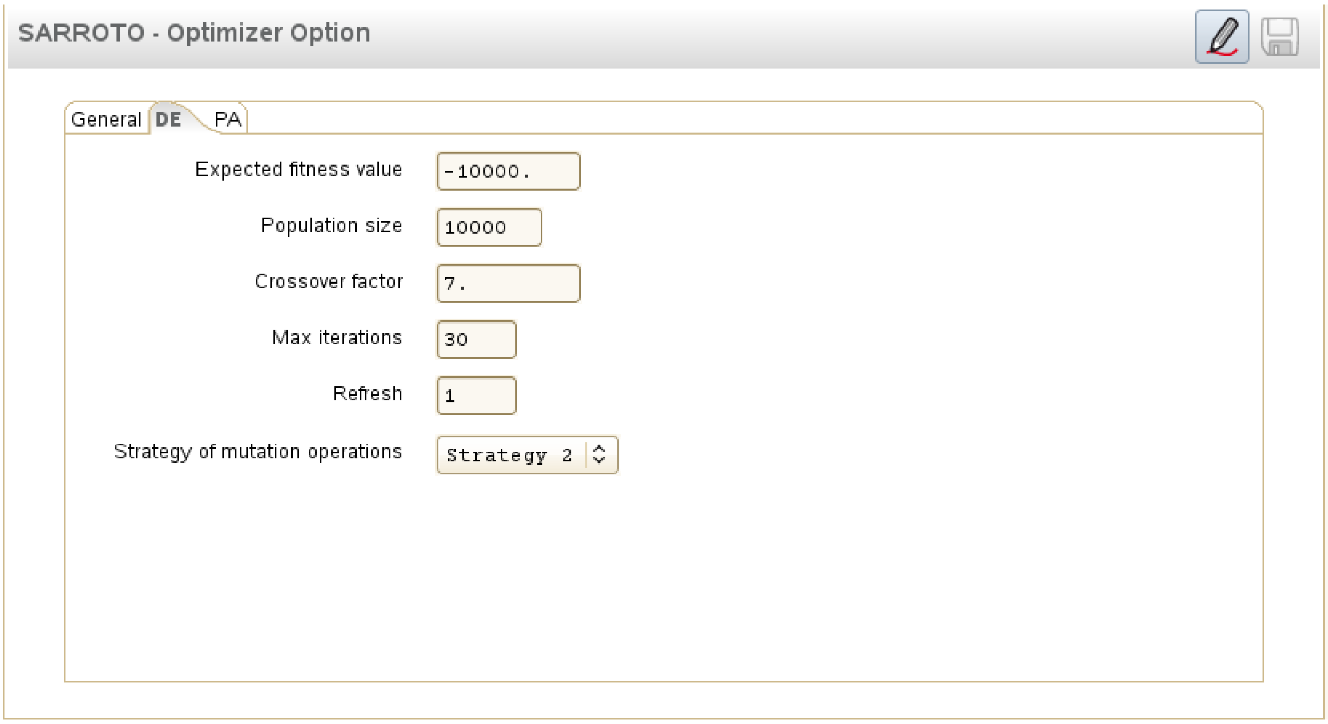

Figure 3.27: ***focussuite*** Differential Evolution main panel.

3.2.5.2 Parametric Analysis Panels

Differently from the Differential Evolution, Parametric Analysis does not need specific input values. As defined in section 3.1.2.2, it receives as input from the main program the number of parameters to be optimised and the related searching intervals.

3.2.6 Standard Output Panels

As already mentioned, this panel is dedicated to the output results representation. A classical format in printed output can be seen in Fig. 3.28. The results which appears in this ***focussuite*** window are the same which are written in the output file.

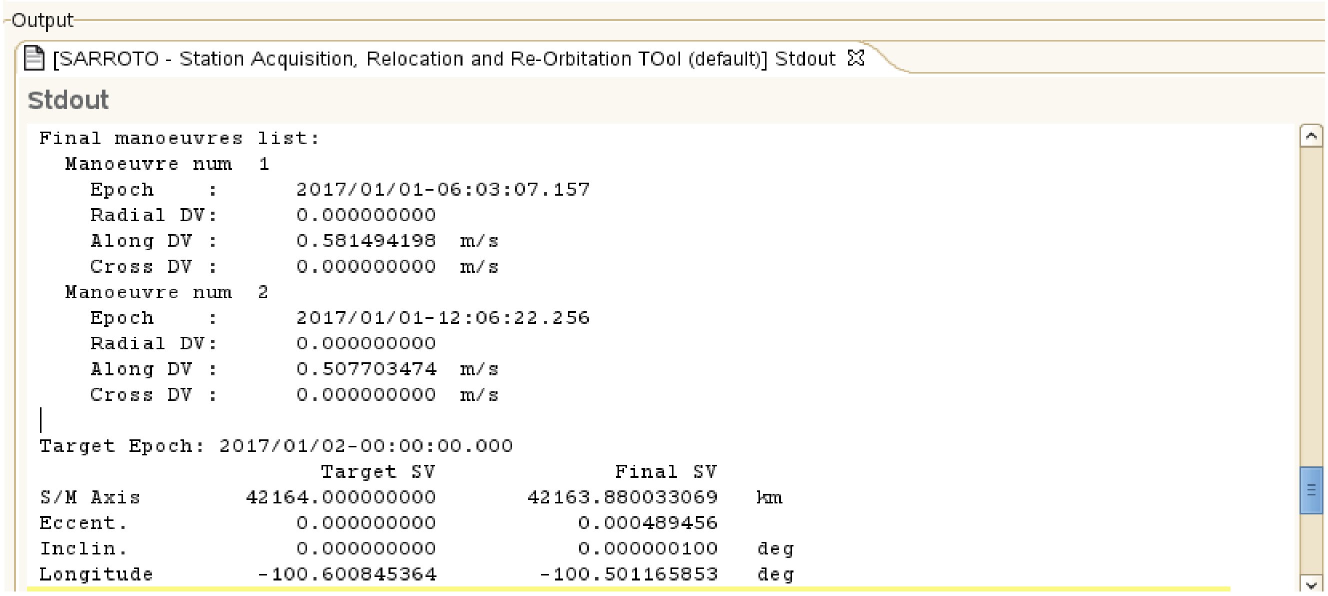

Figure 3.28: ***focussuite*** Standard Output panel.

3.3 Code development and evolution

In this section of the thesis it will be described the evolution of the project along the training. This work has been developed in different phases, many times getting back to previous steps in order to introduce new changes as they came up within the coding and implementing procedures. It has to be said that this project, from the organisational point of view, has combined two main subjects which are the Fortran 90 software codification and the mathematical research about the aerospace field of expertise. This means implementing the physical theories about space project with a computer language.

The two parts have not been considered separately. This is why during the development, further ideas have come up and improvements made. These last not only in the logical coding steps, but also in the physical theories applied. So these features have been overlapped during the entire work.

In adddition to what said above, the first two weeks have been completely devoted to literature review. A general overview has been carried out on the mathematical theories which would have been introduced. The main focus has been directed to the acquisition of the Fortran 90 rules of codification, the ones established by GMV's FDO department.

In particular, special attention has been dedicated to mathematical equations, took from [2], in order to implement the GOLEM propagator. These equations have been studied in greater details within steps of development. Once approached with the Fortran language, coding work has started. At first it has been necessary to give fundamental basis to the SARROTO structure, whose map has been illustrated at the beginning of this thesis and could be found in Fig.3.1.

It has meant creating the first program base structure which corresponds to the manager section of the program. In this module all the main components of the program, from the initial options to the final output file results, have been introduced. Once that this first part has been developed, especially general modules related to each single software routine, it has been possible to focus on these last.

In order to start doing some first tests, it has been created the GOLEM propagator. Its coding phase required a detailed study of the two body equation. This development phase has last roughly two weeks, including the necessary time for its implementation in the coding module. Meanwhile, it has been introduced the differential evolution optimiser whose Fortran code existed yet. As mentioned, it has not been developed in this project, but only adapted to SARROTO main structure. Once a propagator and an optimiser have been associated, it has been possible doing first tests proving the correct code structure. Parametric analysis optimiser has been developed in the last two summer weeks related to first training in GMV, and testing only at the beginning of the internship established with the University. So at the end of the summer training the entire structure had been developed, the DE optimiser implemented, GOLEM optimiser tested and the Parametric Analysis developed but not yet introduced.

With the beginning of the internship obtained with the agreement of the UPM, the project has followed its improvement. In particular the Parametric Analysis has been introduced and its functionality tested. Furthermore some new requirements have been added. One of these is the possibility of avoid combination parameter vector whose manoeuvres epoch overlap, or contain all velocity components equal to zero.
At the same time GOLEM has been improved, too. It has passed from being completely analytic to take a certain semi-analytic features such as the new way of considering the manoeuvre ΔV. This is no more contained in the main equations, but a transformation in Cartesian coordinates and an addiction to the state vector achieved at the considering epoch is required.
Changes in the target state vector have been made consequently. It has been added the possibility to choose among a complete list of parameters, involving most of the coordinates type, the desired target state vector. In this choice there is no limitation to the number of variable to introduce. The only requirement is avoiding the redundant choices (i.e it is not possible to have parameters of the same type).
From the parameters point of view other changes have been performed. It has been introduced the possibility to optimize the number of manoeuvres, defining their permitted maximum number in a time interval and imposing the same searching interval for all of them. Even when fixed manoeuvre number is considered, it has been added the possibility to use a common searching interval for all manoeuvres epoch.
Clearly considerable part of the time has been spent in solving coding problems, such as the ones related to memory leak, due to non deallocated variable defined in the Fortran code. This aspect can not be despised since these issues have caused a partial slowdown in the work activity.
Last step computed, one of the most significant ones, has been the introduction of SAPO propagator into SARROTO. It has taken roughly a month. Indeed not only a new module interface has been created in SARROTO in order to accommodate this propagator, but also some changes in SAPO have been required. From the architecture point of view some new modules have been

introduced in SAPO so as it will be adapted to any possible software which may require its usage.
At the end all tests have been performed, as it could be appreciate in next chapter 4 regarding to results.
It has to be mentioned that associated to the code development it has been performed the creation of the choices panel in the ***focussuite*** environment. Moreover during the main project development, the documentation, this thesis, has been written.

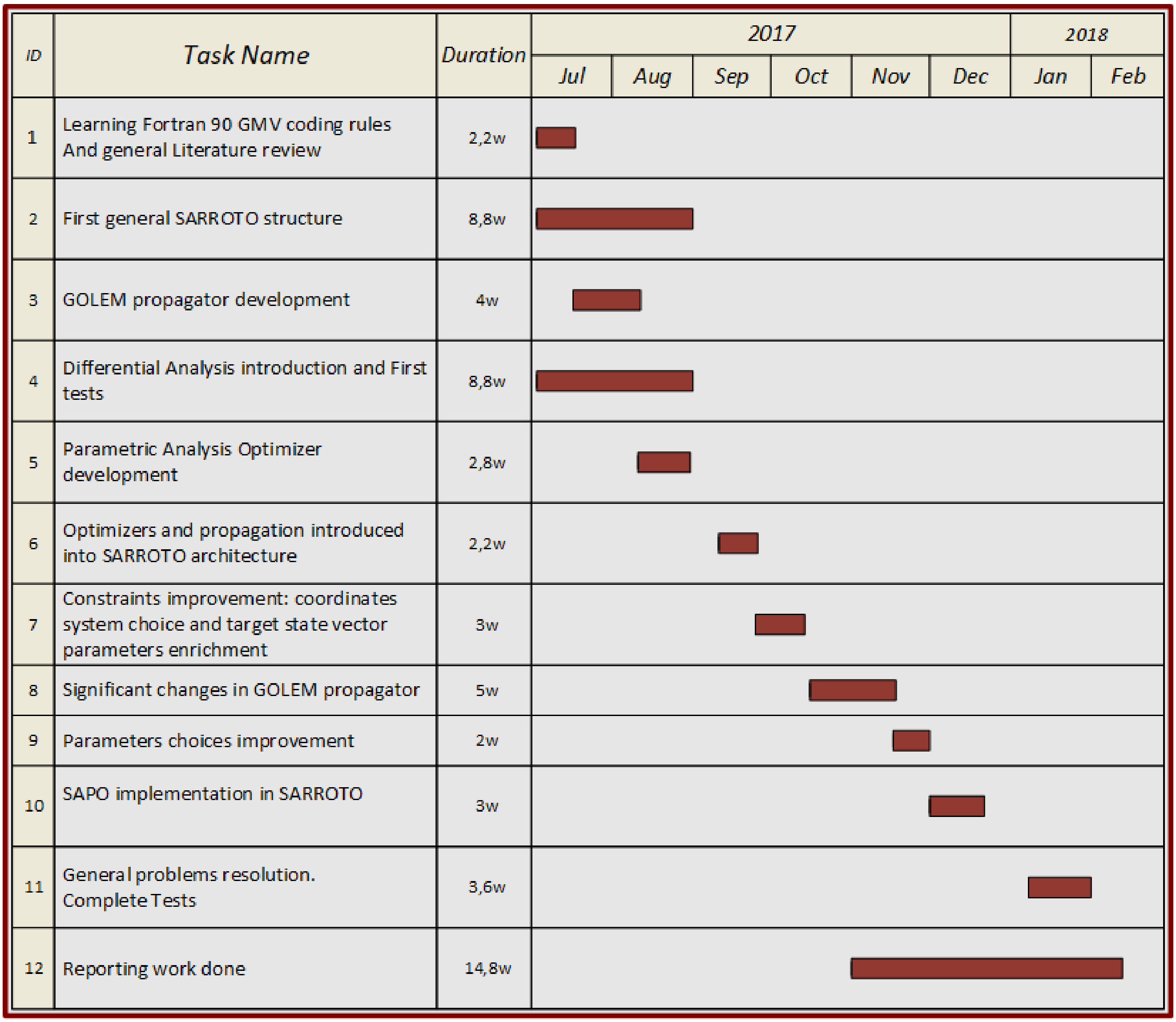

Figure 3.29: Gantt chart SARROTO timeline planning.

Table 3.1: Subroutine and functions in SARROTO's manager and main option of FORTRAN code.

Subroutine/Function Name	Description	Inputs	Outputs
SAmn_MainProcess	This subroutine is the main process of the program	-	-
mn_Done	Deallocating all the variables along the main program	All the allocatable variables	The same as inputs
SAmo_MOptions	This subroutine loads the SARROTO main options values, which consists in satellite name, initial epoch and state vector, propagator to use	Just requires the options themselves	The same of the inputs, plus some constants which will be used in other modules, such the ones which allow distinguishing the main propagator

Table 3.2: Subroutine and functions in SARROTO's constraints module of FORTRAN code.

Subroutine/Function Name	Description	Inputs	Outputs
SAcn_ReadConstraints	Loading the constraints option	Constraints panel option	The same as inputs
cn_ReadTarget	Saving all the constraints information	Constraints panel value	The same as inputs
SAcn_Done	This subroutine computes freeing memory variables	Memory allocatable variables	Deallocating variables

Table 3.3: Subroutine and functions in SARROTO's parameters module of FORTRAN code.

Subroutine/Function Name	Description	Inputs	Outputs
SApa_ReadParameters	Loading the parameters option	Parameters panel option	The same as inputs
pa_FixedNumMano	Saving information of parameters in case of fixed number of manoeuvre (no optimization): manoeuvres information, epoch and ΔV components values	Parameters panel values	The same as inputs
pa_VariableNumMano	Saving information of parameters in case of variable number of manoeuvre (optimization required): manoeuvres information, epoch and ΔV components values	Parameters panel values	The same as inputs
SApa_Done	This subroutine computes freeing memory variables	Memory allocating parameters variables	Deallocating parameters variables

Table 3.4: Subroutine and functions in SARROTO's computation module of FORTRAN code.

Subroutine/Function Name	Description	Inputs	Outputs
SAco_Computation	This subroutine processes all the program computation: initializing local variables, optimizer selection, coming up with the expected results	Parameters panel option, constraints, optimizer information, propagators information, manoeuvres list	Best parameters combination and best cost function values, final state vector, logical flag which inform whether target state vector has been reached
SAco_AuxPanelInit	Allocate auxiliary parameters panel variables	Auxiliary parameters panel variable	Parameters panel variables, best combination vector
SAco_Done	This subroutine computes freeing memory variables present in computation module	Memory allocating parameters variables: best combination vector, final state vector, logical flag for reaching target state vector values	Deallocating variables described in inputs

Table 3.5: Subroutine and functions in SARROTO's cost function module of FORTRAN code.

Subroutine/Function Name	Description	Inputs	Outputs
SAcf_Init	This subroutine processes initialization for local variables and optimizer election	Parameters panel option, constraints, optimizer information, propagators information	Module's local variables
SAcf_Probl	Computation of the objective function, in this case the one called *cost function*	Best computation vector	Cost function value
SAcf_CostFun	This subroutine processes the orbit propagation, computing coordinates transformation if necessary (i.e when target state vector parameters are defined in different coordinated respect to the ones given as inputs), saving manoeuvre information and best vector value, calculate cost function variable	Best computation vector, manoeuvres information	Cost function value, Final state vector, Arc values, logical flag for reaching target state vector values
SAcf_ManoeuverList	Loading manoeuvres information and generating a list of file	Manoeuvre information from parameters panel list	The same as inputs
SAcf_Done	This subroutine computes freeing memory variables present in cost function module	Memory allocating parameters variables: local variables	Deallocating local variables

Table 3.6: Subroutine and functions in SARROTO's GOLEM module of FORTRAN code.

Subroutine/Function Name	Description	Inputs	Outputs
SAgo_ReadOptions	This subroutine loads the GOLEM propagator main option	GOLEM panel option: step size for moving along time interval	Propagator options
SAgo_Propagate	Execute propagation starting from the initial state vector and time epoch	Initial state vector, initial epoch of propagation, final epoch of propagation, step size, manoeuvre list, solar radiation pressure variable value	Arc being generated, Final state vector
go_ComputeInteg	Subroutine which processes the main steps of the propagtions that are integration, coverting keplerian elements into cartesian one, writing the arc file	Initial state vector with initial epoch, number of steps for the considered epoch interval, longitudinal acceleration, constant solar radiation pressure, step size	State vector at each epoch in the considered time interval, parameters vector to update at each stepsize movement along the interval, Arc values
go_Integrate	Subroutine to calculate state vector values through the main equations	Initial state vector with initial epoch, number of steps for the considered epoch interval, longitudinal acceleration, constant solar radiation pressure, step size	State vector at each epoch in the considered time interval, parameters vector to update at each stepsize movement along the interval

Table 3.6 Continued: Subroutine and functions in SARROTO's GOLEM module of FORTRAN code.

Subroutine/Function Name	Description	Inputs	Outputs
go_rLongAcc	Subroutine to calculate longitudinal acceleration with linear interpolation	Osculating longitude, longitudinal accelerations values for each longitude	Longitudinal acceleration value for considering longitude value
go_OrbPeriod	Function to calculate orbital rotation period	Semi major axis	Orbital period value
go_SMAvar	Function to calculate new SMA at each new epoch, considering the SMA itself change of rate	RAAN at previous step epoch, constant longitudinal acceleration, ΔT	New SMA value
go_MeanVelocity	Function to calculate mean velocity value	SMA value	Mean velocity value
go_RaanCal	Function to calculate new RAAN at each new epoch, considering the RAAN itself change of rate	Semi major axis, RAAN at previous step epoch, eccentricity, inclination, ΔT	New RAAN value

Table 3.8: Subroutine and functions in SARROTO's optimization module of FORTRAN code.

Subroutine/Function Name	**Description**	**Inputs**	**Outputs**
SAop_Optim	Loading the information about the selected optimizer	Optimiser parameters panel	The same as inputs

Table 3.9: Subroutine and functions in SARROTO's Parametric analysis module of FORTRAN code.

Subroutine/Function Name	Description	Inputs	Outputs
SApn_Optimization	Compute parametric analysis optimizer: generating combinations	Number of parameters to be optimized, Interval values for optimized value research	Best combination vector and cost function value
pn_CalculateOp	Mathematical calculation for cost function value	Any combination vector	Cost function value
pn_UpdateOptim	Compute best cost function value checking that no redundant epoch are present	Objective function, combination vector, number of parameters to be optimized, number of epochs to be optimized	Optimum combination parameters vector and cost function value
pn_TestEpoch	This subroutine computes test to verify that no equal manoeuvre epoch are present in the same vector	Flag variable which counts epochs, combiantion vector, number of parameters	Flag for testing no redundant epoch of manoeuvre present in the same vector

Table 3.10: Subroutine and functions in SARROTO's SAPO interface module of FORTRAN code.

Subroutine/Function Name	Description	Inputs	Outputs
SAsi_Init	Update parameters value for SAPO propagation	Panels name from which getting data, satellite name, initial and final propagation epoch, initial state vector	-
SAsi_Propagate	Computing propagation steps with SAPO	Satellite name, manoeuvre list	Final state vector, obtained arc
SAcf_Done	This subroutine computes freeing memory variables present in SAPO's interface module	Memory allocating parameters variables	Deallocating variables

Table 3.11: Subroutine and functions in SARROTO's outputs module of FORTRAN code.

Subroutine/Function Name	Description	Inputs	Outputs
SAou_WriteOutputs	General subroutine for processing outputs	Manoeuvre list, Main panel values, constraints and propagators option, final state vector, flag whether target state vector has been reached	-
ou_WriteStdout	Private subroutine for processing the output printing	Manoeuvre list, Main panel values, constraints and propagators option, final state vector, flag whether target state vector has been reached	-
ou_WriteOrbitFile	Private subroutine for printing orbital results into file	Arc values, satellite ID, Initial and Final Epoch	-
ou_WriteManFile	Private subroutine for printing manoeuvre results into file	Arc values, satellite ID, Initial and Final Epoch	-

Chapter 4

Results

This chapter is dedicated to SARROTO output results in terms of comparison between the various tool options and propagators. Different optimisation cases will be illustrated. Results obtained will be discussed. In particular all possible combinations of propagators and optimisers, implemented with different constraints and parameters, will be analysed in order to give the readers a complete background about SARROTO working method.
Within the chapter it will be illustrated tables including the variables and options introduced through the ***focussuite*** interface. Parameters to be optimised will be indicated in table with ***To be Optimised*** statement. On the other hand the ones indicated as ***Fixed*** will take a specific value that will not change during optimisation procedure.

4.1 Orbit propagators comparison

As mentioned in the previous section, many differences can be listed between SAPO and GOLEM. It could be interested to compare GOLEM obtained orbits with a different propagator. In this sense a numeric one, such as Propag, is selected for this analysis. PROPAG is the main propagator used in GMV. It is considered *flight proven*. It has been validated so it can be used as a real propagation reference.
Readers can find more information about this propagator in appendix section E. In next lines all the propagators, two-by-two, will be put in comparison. Even though SAPO has not been developed during this internship, its results will be compared to the PROPAG's ones in order to check its reliability in SARROTO optimization process.

All the orbit comparisons have been obtained thanks to another ***focussuite*** software tool, named **ORBCOMP**, as already mentioned in section 1.2. Its principal function is comparing orbit files that have been given as inputs. It offers, as results, plots diagram which can be configured with user needs. In these comparison cases it will be set in such a way that on the horizontal axis the date will be displayed, and on the vertical one the position components, respectively along, cross and in plane. In addition it will be shown the total position, too. It has to be underlined that plots are based on the numerical difference between the two orbits. For this reason lower values taken by these components are translated in terms of propagator efficiency, compared with PROPAG.

As explained in previous section 3.1.1.3, perturbations are implemented in different ways in GOLEM and SAPO. Indeed in GOLEM the two perturbations introduced are the ones due to solar radiation pressure and not-spherical Earth. This last will be represented through the second zonal harmonic J_2 and longitudinal acceleration.
On the contrary PROPAG includes a wide number of perturbations. The complete list can be seen in Fig. E.2, where the PROPAG ***focussuite*** panel concerning perturbations is shown. Information about this propagator and its main characteristics are present in appendix section E.
These comparison tests are made taking into account perturbations which can be introduced in each propagator. Three cases of comparison will be presented:

1. First comparison between GOLEM and PROPAG. This last will be configured with J_{22} and solar radiation pressure in order to resemble GOLEM. This comparison has been performed in order to check GOLEM working method implemented in SARROTO.

2. Second comparison between GOLEM and PROPAG. The numerical propagator will be configured with all its permitted perturbations. In this way a real orbit will be compared with the GOLEM one. This new case has been computed in order to show propagation error with respect to a real case.

3. Comparison between SAPO and PROPAG. Both will be configured with all the perturbations defined in each one. With this comparison SAPO's lack of precision in representing a perturbed orbit will come up.

In Tab. 4.1 reader can appreciate the constant coefficients which define the main satellite and the ones which describe perturbations effects such as the solar radiation pressure. These information has to be introduced since in PROPAG and SAPO solar radiation pressure constant does not include mass and area contributes. On the contrary in GOLEM, as seen in section 3.1.1.3, satellite's mass and area have already been taken into account in its computation.

Constant coefficients	
Constants Parameters	Manoeuvres events
Mass = 123 kg SRP Area = 10; m^2 SRP = 0.114845	Two manoeuvres at following epoch 2017/01/01-03:20:00.500 2017/01/01-03:40:00.500

Table 4.1: Constant coefficients for satellite's definition in SAPO and PROPAG.

- **First Comparison-GOLEM and PROPAG**

In this first case of comparison between orbits it will be analysed an orbit provided with two manoeuvres events. All the propagated orbit characteristics are shown in Tab. 4.2.

Options case 1			
Initial state vector in Keplerian elements	Initial epoch of propagation [UTC]	Final epoch of propagation [UTC]	Perturbations
SMA=42134.0 km e=0.00001 i=0.0000001 Ω-RAAN=0 deg ω-Arg.Per=0 deg θ-Tr.Anomaly=0 deg	2017/01/01-00:00:000	2017/01/02-00:00:000	SRP, Non-spherical Earth effects

Table 4.2: First Comparison Case - Orbit data.

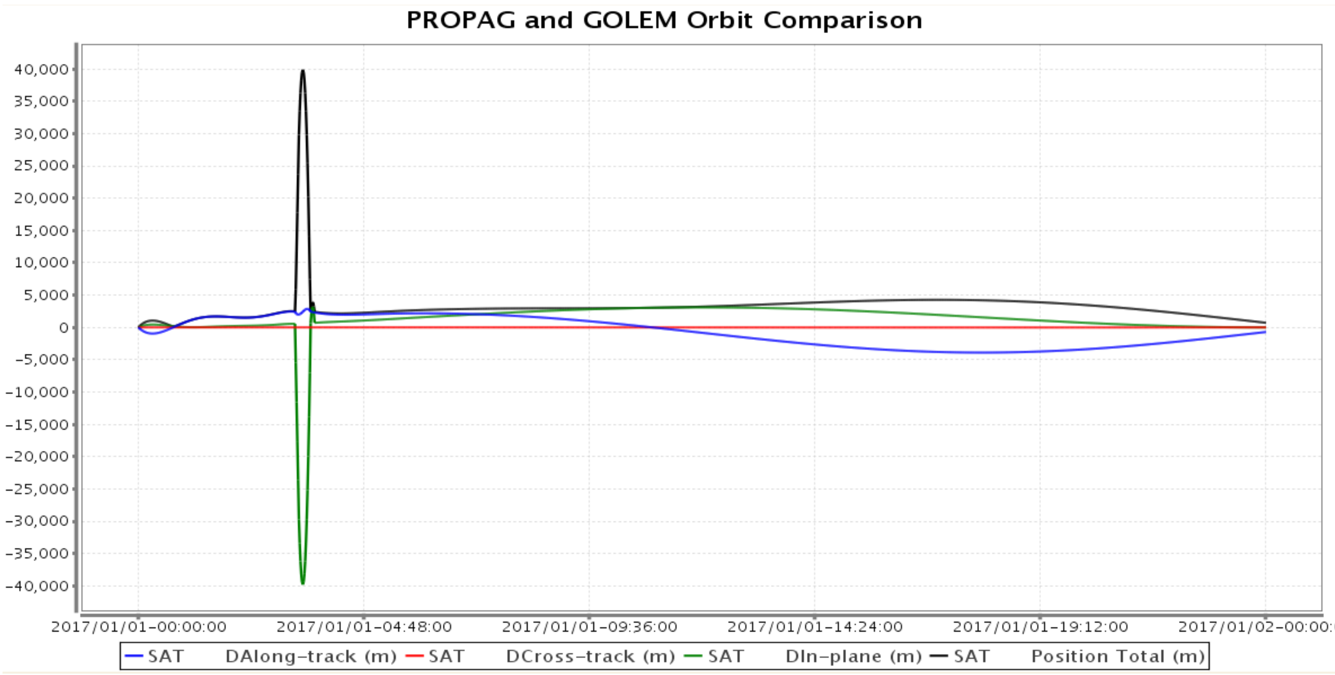

Figure 4.1: Orbit comparison between GOLEM and PROPAG.

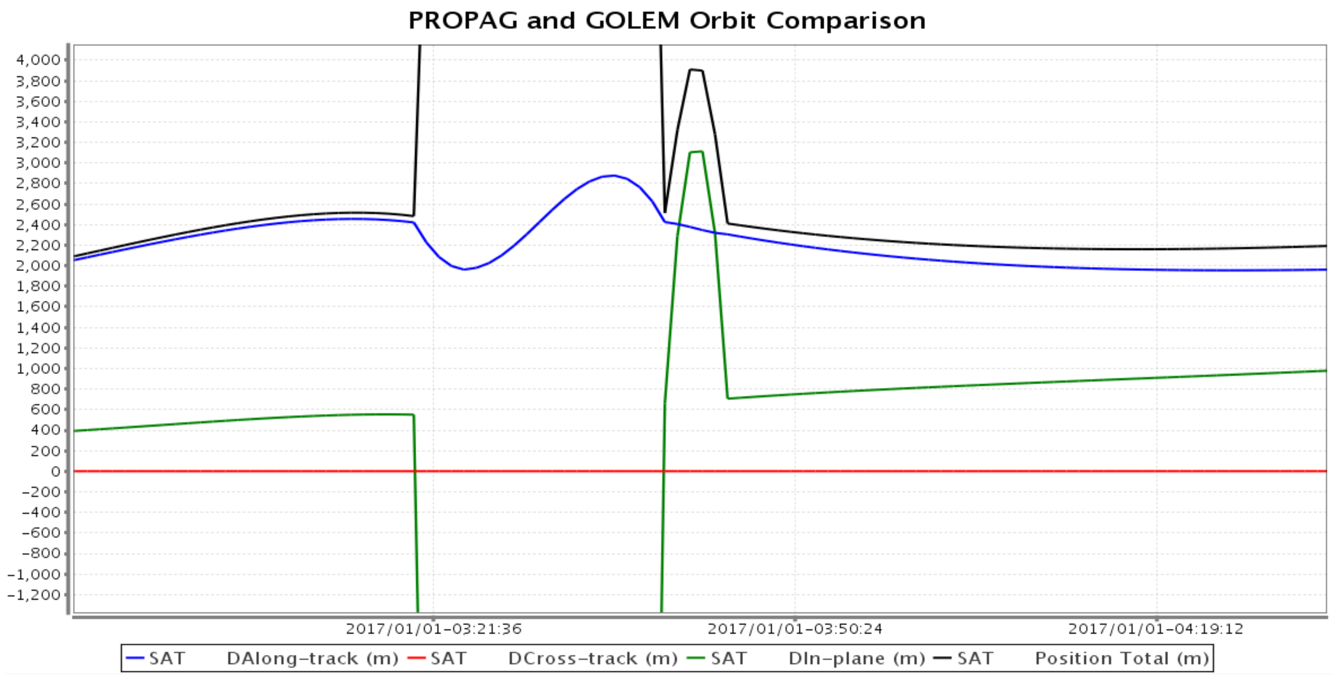

Figure 4.2: Orbit comparison between PROPAG and GOLEM, focusing on the manoeuvres events.

As it can be noticed the difference in terms of position is of the order of kilometres. The greater difference can be observed in the Cross-Track component difference at manoeuvre events epoch, as it is possible to see in Fig. 4.1. This phenomenon can be explained referring to the limitation of analytic theory implemented in GOLEM. Indeed, before manoeuvre event, differences between orbits depend on GOLEM lack of precision due to its way to approach perturbations. After first manoeuvre the in-plane component of GOLEM orbit increases with respect to PROPAG one. The manoeuvre event modifies eccentricity value in such way that GOLEM theory can be no more considered a valid method of orbit propagation.
The presence of a second manoeuvre balances the first manoeuvre effects. In this way propagation between the two orbits gets back to be similar and GOLEM theory can be considered functional in representing orbit.

- **Second Comparison Case - GOLEM and PROPAG**
  In this second case it has been compared the orbit propagated con GOLEM with the one obtained with PROPAG considering all the permitted perturbation. The PROPAG panel configured with these ones is shown in Fig. 4.3. Since the considered orbit is a GEO one, perturbation effects due to aerodynamic forces, atmospheric gravity, ocean tides would not be taken into account. The same is for Albedo radiation and other that would not affect the satellite because of the orbit height.

Options case 2				
Initial state vector in Keplerian elements	Initial epoch of propagation [UTC]	Final epoch of propagation [UTC]	Perturbations in PROPAG	Perturbations in GOLEM
SMA=42134.0 km e=0.00001 i=0.0000001 Ω-RAAN=0 deg ω-Arg.Per=0 deg θ-Tr.Anomaly=0 deg	2017/01/01-00:00:000	2017/01/02-00:00:000	SRP, 30x30 degree and order of gravity field, Moon and Solar Gravity perturbation, J_2 Moon effects	SRP, Non-spherical Earth effects

Table 4.3: Second Comparison Case-Orbit Data.

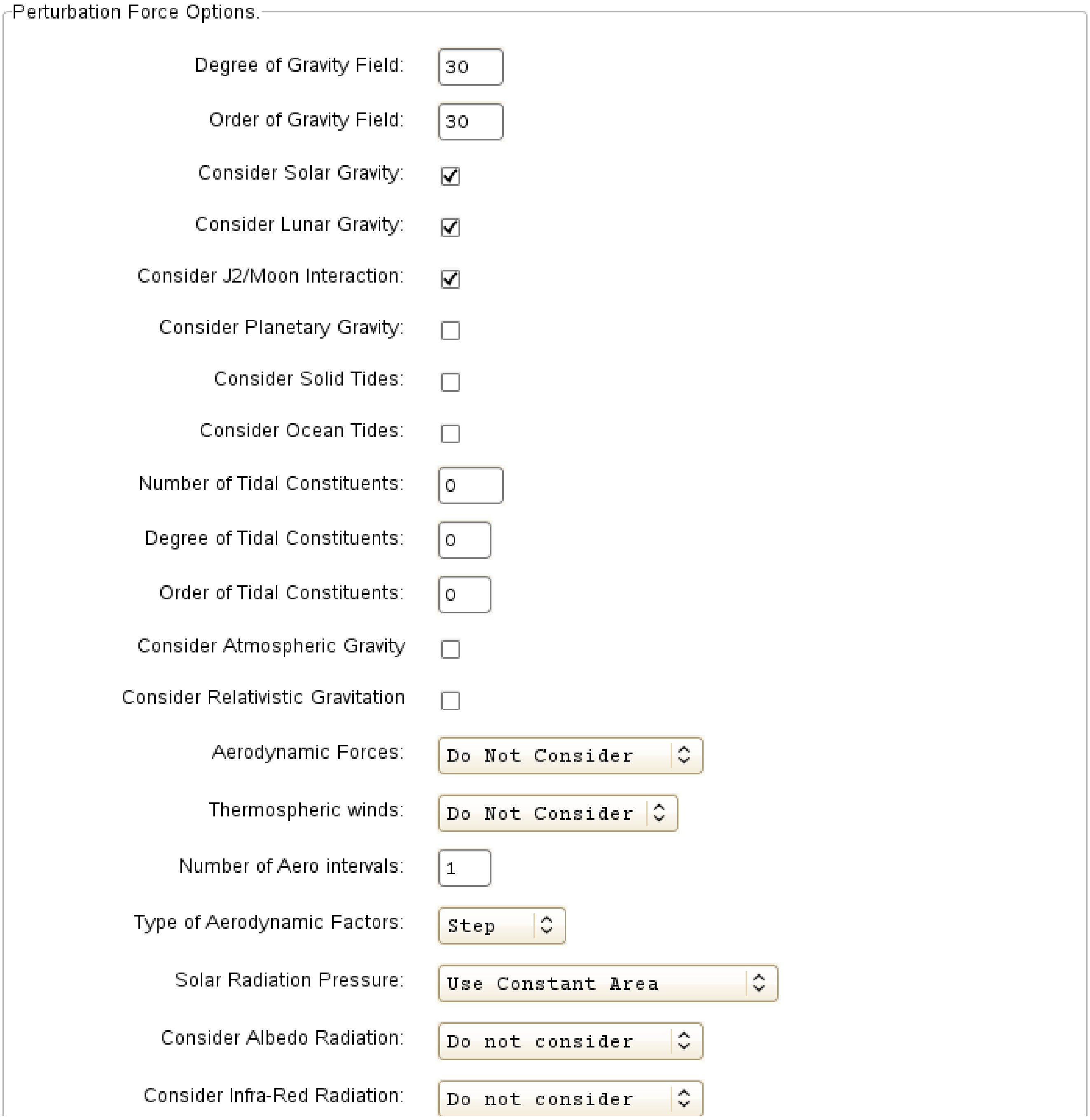

Figure 4.3: PROPAG perturbation panel configured with the significant perturbations for second comparison case.

Same remarks for this case can be introduced. Differently from the first comparison, the difference in along-track and cross-track components of position increase within the orbit propagation. This is due to Sun and Moon gravity perturbations which are not taken into account in GOLEM.

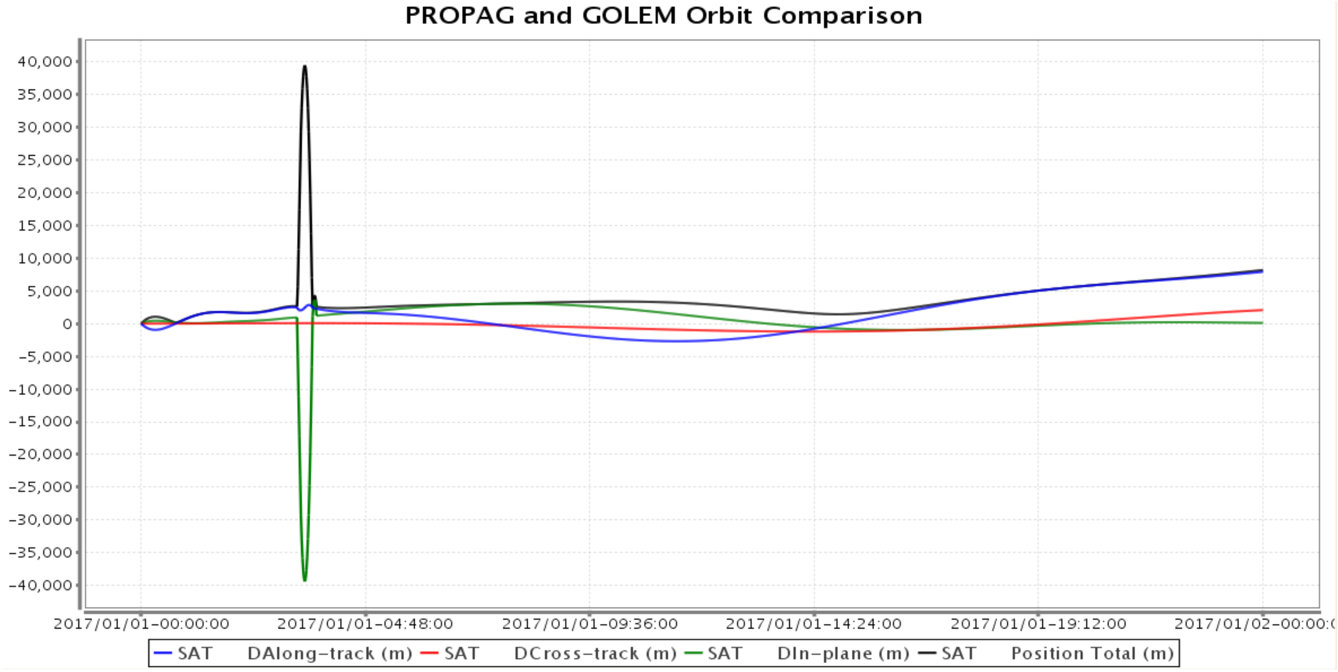

Figure 4.4: Comparison between PROPAG, considering all perturbations effects, and GOLEM.

- **Third Comparison Case - SAPO and PROPAG**
  In this last comparison there will be compared the orbit propagated with SAPO and PROPAG considering all the SAPO permitted perturbations. The information about orbit and perturbations are the ones illustrated in Tab. 4.4.

Options case 2				
Initial state vector in Keplerian elements	Initial epoch of propagation [UTC]	Final epoch of propagation [UTC]	Perturbations in PROPAG	Perturbations in SAPO
SMA=42134.0 km e=0.00001 i=0.0000001 deg Ω-RAAN=0 deg ω-Arg.Per=0 deg θ-Tr.Anomaly=0 deg	2017/01/01-00:00:000	2017/01/02-00:00:000	SRP, 30x30 degree and order of gravity field, Moon and Solar Gravity perturbation, J_2 Moon effects	SRP, 30x30 degree and order of gravity field, Moon and Solar Gravity perturbation, J_2 Moon effects

Table 4.4: Third Comparison Case-Orbit Data.

As it can be seen in Fig. 4.5, the position difference between these two propagation is of order of less than one kilometres. This is a check of the efficiency of SAPO propagator in considering perturbations phenomena.

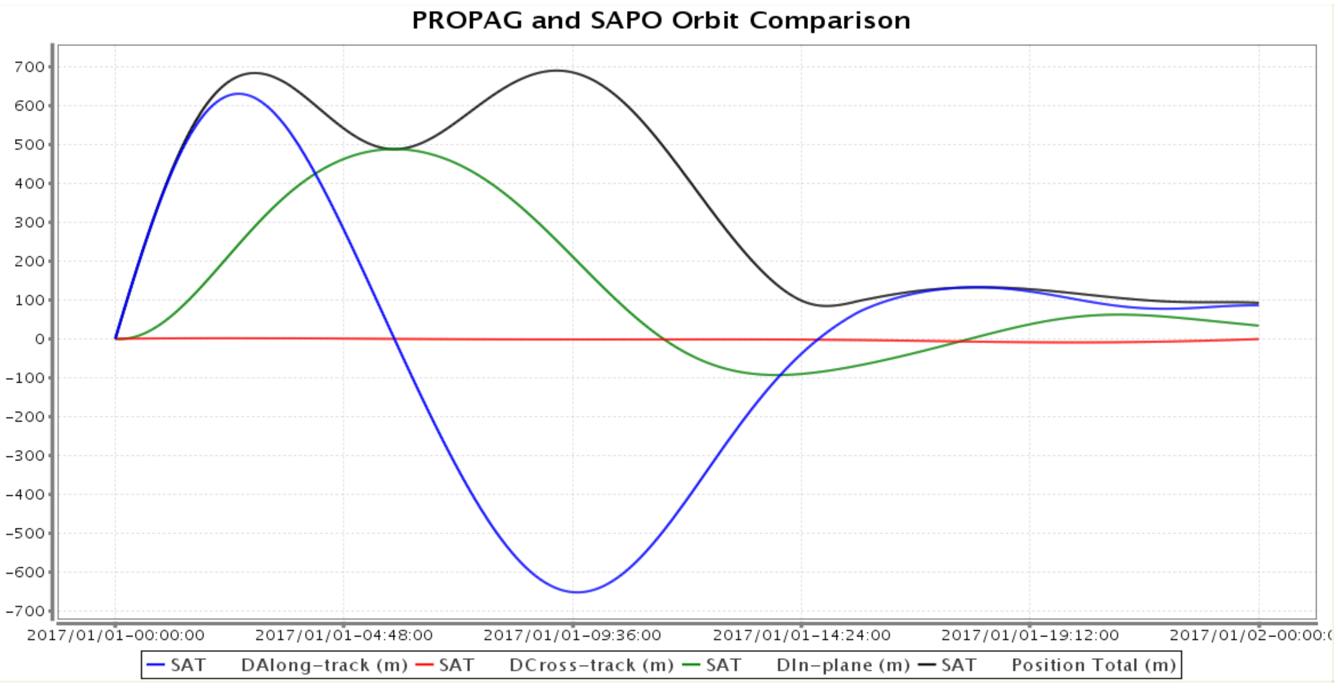

Figure 4.5: Comparison between PROPAG and SAPO, considering all perturbations effects.

4.2 Computational time comparison

Before testing the SARROTO optimisation process, a comparison between computational time obtained performing test cases with DE and PA are presented. A single optimised test has been examined, setting options present in Tab. 4.5. In order to propagate similar orbit, SAPO has been configured with same perturbations permitted in GOLEM: solar radiation pressure and non-spherical Earth effects.
In Tab. 4.6 are present computational time elapsed for each combination of propagator and optimiser. This test has been made in order to make readers aware that this aspect has been considered in preparing tests cases to show.

<table>
<tr><th colspan="4">Comparision Case Option</th></tr>
<tr><td rowspan="2">One day propagated orbit</td><td colspan="3">Initial Epoch
2017/01/01-00:00:00:000</td></tr>
<tr><td colspan="3">Final Epoch
2017/01/02-00:00:00:000</td></tr>
<tr><td rowspan="8">Two manoeuvres with
ΔV along track
and manoeuvre epochs to be optimised</td><td rowspan="4">First Manoeuvre</td><td rowspan="3">ΔV_1</td><td>Cross
Fixed</td></tr>
<tr><td>Along
To Be Optimised</td></tr>
<tr><td>Radial
Fixed</td></tr>
<tr><td colspan="2">$Epoch_1$
To Be Optimised</td></tr>
<tr><td rowspan="4">Second Manoeuvre</td><td rowspan="3">ΔV_2</td><td>Cross
Fixed</td></tr>
<tr><td>Along
To Be Optimised</td></tr>
<tr><td>Radial
Fixed</td></tr>
<tr><td colspan="2">$Epoch_2$
To Be Optimised</td></tr>
</table>

Table 4.5: Analyzed case for comparison tests.

Comparison between computational times for the same analysed case		
Time spent	SAPO	GOLEM
Differential Evolution	16.06 min	0.108 min
Parametric Analysis	62 min	0.383 min

Table 4.6: Comparison between computational time spent for achieving the best result.

As it can be appreciated, the combination differential evolution and GOLEM propagator is the one which takes less time to give results as output. On the other hand SAPO and Parametric analysis is the one which takes most.

4.3 Analyzed optimization case

In next tables there will be presented all the tests made, with the selected requirements and options. The inputs are the ones concerning initial state vector and target state vector elements choices. Parameters table shows the manoeuvres information, with parameters to be optimised. The output is related to final state vector and parameters values coming from the optimisation. The cases have been performed testing all the tools introduced into the main program, in order to demonstrate the reliability of this main software. The different combinations concern the propagators and optimizers different solutions.
Tests have been performed following the steps presented in the below list:

- Defining through the ***focussuite*** interface the input values, from the initial state vector to the desired propagator;
- Selecting the desired target state vector elements, with the respective weight and tolerance;
- Choosing the parameters to be optimised, introducing the searching interval and stepsize;
- Selecting the optimizer and its options;
- Once results have been obtained, the output file has been analysed;
- All combination between propagators and optimziers, when permitted by chosen options, have been tested;

In order to keep the readers comfortable with the tests which will be illustrated, acronyms have been used to refer to propagators and optimizers combination case, in particular

- **GODE**: GOLEM and Differential Evolution
- **SADE**: SAPO and Differential Evolution

- **GOPA**: GOLEM and Parametric Analysis
- **SAPA**: SAPO and Parametric Analysis

4.3.1 Hohmann transfer calculation

As mentioned during different SARROTO's modules explanation, there are some limitations which have affected the way of doing tests. In particular when the choice to optimize both epoch and ΔV is made, to narrow down the searching interval, it is a reasonable idea to estimate the components of velocity to be optimised. This last requirement is introduced in order to spend less computational time to get the correct solution. So, in order to give a first approximated value for the ΔV along track component of velocity, the Hohmann transfer calculation has been utilised. In particular equations used to perform calculation can be found in appendix section B.1. Detailed notions about this transfer process between two orbits can be found in appendix section B.
As explained in appendix section B, the along track component of velocity is the only, among the three, taken into account when a circular geostationary orbit would be obtained manoeuvring from a lower circular orbit to an higher circular one without changing orbit plane. Indeed, even though radial component may allow this transfer, the manoeuvre with along track is the best in terms of efficiency. The other components are set to zero value in all test cases.
So in this case, two tangential manoeuvres have been chosen to transfer satellite from the first circular orbit to the new circular one. Actually this transfer may be performed with greater number of manoeuvres. The first orbit semi-major axis is indicated with the subscript **I**, while the second one with **T**, which means respectively initial and target.
Actually, as it will be appreciated in next performed cases, the considered orbit is not perfectly circular, since this case does not exist in reality. For this reason the better case is the one with eccentricity and inclination case close to zero value. The velocity speed of the satellite related to the departure point is

$$\Delta V_{Transf} = \sqrt{\frac{\mu_{Earth}}{R_{ISV}}} \cdot \left(\sqrt{\frac{2R_{TSV}}{R_{ISV} + R_{TSV}}} - 1 \right)$$

where $R_{ISV} = SMA_{ISV} + R_E = 42134\ km$ and $R_{TSV} = SMA_{TSV} + R_E = 42164\ km$ with R_E the Earth radius.
For this calculation it is assumed that eccentricity and inclination are equal to zero. This statement is a reasonable simplification for the considered case. Applying this simple formula it results that the increment of velocity component, for both manoeuvres, should take value within $\Delta V = 0.55\ m/s$. This result will be used as input for the searching interval bounds, and as it will be illustrated, it will be quite verified from computational calculation.

4.4 Test Phase

Some remarks are necessary in order to focusing on the main points of this program validation phase and in particular to better understand cases which will be presented.

As readers will notice within this test section, all tests, except from Test 4 and 5 where different test option have been required, introduce as target state vector just three elements

- SMA: the semi major axis,
- e: eccentricity,
- λ: longitude

Before discussing about this last assumption, it has to be reminded that, due to the GOLEM propagator main definition, a geosynchronous orbit has been taken as reference for all calculations made within this thesis.
So, once this aspect has been clarified, the previous statement can be explained. It has to be considered that, when speaking about a GEO orbit with inclination and eccentricity equals to zero value, or really closer to this value, it has no meaning defining the RAAN and the Arg. of Periapsis. In fact for this type of orbit, a non-inclined circular orbit, Ω and ω are not-defined and for this reason it is usual to give them zero value in order to refer to the vernal equinox direction. In addition, all the analyzed cases consider a East-West Manoeuvre. As a consequence inclination has not been introduced in the target state vector, since there are not changes in the orbit plane.
On the other hand longitude can be very useful in describing spacecraft position for a geosynchronous circular and equatorial orbits, since a satellite orbiting along a GEO one stays *in place over a single longitude*, as defined in [18].

4.4.1 TEST 1-Two Manoeuvre with velocity along track component and epoch to be optimised

This first test has been carried out considering two manoeuvres with both along-track component of velocity and epoch to be optimised. It has to be said that in all performed tests, among the components of velocity, just the along track will be taken into account. It will be the only velocity component with a no-zero value, both in optimised or fixed analysed case.

It has to be reminded that an along track manoeuvre can be defined in other words as east-west or longitude manoeuvre in GEO orbit. This manoeuvre type just considers the tangential thrust to the orbit and it is performed when it is required a change in the orbital semi-major axis, in the longitude drift rate or in the eccentricity vector. For these reasons this manoeuvre is commonly implemented when a geostationary orbit is desired.

This is due to the intention of implement an orbit propagation with manoeuvre events which could simulate the satellite's transfer from a circular orbit to an higher circular one. The first circular orbit semi-major axis will be numerically lower than the geostationary orbit height, and the height of the new one will be equal to this last said. This transfer will be performed through two manoeuvres, which can be approximated with the Hohmann transfer, as said in 4.3.1.
Actually the considered manoeuvre, in order to be circular and in plane, should have eccentricity and inclination values equals to zero. This case may introduce problems in performing SAPO

propagators, due to some limitations in the numeric computation and truncation algorithms. To avoid this issue carrying on the desired test, very little values have been assigned to these keplerian elements, as it can be seen in Tab. 4.7. So as, the first assumption about a transfer between two circular in plane orbits can be preserved.

In Tab. 4.8 parameters values associated to this first test can be found. It must be underlined the importance of the tolerance and step size values. The last cited is very important in order to achieve solution: the secondary problem associated to its choice concerns the computational time required when small step size is introduced.
In addition to what previously mentioned about tolerances, some comments have to be made. Many tests have been performed, and clearly not all of them have come up with the expected solution. In some cases, the selection of very narrow tolerances values has not allowed to get the correct solution. This has meant not achieving target orbit, even thought only few meters and degrees of difference have occurred.

It has been chosen to define just three parameters in the target state vector, to better visualising the solution, and it has been preferred to give them the same weight. Parameters affected by the manoeuvre event, and more specifically by the along-track component of velocity, have been selected. These parameters are semi-major axis, eccentricity and longitude. Inclination has not been taken into account since cross-track component of velocity is set to zero. About manoeuvre events, two manoeuvres have been selected, whose parameters value can be found in Tab. 4.8. Both epoch and velocity along track component have been introducing as variables to be optimised, with fixed step size.

<table>
<tr><th colspan="4">SARROTO Inputs Test 1</th></tr>
<tr><td>Main features</td><td colspan="3">Two manoeuvres with epoch and along track velocity component to be optimized</td></tr>
<tr><td rowspan="2">Initial State Vector</td><td colspan="3">Initial Epoch
2017/01/01-00:00:00:000</td></tr>
<tr><td colspan="3">SMA = 42134.0 km
e = 0.00001
i = 0.0000001 deg
Ω = 0 deg
ω = 0 deg
θ = 0 deg</td></tr>
<tr><td rowspan="3">Target State Vector</td><td colspan="3">Final Epoch
2017/01/02-0:00:00:000</td></tr>
<tr><td>Coordinate
Value</td><td>Weight</td><td>Tolerance</td></tr>
<tr><td>SMA = 42164.0 km
e = 0.0000000000001
λ = -100.600845364 deg</td><td>1.0
1.0
1.0</td><td>0.01 km
0.01
0.1 deg</td></tr>
</table>

Table 4.7: Inputs Test 1.

<table>
<tr><th colspan="4">SARROTO Parameters Test 1</th></tr>
<tr><td>Manoeuvre definition</td><td colspan="3">Two manoeuvres with both ΔV-along track and epoch to optimize</td></tr>
<tr><td rowspan="8">ΔV components value</td><td rowspan="4">First Manoeuvre</td><td colspan="2">Radial
0.0 m/s</td></tr>
<tr><td rowspan="2">Along
To Be Optimized</td><td>Searching interval
[0.4, 0.6] m/s</td></tr>
<tr><td>StepSize
0.001 m/s</td></tr>
<tr><td colspan="2">Cross
0.0 m/s</td></tr>
<tr><td rowspan="4">Second Manoeuvre</td><td colspan="2">Radial
0.0 m/s</td></tr>
<tr><td rowspan="2">Along
To Be Optimized</td><td>Searching interval
[0.5, 0.6] m/s</td></tr>
<tr><td>StepSize
0.001 m/s</td></tr>
<tr><td colspan="2">Cross
0.0 m/s</td></tr>
<tr><td rowspan="4">Manoeuvre Epoch</td><td rowspan="2">First Manoeuvre</td><td rowspan="2">Epoch value
To Be Optimized</td><td>Searching interval
[2017/01/01/-00:00:00.000,
2017/01/01/-12:00:00.000]</td></tr>
<tr><td>StepSize
600 s</td></tr>
<tr><td rowspan="2">Second Manoeuvre</td><td rowspan="2">Epoch value
To Be Optimized</td><td>Searching interval
[2017/01/01/-12:00:00.000,
2017/01/02-00:00:00.000]</td></tr>
<tr><td>StepSize
600 s</td></tr>
</table>

Table 4.8: Parameters Test 1.

- ▷ **GOPA**
  It has been checked the combination between the GOLEM propagator and the parametric analysis optimizer. Inputs value can be reads in Tab. 4.7, where the main characteristics have been presented.

<table>
<tr><th colspan="3">SARROTO GOPA Output Test 1</th></tr>
<tr><td>Output results</td><td colspan="2">Execution Time =99.7071 min
Tested combinations = 108184029</td></tr>
<tr><td>Final State Vector</td><td colspan="2">SMA = 42164.009827224 km
e = 0.000427042
λ =-100.501470564 deg</td></tr>
<tr><td rowspan="4">Optimized values</td><td rowspan="2">First Manoeuvre</td><td>Epoch
2017/01/01-05:30:01.000</td></tr>
<tr><td>ΔV along
0.594 m/s</td></tr>
<tr><td rowspan="2">Second Manoeuvre</td><td>Epoch
2017/01/01-12:30:00.000</td></tr>
<tr><td>ΔV along
0.50 m/s</td></tr>
</table>

Table 4.9: Output GOPA Test 1.

- **SAPA**
  In this test case, SAPO and Parametric Analysis has not been tested since necessary time to get the final result was quite great to allow a reasonable computation. In fact number of combinations to be tested with parameters values detailed in Tab. 4.8 was equal to $N_{comb} = 108184029$ which means a computational time $t \approx 7336\ min = 122879\ hr \approx 5.11\ days$. SAPA tests will be performed with a reduced number of combinations as illustrated in next paragraphs.

- **GODE**
  In this second part of Test 1, it will be computed optimization through the differential evolution optimizer. As explained in dedicated sec. 3.1.2.1, there are some variables values which have to be defined when this choice is made. These ones can be found in Tab. 4.10

DE main option Test 1	
Fitness Expected	-10000
Population size	10000
Crossover factor	7
Iteration Number	80
Strategy	2

Table 4.10: Differential Evolution Main Option Test 1.

<table>
<tr><th colspan="3">SARROTO GODE Output Test 1</th></tr>
<tr><td>Output results</td><td colspan="2">Execution Time = 0.805 min</td></tr>
<tr><td>Final State Vector</td><td colspan="2">SMA = 42163.990002913 km
e = 0.000530626
λ =-100.500859070 deg</td></tr>
<tr><td rowspan="4">Optimized values</td><td rowspan="2">First Manoeuvre</td><td>Epoch 2017/01/01-06:33:48.382</td></tr>
<tr><td>ΔV along 0.593166406 m/s</td></tr>
<tr><td rowspan="2">Second Manoeuvre</td><td>Epoch 2017/01/01-12:00:00.000</td></tr>
<tr><td>ΔV along 0.500033232 m/s</td></tr>
</table>

Table 4.11: Output GODE Test 1.

- **SADE**
  Same inputs, parameters and constraints values used for GOLEM have been implemented in this second phase of test 1. It has just been choosing SAPO propagators as mean of orbital propagation mixed with the differential evolution analysis. As it can be appreciated from Tab. 4.12 target is achieved for all elements selected. In this case the total increment of velocity of the transfer is

$$\Delta V_{Tot} = |\Delta V_1| + |\Delta V_2| = 0.530450737 + 0.555343009 = 1.085793746 \; m/s$$

<table>
<tr><th colspan="3">SARROTO SADE Output Test 1</th></tr>
<tr><td>Output results</td><td colspan="2">Execution Time = 144.814 min</td></tr>
<tr><td>Final State Vector</td><td colspan="2">SMA = 42163.990077067 km
e = 0.000107511
λ =-100.500929115 deg</td></tr>
<tr><td rowspan="4">Optimized values</td><td rowspan="2">First Manoeuvre</td><td>Epoch 2017/01/01-00:43:14.109</td></tr>
<tr><td>ΔV along 0.530450737 m/s</td></tr>
<tr><td rowspan="2">Second Manoeuvre</td><td>Epoch 2017/01/01-12:00:00.301</td></tr>
<tr><td>ΔV along 0.555343009 m/s</td></tr>
</table>

Table 4.12: Output SADE Test 1.

Comments to test 1 First of all, it has to be said that it has not be possible to produce test with SAPA, due to the large amount of time required for the numerical computation. In fact, in order to obtain a precise solution with this combination case, not only days, but in few cases, weeks will be required.
It has not been possible to compare solution between the two propagators, but despite of this negative aspect, results obtained with GOPA can be evaluated positively since not only target has been achieved but the manoeuvre velocity component is in the range expected with Hohmann transfer ΔV calculation.
These expected results for each manoeuvre increment of velocity and total transfer ΔV can be appreciated looking at Fig. 4.6. In this bars diagram velocity value of each manoeuvre for analysed combinations cases are represented. As readers may appreciate, the total ΔV for each combination case is equal for all the tested case, and in addition is the same obtained for the Hohmann transfer. Actually, it cannot be guaranteed that a result which is equal o similar to the one obtained with Hohmann is the best solution. On the contrary, a result which significantly diverges from this last can be considered unreliable. However, as general final comment, it can be said that both propagators meet the expectations.

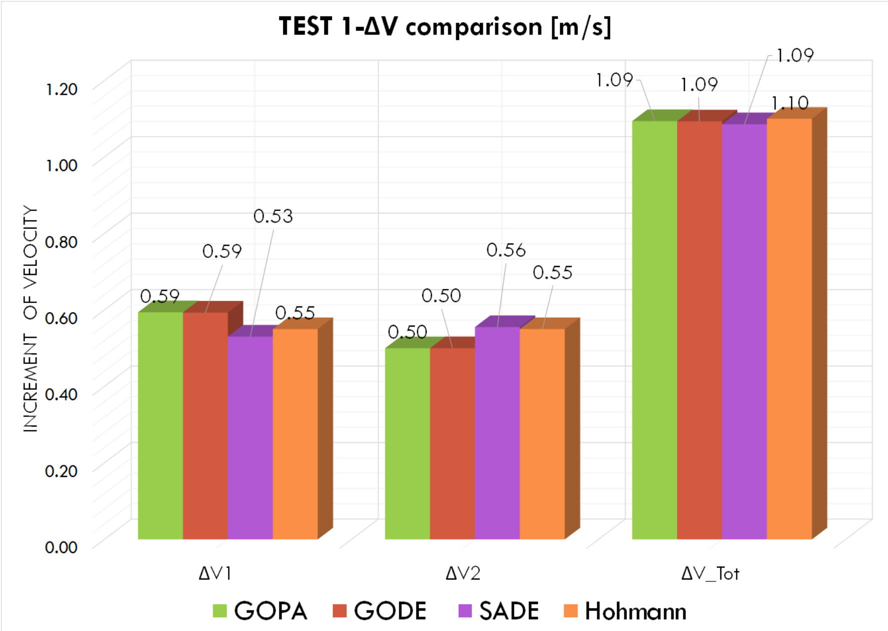

Figure 4.6: Comparison between ΔV for combination cases Test 1.

It has to be mentioned that similar values in final state vector are obtained in both cases. GODE solution results in the achievement of the target state vector, so does SADE. Target state vector has been reached without exceeding permitted tolerance values defined in Tab. 4.7 to input options. This tolerance range check can be verified in Fig. 4.7, 4.9, 4.8, where difference between each final and target state vector element is compared to the tolerance value taken as initial reference. In particular in these diagram bars represent difference between final and target SV element, the red line represents the tolerance value permitted.

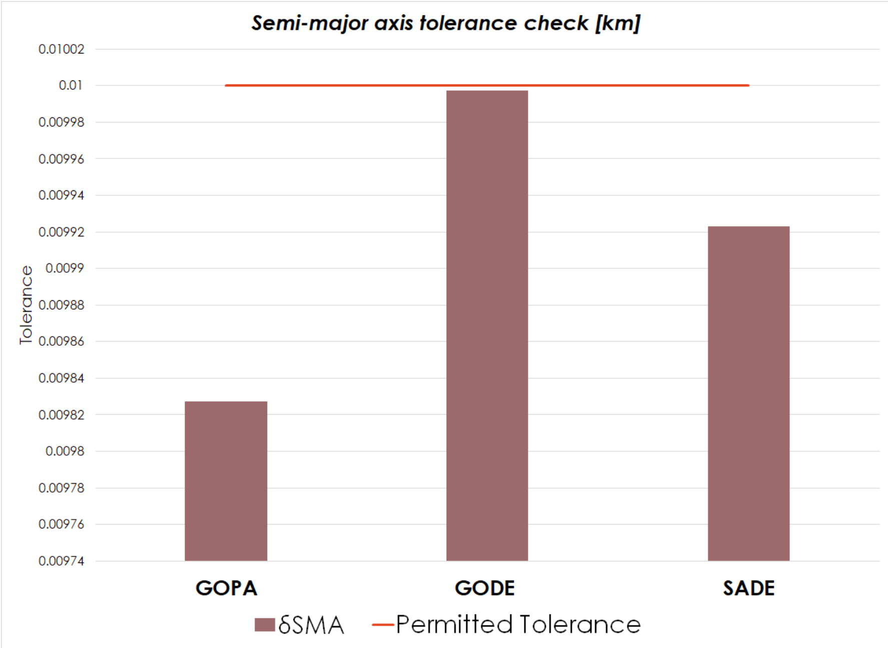

Figure 4.7: Checking tolerances for Semi-major axis-Test 1.

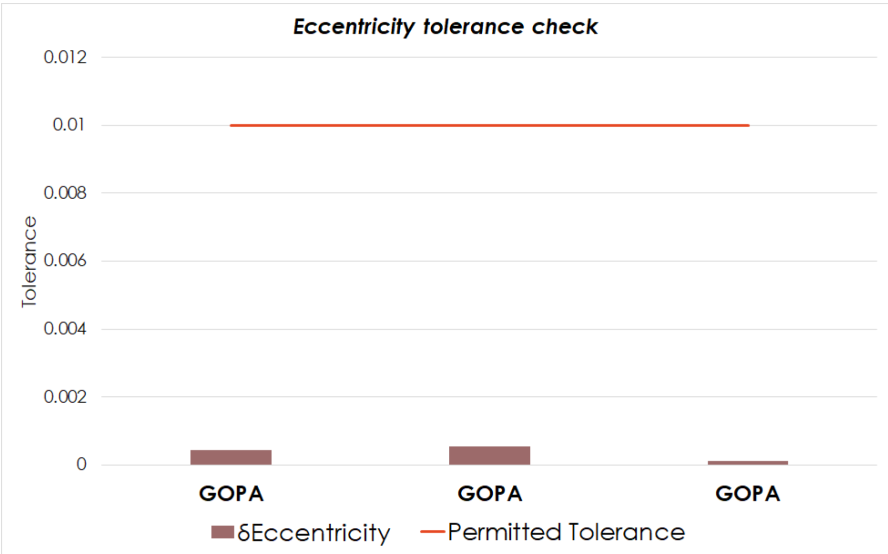

Figure 4.8: Checking tolerances for Eccentricity-Test 1.

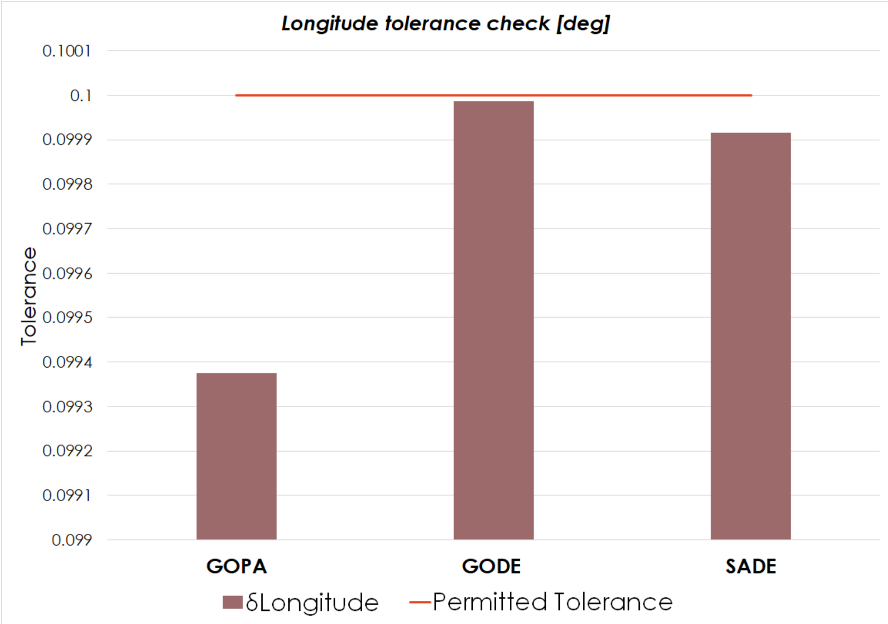

Figure 4.9: Checking tolerances for Longitude-Test 1.

In Fig. 4.10 a time axis with epochs at which occur manoeuvre events is shown. Readers could note that second manoeuvre epoch is located roughly on the same point for all the considered cases. On the other hand, about first manoeuvre epoch, similarities can be found only for GOLEM propagator cases.

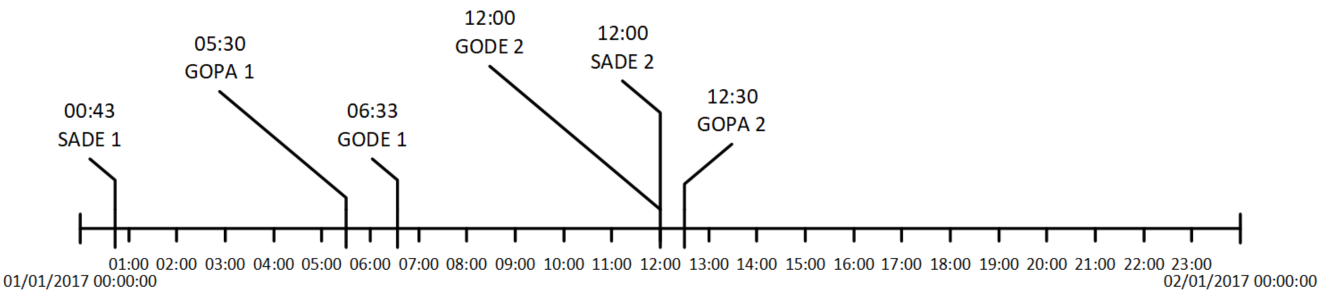

Figure 4.10: Manoeuvre events on propagation timeline Test 1.

In Fig. 4.11 computational time spent for execute SARROTO in each combination case is represented. It is possible to notice that SAPO is the one which requires major time for returning results. At second place, GOPA choice is the one which results with a bigger time elapsed.

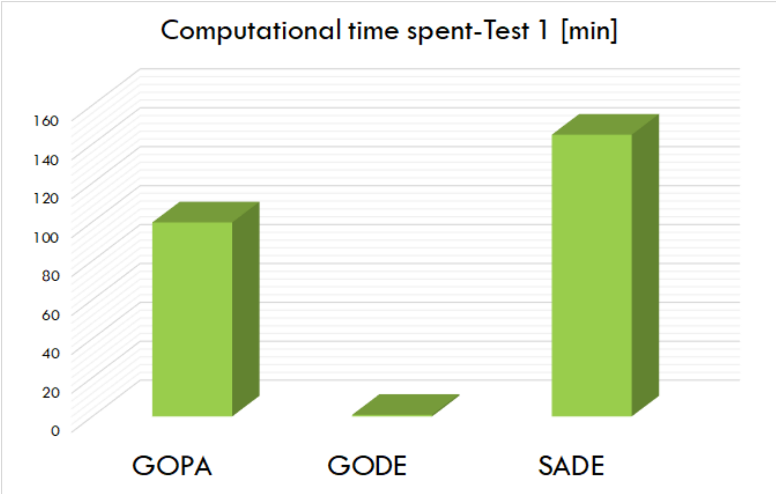

Figure 4.11: SARROTO execution time spent-Test 1.

4.4.2 TEST 2-Two Manoeuvre with epoch and along track component to be optimised searching in reduced intervals

In this second case it will be tested a case similar to the first but with searching intervals smaller than previous one. In addition smaller step size has been introduced. In this way SAPO functionality mixed with PA can be proved.In Tab.4.13 can be found data regarding input values for this second case test, indeed in Tab. 4.14 parameters variables.
In the SAPA case, it has been selected the option to use common searching interval for both manoeuvres, setting this last equals to the entire orbit propagation time. On the other hand for SADE test it has been necessary to split this interval in two sub-ranges since this functionality it is not implemented for the DE optimiser choice. This difference in the searching interval choice can be observed in Tab. 4.14 in the field named **Manoeuvre Epoch**.

<table>
<tr><th colspan="4">SARROTO Inputs Test 2</th></tr>
<tr><td>Main features</td><td colspan="3">Two manoeuvers
ΔV along track to be optimized</td></tr>
<tr><td rowspan="2">Initial State Vector</td><td colspan="3">Initial Epoch
2017/01/01-00:00:00:000</td></tr>
<tr><td colspan="3">SMA = 42134.0 km
e = 0.00001
Ω=0 deg
ω=0 deg
θ =0 deg</td></tr>
<tr><td rowspan="5">Target State Vector</td><td colspan="3">Final Epoch
2017/01/02-0:00:00:000</td></tr>
<tr><td>Coordinate
Value</td><td>Weight</td><td>Tolerance</td></tr>
<tr><td>SMA = 42164.0 km</td><td>1.0</td><td>0.12 km</td></tr>
<tr><td>e = 0.0000000000001</td><td>1.0</td><td>0.01</td></tr>
<tr><td>λ = -100.600845364 deg</td><td>1.0</td><td>0.1 deg</td></tr>
</table>

Table 4.13: Inputs Test 2.

<table>
<tr><th colspan="4">SARROTO Parameters Test 2</th></tr>
<tr><td>Manoeuvre definition</td><td colspan="3">Two manoeuvres with both ΔV-along track and epoch to optimize</td></tr>
<tr><td rowspan="8">ΔV components value</td><td rowspan="4">First Manoeuvre</td><td colspan="2">Radial
0.0 m/s</td></tr>
<tr><td rowspan="2">Along
To Be Optimized</td><td>Searching interval
[0.5, 0.6] m/s</td></tr>
<tr><td>StepSize
0.01 m/s</td></tr>
<tr><td colspan="2">Cross
0.0 m/s</td></tr>
<tr><td rowspan="4">Second Manoeuvre</td><td colspan="2">Radial
0.0 m/s</td></tr>
<tr><td rowspan="2">Along
To Be Optimized</td><td>Searching interval
[0.5, 0.6] m/s</td></tr>
<tr><td>StepSize
0.01 m/s</td></tr>
<tr><td colspan="2">Cross
0.0 m/s</td></tr>
<tr><td rowspan="6">Manoeuvre Epoch</td><td rowspan="3">First Manoeuvre</td><td rowspan="3">Epoch value
To Be Optimized</td><td>Searching interval PA
[2017/01/01-00:00:00.000,
2017/01/02-00:00:00.000]</td></tr>
<tr><td>Searching interval DE
[2017/01/01-00:00:00.000,
2017/01/01-12:00:00.000]</td></tr>
<tr><td>StepSize
900 s</td></tr>
<tr><td rowspan="3">Second Manoeuvre</td><td rowspan="3">Epoch value
To Be Optimized</td><td>Searching interval PA
[2017/01/01-00:00:00.000,
2017/01/02-00:00:00.000]</td></tr>
<tr><td>Searching interval DE
[2017/01/01-12:00:00.000,
2017/01/02-00:00:00.000]</td></tr>
<tr><td>StepSize
900 s</td></tr>
</table>

Table 4.14: Parameters Test 2.

▷ **SAPA**
SAPO and the parametric analysis have been tested as first in this second test. As it can be noticed in Tab. 4.13 tolerance value for SMA has been increased in order to achieve target.

<table>
<tr><th colspan="3">SARROTO SAPA Output Test 2</th></tr>
<tr><td>Output result</td><td colspan="2">Execution Time = 77.209 min
Tested combinations = 1138489</td></tr>
<tr><td>Final State Vector</td><td colspan="2">SMA = 42164.115831782 km
e = 0.000664015
λ =-100.501287321 deg</td></tr>
<tr><td rowspan="4">Optimized values</td><td rowspan="2">First Manoeuvre</td><td>Epoch
2017/01/01-09:15:00.000</td></tr>
<tr><td>ΔV along
0.590000000 m/s</td></tr>
<tr><td rowspan="2">Second Manoeuvre</td><td>Epoch
2017/01/01-09:30:00.000</td></tr>
<tr><td>ΔV along
0.500000000 m/s</td></tr>
</table>

Table 4.15: Output SAPA Test 2.

▷ **SADE**
The differential evolution has been used in this case keeping fixed constraints and parameters values utilised for parametric analysis computation. The searching interval, as mentioned at first, has been changed. The Differential evolution have been performed with main options values represented in Tab. 4.16

DE main option Test 2	
Fitness Expected	-10000
Population size	10000
Crossover factor	7
Iteration Number	30
Strategy	2

Table 4.16: Differential Evolution Main Option Test 2.

<table>
<tr><th colspan="3">SARROTO SADE Output Test 2</th></tr>
<tr><td>Output results</td><td colspan="2">Execution Time = 41.752 min</td></tr>
<tr><td>Final State Vector</td><td colspan="2">SMA = 42163.880091698 km
e = 0.000458195
λ =-100.501726663 deg</td></tr>
<tr><td rowspan="4">Optimized values</td><td rowspan="2">First Manoeuvre</td><td>Epoch 2017/01/01-05:47:27.462</td></tr>
<tr><td>ΔV along 0.580857981 m/s</td></tr>
<tr><td rowspan="2">Second Manoeuvre</td><td>Epoch 2017/01/01-12:00:00.000</td></tr>
<tr><td>ΔV along 0.500749728 m/s</td></tr>
</table>

Table 4.17: Output SADE Test 2.

- **GOPA**
  GOLEM and the parametric analysis have been tested as next step. Same comments made for SAPO have to be taken in account.

SARROTO GOPA Output Test 2		
Output result	Execution Time = 0.502 min Tested combinations = 1138489	
Final State Vector	SMA = 42164.115831782 km e = 0.000664015 i = 0.000300538 deg λ =-100.501287321 deg	
Optimized values	First Manoeuvre	Epoch 2017/01/01-09:00:00.000
		ΔV along 0.590000000 m/s
	Second Manoeuvre	Epoch 2017/01/01-09:15:00.000
		ΔV along 0.500000000 m/s

Table 4.18: Output GOPA Test 2.

- **GODE**
  The Differential evolution have been configured with the same features present in Tab. 4.16.

<table>
<tr><th colspan="3">SARROTO GODE Output Test 2</th></tr>
<tr><td>Output results</td><td colspan="2">Execution Time = 0.302 min</td></tr>
<tr><td>Final State Vector</td><td colspan="2">SMA = 42163.880033069 km
e = 0.000489456
λ =-100.501165853 deg</td></tr>
<tr><td rowspan="4">Optimized values</td><td rowspan="2">First Manoeuvre</td><td>Epoch 2017/01/01-06:03:07.157</td></tr>
<tr><td>ΔV along 0.581494198 m/s</td></tr>
<tr><td rowspan="2">Second Manoeuvre</td><td>Epoch 2017/01/01-12:06:22.256</td></tr>
<tr><td>ΔV along 0.507703474 m/s</td></tr>
</table>

Table 4.19: Output GODE Test 2.

Comments to test 2 In all the combination cases target state vector is achieved. This achievement is displayed in Fig. 4.12, 4.14, 4.13. Each difference value between target and final element with its related permitted tolerance is shown.

It is possible to see in Fig. 4.15 that in case of PA choice, manoeuvre events epochs obtained with both propagators are roughly located in the same time interval for all the considered combinations cases. Actually some overlapping epochs are present. About GODE and SADE same remarks can be made, even if greater separation between first and second manoeuvre occurs. Thanks to these results GOLEM theory with respect to semi-analytic model is verified and finally approved.

Difference in obtained epochs between PA and DE execution cases depends on the optimization option defined as input. Please remind that the objective of this test has been testing the PA option of using the same common searching interval for all the manoeuvres. As said at the beginning of this test, this functionality is not allowed in DE choice.

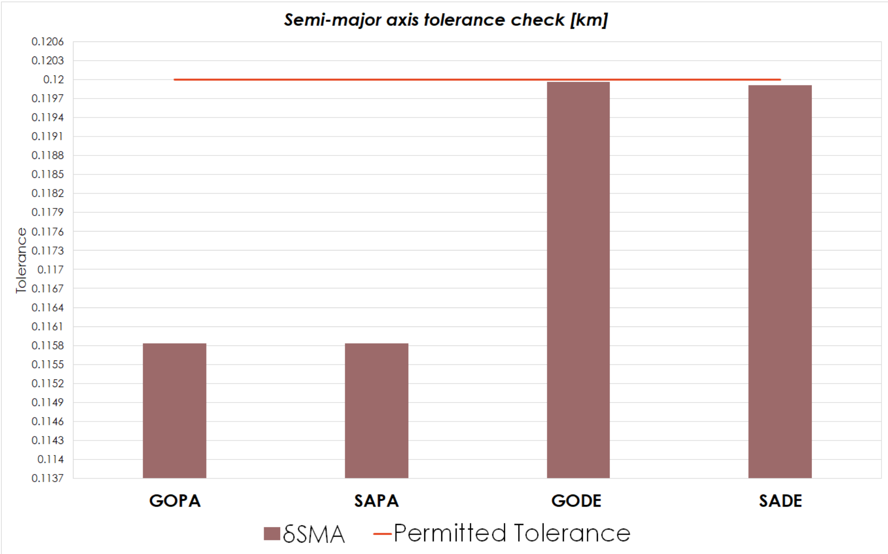

Figure 4.12: Checking tolerances for Semi-major axis-Test 2.

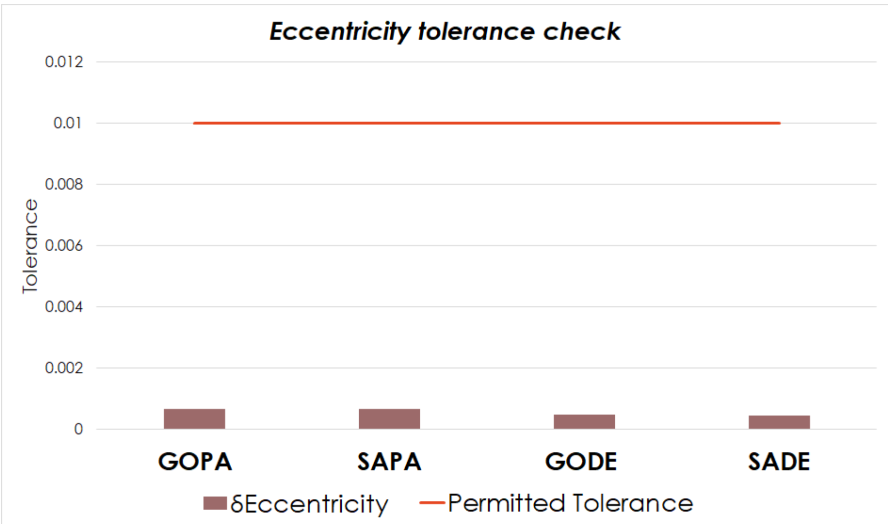

Figure 4.13: Checking tolerances for Eccentricity-Test 2.

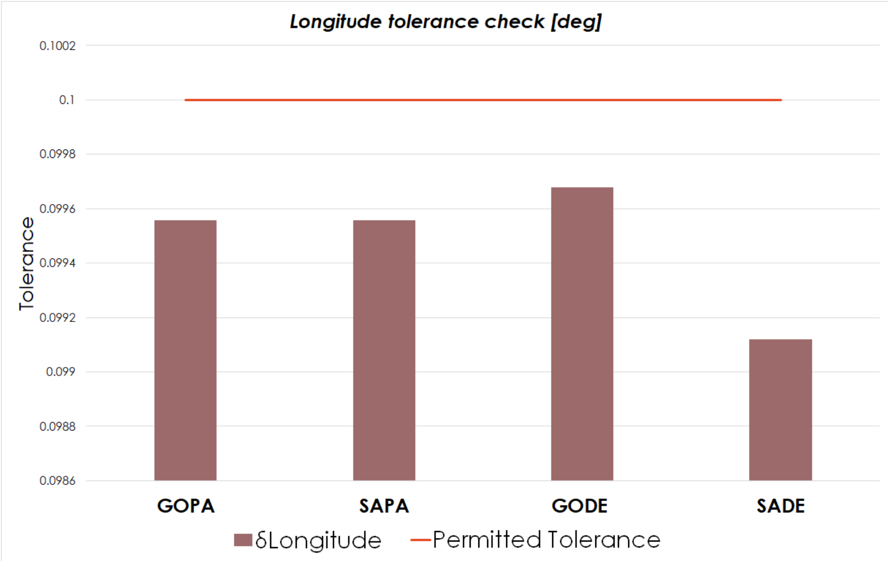

Figure 4.14: Checking tolerances for Longitude-Test 2.

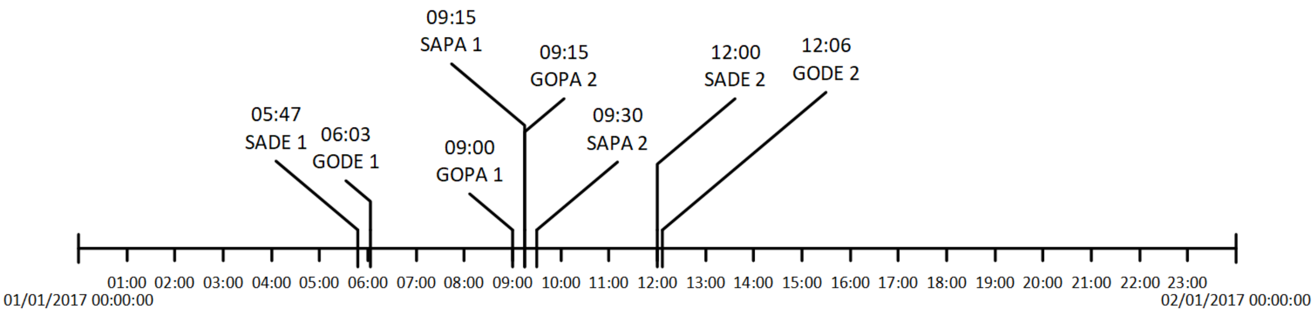

Figure 4.15: Manoeuvre events on propagation timeline Test 2.

As regards velocity, optimum results have been obtained in all cases. Indeed as it possible to see in Fig. 4.16 the results are round the ones predicted with the Hohmann transfer calculation. In particular GOPA and SAPA come up with same ΔV for both manoeuvres.

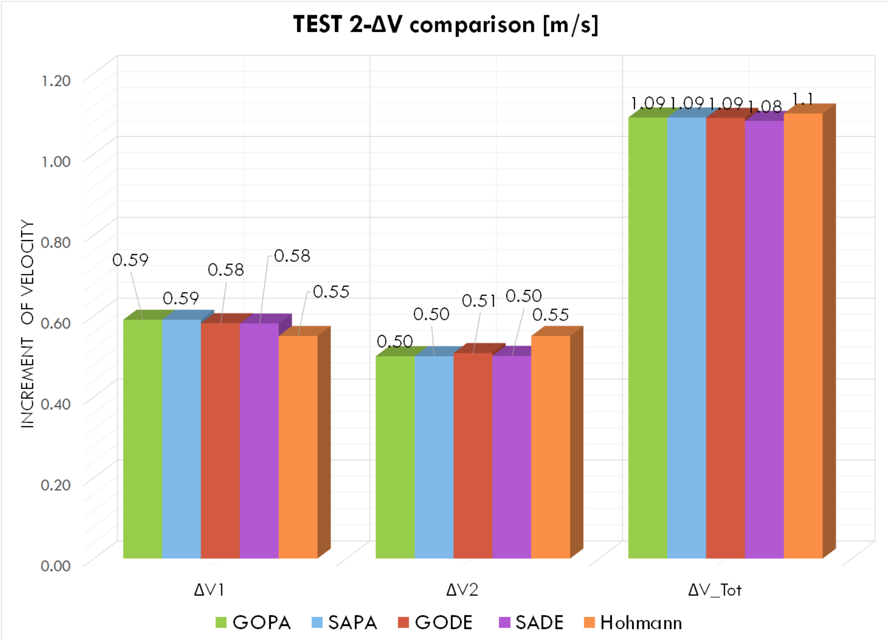

Figure 4.16: Comparison between ΔV for combination cases Test 2.

About the DE options it can be demonstrated, performing similar test but imposing 30 iterations, that results does not change. So it is not necessary to increment the iterations since after a certain number of generating vector, result presents not significant changes. On the other hand an other option may be increase the population size to come up with a more precise result.
In Fig. 4.17 comparison between SARROTO execution elapsed time in all cases is presented. As predicted, SAPA is the case with higher computational time spent. Actually, it is SAPO propagation which causes delays in execution test. Both DE and PA combined with SAPO return high elapsed time.

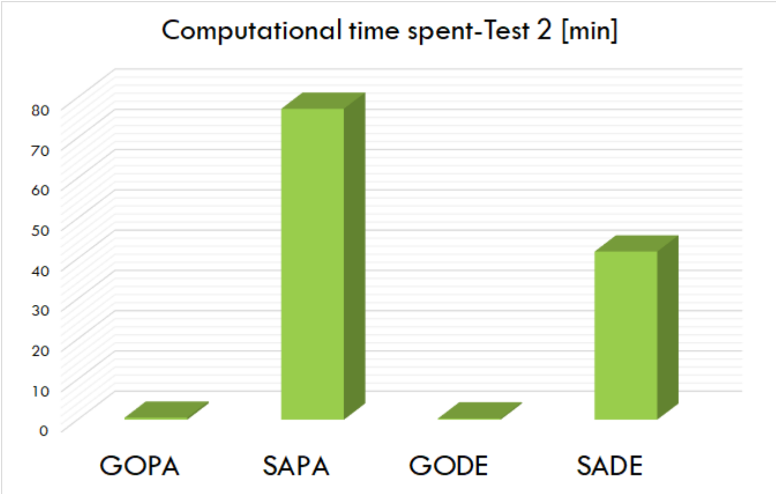

Figure 4.17: SARROTO execution time spent-Test 2.

4.4.3 TEST 3-Number of manoeuvre and velocity along track component to be optimised

This third case has been carried out using the Parametric analysis since the choice to optimise the manoeuvres number can be implemented only with this optimiser, as mentioned in section 3.1.2.3. In particular, in addition to epoch which is the forced variable to be optimised, it has been chosen to looking for the optimum along track component of velocity.
This test has been divided in two part:

- A first execution computed considering two manoeuvres events at most, performed with both SAPO and GOLEM
- A second test introducing three as maximum number of possible manoeuvre inside orbit propagation, this test has been executed only with GOLEM.

The inputs values are the ones shown in Tab. 4.20 and they are common for both the test cases mentioned above. Parameter values for the first case can be found in Tab. 4.21, while the second ones in Tab. 4.22.

<table>
<tr><th colspan="4">SARROTO Inputs Test 3</th></tr>
<tr><td>Main features</td><td colspan="3">Number of manoeuvres and
ΔV along track to be optimized</td></tr>
<tr><td rowspan="2">Initial State Vector</td><td colspan="3">Initial Epoch
2017/01/01-00:00:00:000</td></tr>
<tr><td colspan="3">SMA = 42134.0 km

e = 0.00001

i = 0.0000001 deg

Ω=0 deg

ω=0 deg

θ = 0 deg</td></tr>
<tr><td rowspan="3">Target State Vector</td><td colspan="3">Final Epoch
2017/01/02-0:00:00:000</td></tr>
<tr><td>Coordinate
Value</td><td>Weight</td><td>Tolerance</td></tr>
<tr><td>SMA = 42164.0 km

e = 0.0000000000001

λ = -106.00845364 deg</td><td>1.0

1.0

1.0</td><td>0.1 km

0.01

0.1</td></tr>
</table>

Table 4.20: Inputs Test 3.

<table>
<tr><th colspan="5">SARROTO Parameters Test 3.1</th></tr>
<tr><td>Manoeuvre definition</td><td colspan="4">Maximum Number of permitted manoeuvres and ΔV-along track to optimize</td></tr>
<tr><td rowspan="6">Manoeuvre Epoch</td><td rowspan="6">General Manoeuvre</td><td colspan="3">Radial
0.0 m/s</td></tr>
<tr><td rowspan="2">Along
To Be Optimized</td><td colspan="2">Searching interval
[0.53, 0.56] m/s</td></tr>
<tr><td colspan="2">StepSize
0.001 m/s</td></tr>
<tr><td colspan="3">Cross
0.0 m/s</td></tr>
<tr><td rowspan="2">Epoch value
To Be Optimized</td><td colspan="2">Searching interval
[2017/01/01-00:00:00.000
-2017/01/02-13:00:00.000]</td></tr>
<tr><td colspan="2">StepSize
900 s</td></tr>
</table>

Table 4.21: Parameters Test 3.1.

<table>
<tr><th colspan="4">SARROTO Parameters Test 3.2</th></tr>
<tr><td>Manoeuvre definition</td><td colspan="3">Maximum Number of permitted manoeuvres and ΔV-along track to optimize</td></tr>
<tr><td rowspan="6">Manoeuvre Epoch</td><td rowspan="6">General Manoeuvre</td><td colspan="2">Radial
0.0 m/s</td></tr>
<tr><td rowspan="2">Along
To Be Optimized</td><td>Searching interval
[0.2, 0.8] m/s</td></tr>
<tr><td>StepSize
0.01 m/s</td></tr>
<tr><td colspan="2">Cross
0.0 m/s</td></tr>
<tr><td rowspan="2">Epoch value
To Be Optimized</td><td>Searching interval
[2017/01/01-07:00:00.000
-2017/01/02-13:00:00.000]</td></tr>
<tr><td>StepSize
1800 s</td></tr>
</table>

Table 4.22: Parameters Test 3.2.

First Case

As said before in this test it has been checked the functionality of optimising the manoeuvres number. Two manoeuvres has been imposed as maximum number of events which may be found in this propagation. In order to avoid larger computational time, searching intervals have been reduced. These bounds limits have been selected considering previous test results for epoch and Hohmann transfer calculation for the velocity. In this way convergence to the target state vector may be achieved easier.

▷ **GOPA**
In Tab. 4.23 they are shown results obtained with GOLEM and Parametric analysis. It can be noticed that target is achieved for all the target elements. In the **Output results** box the number of combinations for test with one and two manoeuvres are indicated.

<table>
<tr><th colspan="3">SARROTO GOPA Output Test 3.1</th></tr>
<tr><td>Output results</td><td colspan="2">Execution Time = 1.224 min
Tested combinations for one manoeuvre =1696
Tested combinations for two manoeuvre = 2876416</td></tr>
<tr><td>Final State Vector</td><td colspan="2">SMA = 42163.906688471 km
e = 0.000700178
λ =-100.500995139 deg</td></tr>
<tr><td rowspan="4">Optimized values</td><td rowspan="2">First Manoeuvre</td><td>Epoch
2017/01/01-09:00:00.000</td></tr>
<tr><td>ΔV along
0.56000000 m/s</td></tr>
<tr><td rowspan="2">Second Manoeuvre</td><td>Epoch
2017/01/01-09:15:00.000</td></tr>
<tr><td>ΔV along
0.530000000 m/s</td></tr>
</table>

Table 4.23: Output GOPA Test 3.1.

- **SAPA**
  Output results can be found in Tab. 4.24. Number of combinations corresponds to the one already defined in GOPA since parameters option have not been changed. Target is achieved since tolerance value range is fulfilled.

<table>
<tr><th colspan="4">SARROTO SAPA Output Test 3.1</th></tr>
<tr><td>Output results</td><td colspan="3">Execution Time = 182.563 min
Tested combinations for one manoeuvre =1696
Tested combinations for two manoeuvre = 2876416</td></tr>
<tr><td>Final State Vector</td><td colspan="3">SMA = 42163.923690352 km
e = 0.000660865
λ =-100.503185811 deg</td></tr>
<tr><td rowspan="4">Optimized values</td><td rowspan="2">First Manoeuvre</td><td colspan="2">Epoch
2017/01/01-09:00:00.000</td></tr>
<tr><td colspan="2">ΔV along
0.553000000 m/s</td></tr>
<tr><td rowspan="2">Second Manoeuvre</td><td colspan="2">Epoch
2017/01/01-09:30:00.000</td></tr>
<tr><td colspan="2">ΔV along
0.530000000 m/s</td></tr>
</table>

Table 4.24: Output SAPA Test 3.1.

Second Case

In this second part of Test 3 the number of manoeuvres to be optimised is imposed to three. Since this computation requires grater amount of time with respect to the other ones, SAPO choice will not be performed.

- **GOPA**
  Output results related to this computed case can be found in Tab. 4.25. Thanks to this test it can be appreciated how the program works when this option is selected: at first it is tested the case with just one manoeuvre, and best results are saved. Then it is checked the solution with two manoeuvres, and if results are better than previous, this second are overlapped to the previous and so on until the N-number of selected manoeuvres among which looked for the best solution that in this case has been set to three.

SARROTO GOPA Output Test 3.2		
Output results	Execution Time = 60.276 min Tested combinations for one manoeuvre = 793 Tested combinations for two manoeuvre = 628849 Tested combinations for three manoeuvre = 498677257	
Final State Vector	SMA = 42163.906660706 km e = 0.000699341 λ =-100.501821848 deg	
Optimized values	First Manoeuvre	Epoch 2017/01/01-09:00:00.000
		ΔV along 0.800000000 m/s
	Second Manoeuvre	Epoch 2017/01/01-09:30:00.000
		ΔV along 0.290000000 m/s

Table 4.25: Output GOPA Test 3.2.

Comments to test 3 As it can be seen in Tab. 4.25, when three manoeuvres may be permitted, the number of total combination to be tested becomes extremely big to allow computing this test with SAPO. Performing these two test cases, it results that the best solution for this orbital transfer is the one performed with two manoeuvres with along track velocity values approximately near to the ones established with Hohmann calculation and obtained in previous tests. These ΔV are illustrated in Fig. 4.18.

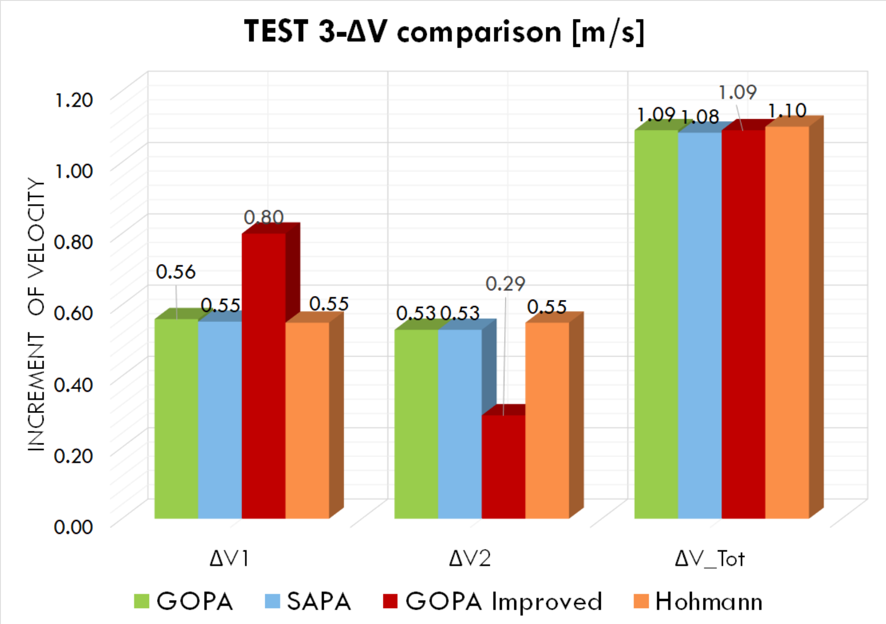

Figure 4.18: Comparison between ΔV for combination cases Test 3.

In all cases it can be noticed that propagators allow the achievment of the target. Results can be found in Fig. 4.19, 4.21, 4.20.

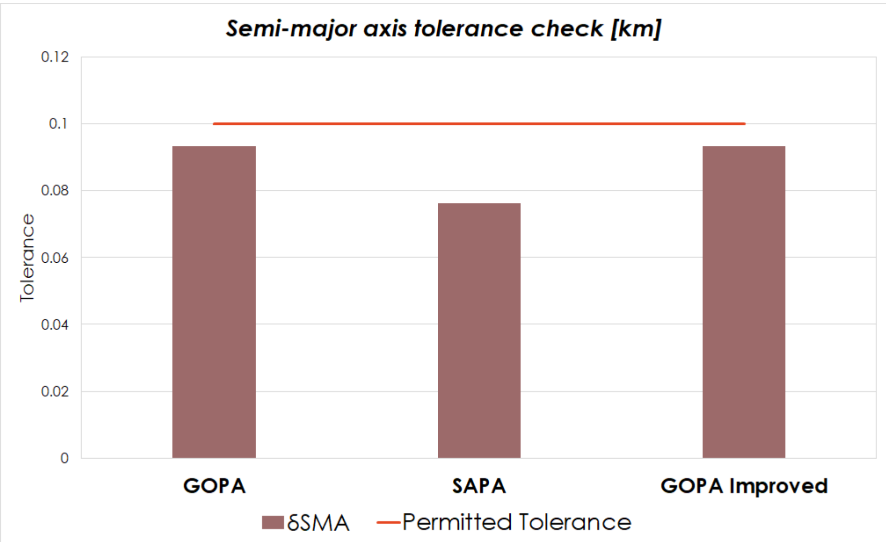

Figure 4.19: Checking tolerances for Semi-major axis-Test 3.

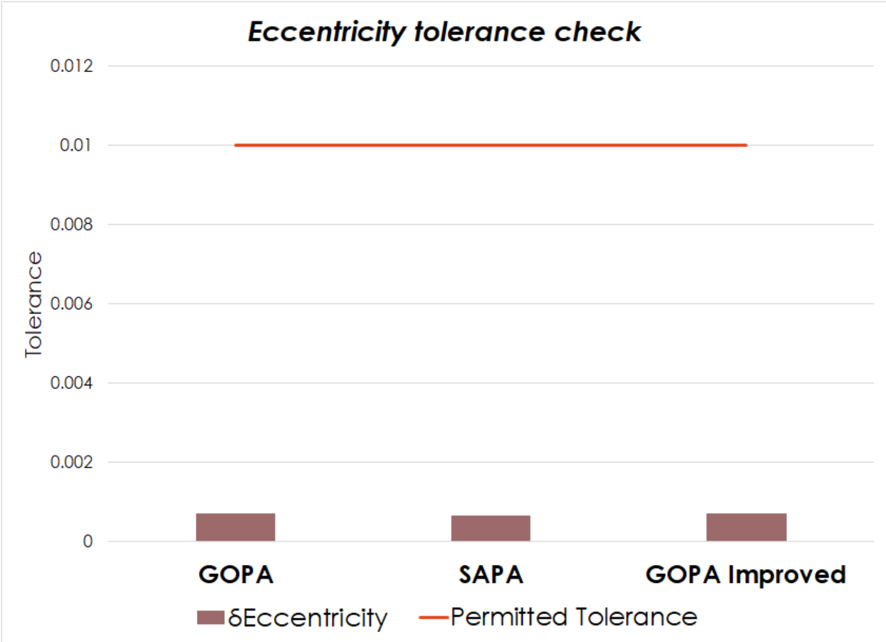

Figure 4.20: Checking tolerances for Eccentricity-Test 3.

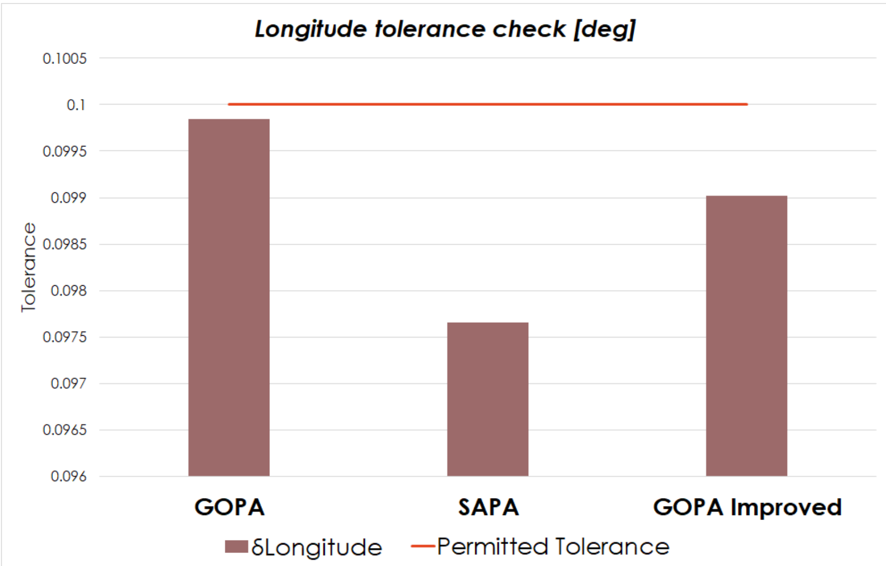

Figure 4.21: Checking tolerances for Longitude-Test 3.

In GOPA and SAPA performed with two maximum number of permitted manoeuvres, optimised manoeuvre epochs result to be almost coincident, as shown in Fig. 4.22. Actually epochs obtained with GOPA configured with three as maximum number of possible manoeuvres, indicated in the timeline with red colour, result to be located round the same values obtained with the previous two manoeuvres test. This is an excellent result which proves efficiency of PA in this optimisation option. Please remind that this option can not be chosen in case of DE.

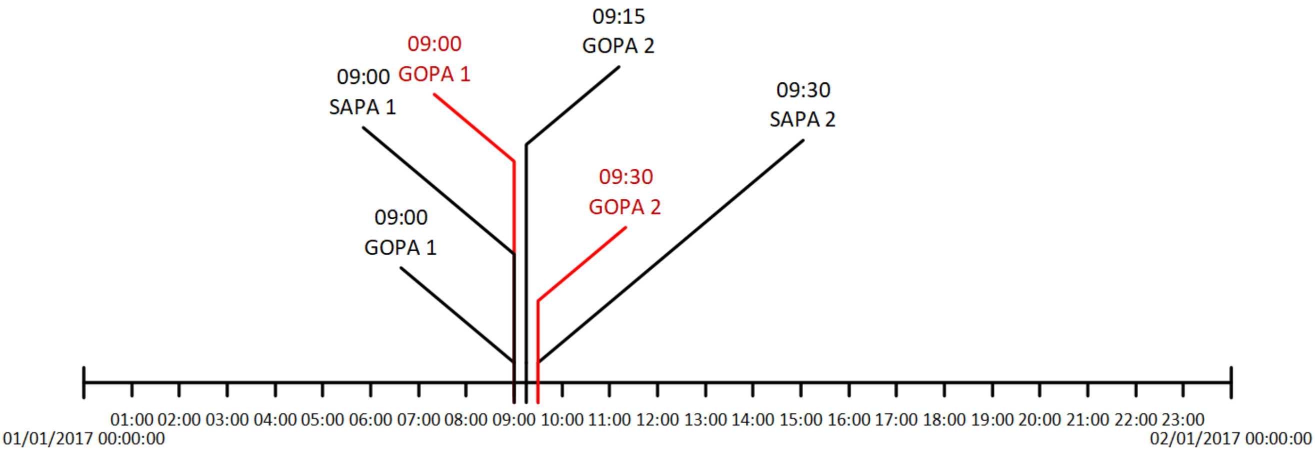

Figure 4.22: Manoeuvre events on propagation timeline Test 3.

Elapsed time for each computation case can be found in Fig. 4.23. Despite GOPA case configured with three manoeuvres as maximum, SAPO continues to affect execution more than other selected options with GOLEM as chosen propagator.

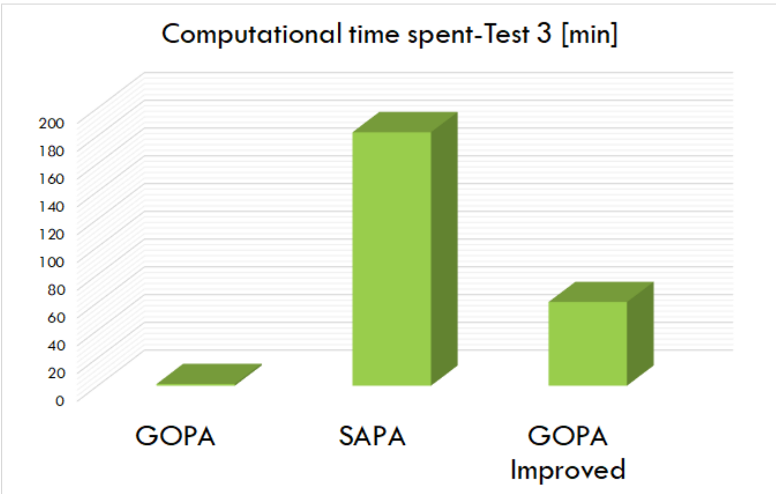

Figure 4.23: SARROTO execution time spent-Test 3.

4.4.4 Test 4-Two manoeuvre with all Keplerian elements as state vector, perfomed with GOLEM propagator.

In this test will be shown the possibility to introduce a bigger number of target elements, in particular it has been chosen to represent some elements of the Keplerian state vector as final results, even if more and more may be added to prove its functionality. Two manoeuvre will be tested with ΔV along track and epoch to be optimised. Tests will be performed will all propagators and optimisers combination.

<table>
<tr><th colspan="4">SARROTO Inputs Test 4</th></tr>
<tr><td>Main features</td><td colspan="3">Two manoeuvres
ΔV along track and epoch to be optimized
Keplerian Target elements</td></tr>
<tr><td rowspan="2">Initial State Vector</td><td colspan="3">Initial Epoch
2017/01/01-00:00:00:000</td></tr>
<tr><td colspan="3">SMA = 42134.0 km
e = 0.00001
i = 0.0000001 deg
Ω=0 deg
ω=0 deg
θ =0</td></tr>
<tr><td rowspan="3">Target State Vector</td><td colspan="3">Final Epoch
2017/01/02-0:00:00:000</td></tr>
<tr><td>Coordinate
Value</td><td>Weight</td><td>Tolerance</td></tr>
<tr><td>SMA = 42164.0 km
e = 0.0000000000001
ω = 103.422367683 deg
θ = 257.645382465deg</td><td>1.0
1.0
1.0
1.0</td><td>0.1 km
0.01
0.01 deg
0.01 deg</td></tr>
</table>

Table 4.26: Inputs Test 4.

<table>
<tr><th colspan="4">SARROTO Parameters Test 4</th></tr>
<tr><td>Manoeuvre definition</td><td colspan="3">Two manoeuvres with both ΔV-along track and epoch to be optimized</td></tr>
<tr><td rowspan="8">ΔV components value</td><td rowspan="4">First Manoeuvre</td><td colspan="2">Radial
0.0 m/s</td></tr>
<tr><td rowspan="2">Along
To Be Optimized</td><td>Searching interval
[0.3, 0.8] m/s</td></tr>
<tr><td>StepSize
0.01 m/s</td></tr>
<tr><td colspan="2">Cross
0.0 m/s</td></tr>
<tr><td rowspan="4">Second Manoeuvre</td><td colspan="2">Radial
0.0 m/s</td></tr>
<tr><td rowspan="2">Along
To Be Optimized</td><td>Searching interval
[0.3, 0.8] m/s</td></tr>
<tr><td>StepSize
0.01 m/s</td></tr>
<tr><td colspan="2">Cross
0.0 m/s</td></tr>
<tr><td rowspan="4">Manoeuvre Epoch</td><td rowspan="2">First Manoeuvre</td><td rowspan="2">Epoch value
To Be Optimized</td><td>Searching interval
[2017/01/01-00:00:00.000,
2017/01/01-12:00:00.000]</td></tr>
<tr><td>StepSize
600 s</td></tr>
<tr><td rowspan="2">Second Manoeuvre</td><td rowspan="2">Epoch value
OPtimize</td><td>Searching interval
[2017/01/01-12:00:00.000,
2017/01/02-00:00:00.000]</td></tr>
<tr><td>StepSize
600 s</td></tr>
</table>

Table 4.27: Parameters Test 4.

- **GOPA**
  In order to obtain a first approximation of the expected values at first orbit has been propagated keeping fixed the component of velocity, with values roughly near to the expected, and then once target state vector has been obtained, new computation has been performed, allowing both epoch and velocity component optimisation. In this way target state vector

has been achieved.

SARROTO GOPA Output Test 4		
Output results	Execution Time = 12.541 min Tested combinations = 13860729	
Final State Vector	SMA = 42163.901772830 km e = 0.000513294 ω = 103.425564730 deg θ = 257.642273435 deg	
Optimized values	First Manoeuvre	Epoch 2017/01/01-05:30:01.000
		ΔV along 0.790000000 m/s
	Second Manoeuvre	Epoch 2017/01/01-12:10:00.000
		ΔV along 0.30000000 m/s

Table 4.28: Output GOPA Test 4.

▷ **GODE**
The Differential evolution is configured with the same option which can be find in 4.16 but considering 30 iterations instead of 80.
Output values can be found in Tab. 4.29.

<table>
<tr><th colspan="3">SARROTO GODE Output Test 4</th></tr>
<tr><td>Output results</td><td colspan="2">Execution Time = 0.302 min</td></tr>
<tr><td>Final State Vector</td><td colspan="2">SMA = 42163.900249428 km
e = 0.000452967
ω = 103.423448023 deg
θ = 257.654810951 deg</td></tr>
<tr><td rowspan="4">Optimized values</td><td rowspan="2">First Manoeuvre</td><td>Epoch
2017/01/01-05:09:13.025</td></tr>
<tr><td>ΔV along
0.746434857 m/s</td></tr>
<tr><td rowspan="2">Second Manoeuvre</td><td>Epoch
2017/01/01-12:32:07.105</td></tr>
<tr><td>ΔV along
0.343529368 m/s</td></tr>
</table>

Table 4.29: Output GODE Test 4.

Comments to test 4 Many similarities have been found in this execution between results obtained with PA and DE.

First of all, in both cases the target state vector has been achieved. Tolerance values for each target element can be found in Fig. 4.24, 4.26, 4.25, 4.27. In addition, focusing on Fig. 4.28, it appears that optimised velocity component get roughly the same values for both DE and PA. The total transfer manoeuvre velocity for each combination case takes similar value to the Hohmann transfer. These results can be taken as an evidence of GOLEM reliability. Manoeuvre events occur within same epochs for both the optimizers as shown in Fig. 4.29.

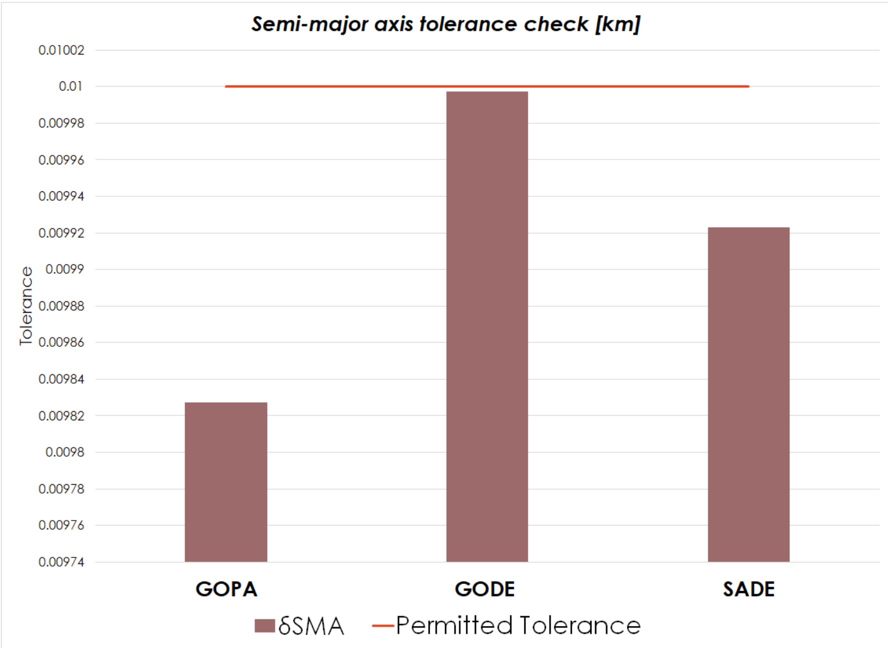

Figure 4.24: Checking tolerances for Semi-major axis-Test 4.

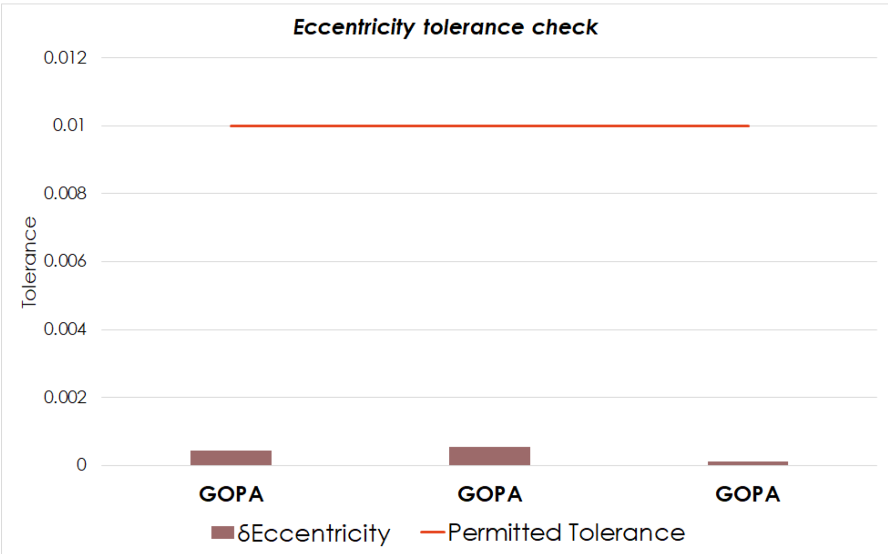

Figure 4.25: Checking tolerances for Eccentricity-Test 4.

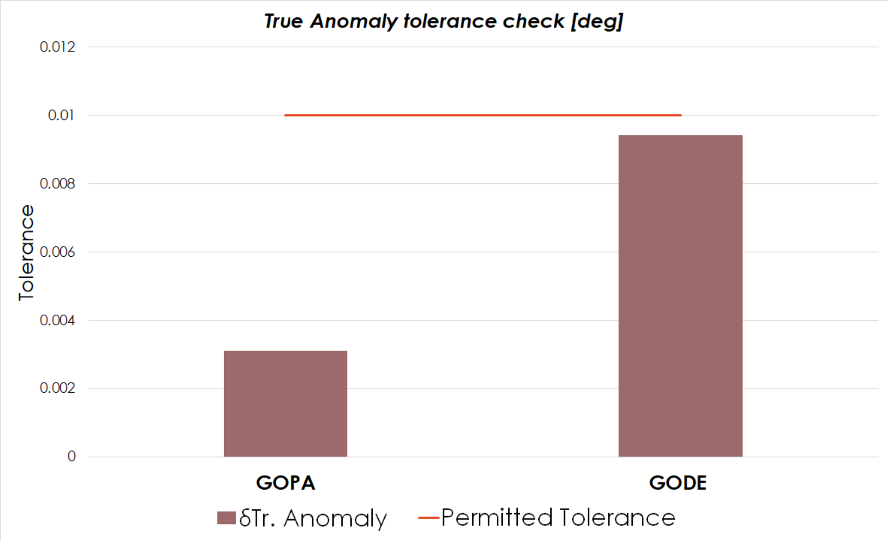

Figure 4.26: Checking tolerances for True Anomaly-Test 4.

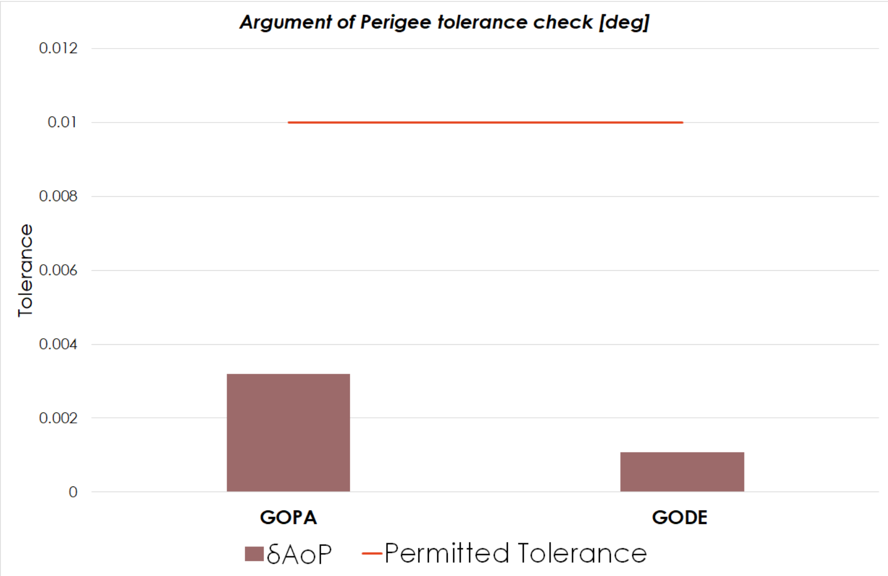

Figure 4.27: Checking tolerances for Arg. of Perigee-Test 4.

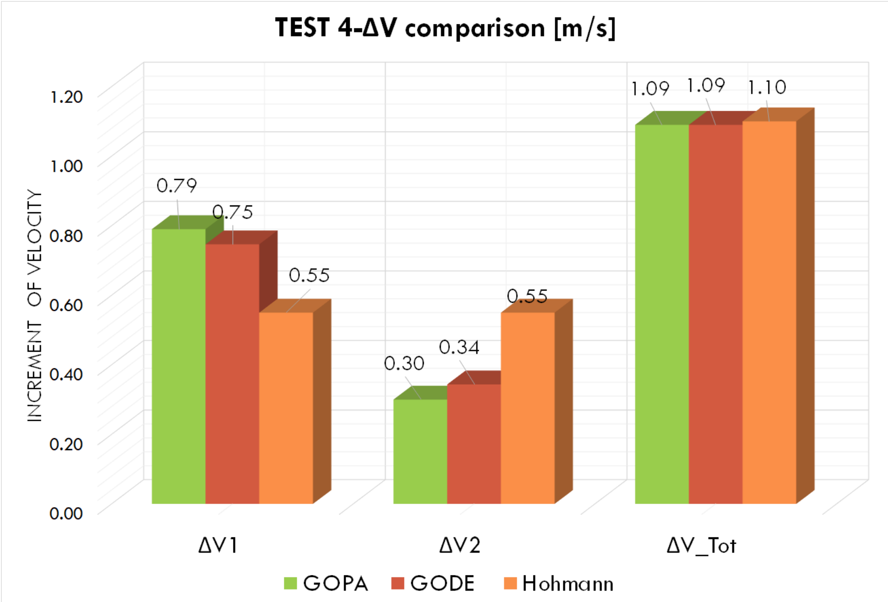

Figure 4.28: Comparison between ΔV for combination cases Test 4.

About manoeuvre epochs, it is possible to see in Fig. 4.29 that, except for few minutes of difference, the manoeuvre events occur at same epoch for both DE and PA choice.

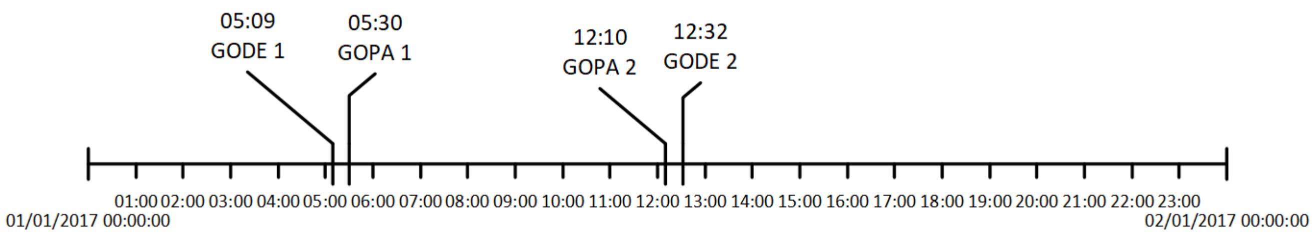

Figure 4.29: Manoeuvre events on propagation timeline Test 4.

As it could be appreciated looking at Fig. 4.30, PA choice return greater elapsed time with respect to DE solution. Clearly this depends on the different optimisation methodology between the two optimizers.

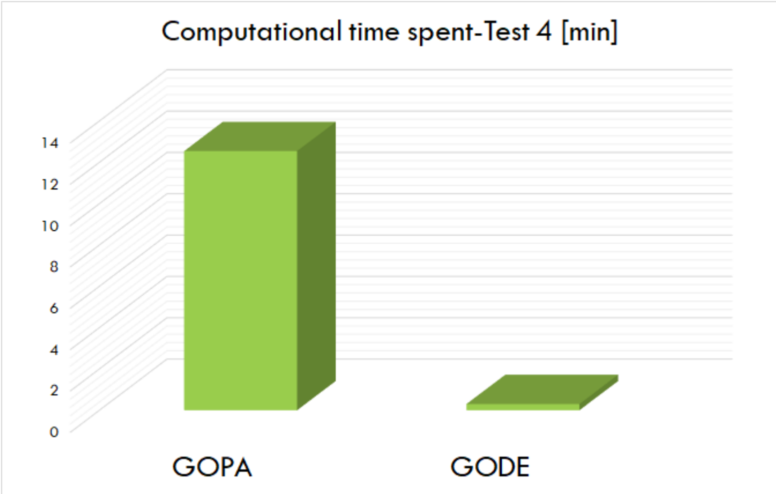

Figure 4.30: SARROTO execution time spent-Test 4.

4.4.5 Test 5-Two manoeuvre with all Keplerian elements as target state vector, performed with SAPO propagator.

This test has been computed in a similar way with respect to Test 4 but performing propagation with SAPO. This computation has been considered separately from the one explained previously since the target state vector elements values are different. New parameters values are illustrated in Tab. 4.31

<table>
<tr><th colspan="4">SARROTO Inputs Test 5</th></tr>
<tr><td>Main features</td><td colspan="3">Two manoeuvres
ΔV along track to be optimized</td></tr>
<tr><td rowspan="2">Initial State Vector</td><td colspan="3">Initial Epoch
2017/01/01-00:00:00:000</td></tr>
<tr><td colspan="3">SMA = 42134.0 km
e = 0.00001
i = 0.0000001 deg
Ω=0 deg
ω=0 deg
θ =0</td></tr>
<tr><td rowspan="6">Target State Vector</td><td colspan="3">Final Epoch
2017/01/02-0:00:00:000</td></tr>
<tr><td>Coordinate
Value</td><td>Weight</td><td>Tolerance</td></tr>
<tr><td>SMA = 42164.0 km</td><td>1.0</td><td>0.1 km</td></tr>
<tr><td>e = 0.0000000000001</td><td>1.0</td><td>0.01</td></tr>
<tr><td>ω = 58.789492160 deg</td><td>1.0</td><td>0.1 deg</td></tr>
<tr><td>θ = 212.518634984 deg</td><td>1.0</td><td>0.01 deg</td></tr>
</table>

Table 4.30: Inputs Test 5.

<table>
<tr><th colspan="5">SARROTO Parameters Test 5</th></tr>
<tr><td>Manoeuvre definition</td><td colspan="4">Two manoeuvres with both ΔV-along track and epoch to be optimized</td></tr>
<tr><td rowspan="8">ΔV components value</td><td rowspan="4">First Manoeuvre</td><td colspan="2">Radial
0.0 m/s</td></tr>
<tr><td rowspan="2">Along
To Be Optimized</td><td>Searching interval
[0.53, 0.56] m/s</td></tr>
<tr><td>StepSize
0.01 m/s</td></tr>
<tr><td colspan="2">Cross
0.0 m/s</td></tr>
<tr><td rowspan="4">Second Manoeuvre</td><td colspan="2">Radial
0.0 m/s</td></tr>
<tr><td rowspan="2">Along
To Be Optimized</td><td>Searching interval
[0.53, 0.56] m/s</td></tr>
<tr><td>StepSize
0.01 m/s</td></tr>
<tr><td colspan="2">Cross
0.0 m/s</td></tr>
<tr><td rowspan="4">Manoeuvre Epoch</td><td rowspan="2">First Manoeuvre</td><td rowspan="2">Epoch value
To Be Optimized</td><td>Searching interval
[2017/01/01/-00:00:00.000,
2017/01/01/-12:00:00.000]</td></tr>
<tr><td>StepSize
1000 s</td></tr>
<tr><td rowspan="2">Second Manoeuvre</td><td rowspan="2">Epoch value
To Be Optimized</td><td>Searching interval
[2017/01/01-12:00:00.000,
2017/01/02-00:00:00.000]</td></tr>
<tr><td>StepSize
1000 s</td></tr>
</table>

Table 4.31: Parameters Test 5.

- ▷ **SAPA**
  Parametric Analysis has been implemented together with SAPO. As it can be seen from results in Tab. 4.32, time needed for testing all the combinations corresponds to more than two hour of computation, which considering the achievement of the solution can be

considered a good result.

SARROTO SAPA Output Test 5		
Output results	Execution Time = 145.652 min Tested combinations = 1149120	
Final State Vector	SMA = 42163.920671743 km e = 0.000525891 ω = 58.789047348 deg θ = 212.512919417 deg	
Optimized values	First Manoeuvre	Epoch 2017/01/01-08:06:41.000
		ΔV along 0.548000000 m/s
	Second Manoeuvre	Epoch 2017/01/01-12:33:20.000
		ΔV along 0.535000000 m/s

Table 4.32: Output SAPA Test 5.

▷ **SADE**
In this phase SAPO will be tested together with Differential Evolution. The DE options have been kept fixed as the ones illustrated in Tab. 4.16 just changing the number of iterations to 30.

<table>
<tr><th colspan="3">SARROTO SADE Output Test 5</th></tr>
<tr><td>Output results</td><td colspan="2">Execution Time = 42.886 min</td></tr>
<tr><td>Final State Vector</td><td colspan="2">SMA = 42163.972026490 km
e = 0.000574693
ω = 58.794930448 deg
θ = 212.515824838 deg</td></tr>
<tr><td rowspan="4">Optimized values</td><td rowspan="2">First Manoeuvre</td><td>Epoch
2017/01/01-08:35:47.167</td></tr>
<tr><td>ΔV along
0.550908929 m/s</td></tr>
<tr><td rowspan="2">Second Manoeuvre</td><td>Epoch
2017/01/01-12:00:00.000</td></tr>
<tr><td>ΔV along
0.533919418 m/s</td></tr>
</table>

Table 4.33: Output SADE Test 5.

Comments to test 5 In this test it has been proved the same functionality illustrated in test 4 but considering SAPO as propagator. As it can be appreciated, results are perfectly coincident with ones obtained with Hohmann transfer, with velocity values roughly equals to the predicted, for both DE and PA test cases. Increment of velocity of each manoeuvre and total ΔV necessary for the transfer are represented through bar diagram in Fig. 4.35. Thanks to this diagram it is possible to appreciate that results obtained for each case meet the expectations when compared with Hohmann values. Achievements of target elements occurs for both DE and PA execution. Tolerances for each element are illustrated in Fig. 4.31, 4.32, 4.33. 4.34.

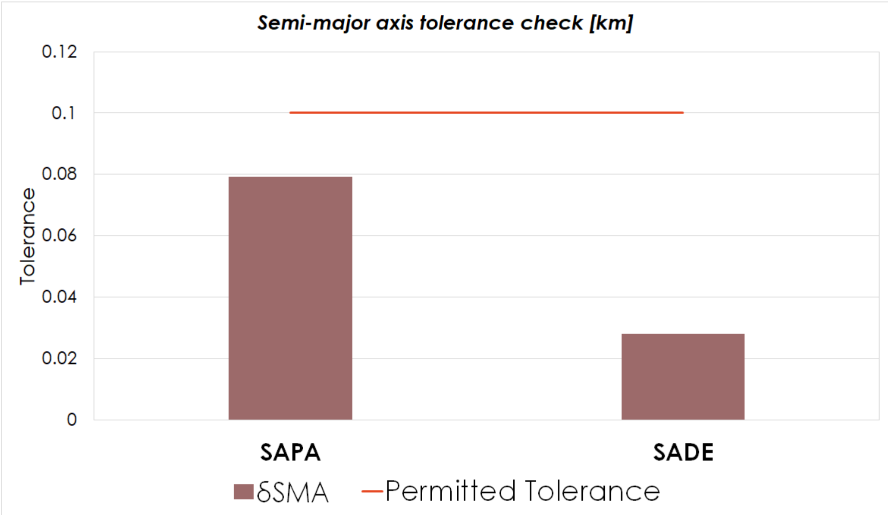

Figure 4.31: Checking tolerances for Semi-major axis-Test 5.

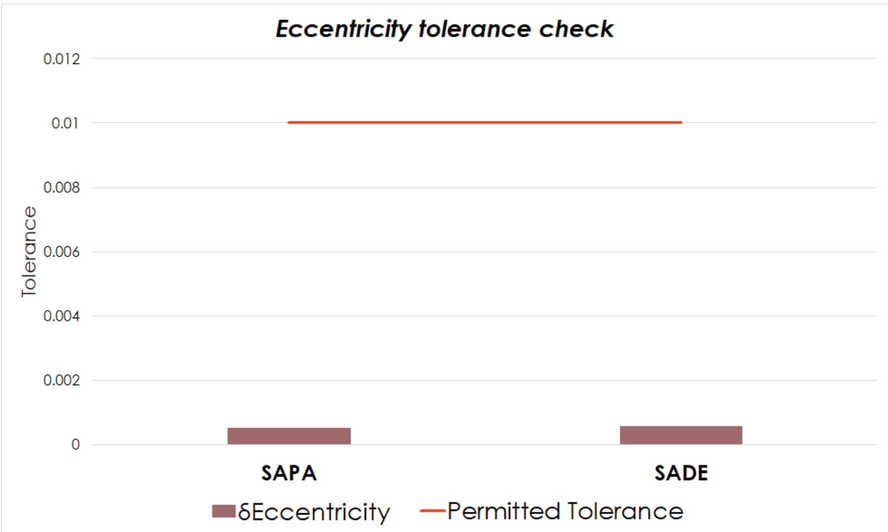

Figure 4.32: Checking tolerances for Eccentricity-Test 5.

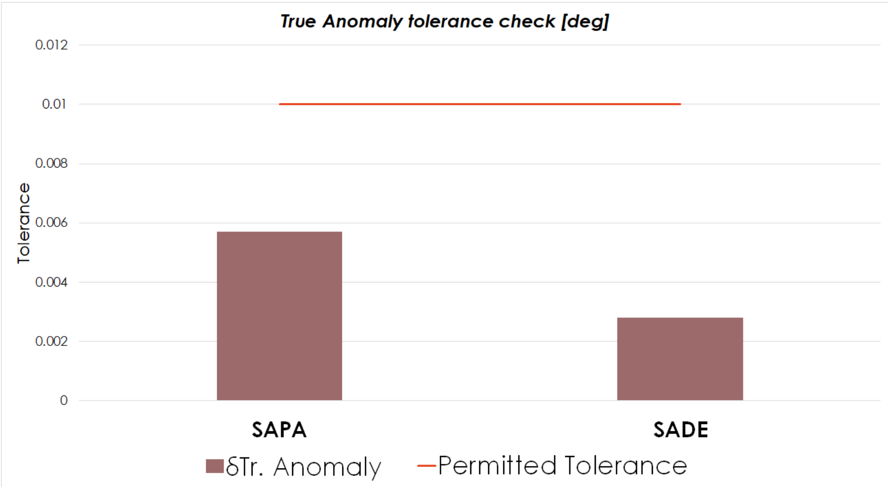

Figure 4.33: Checking tolerances for True Anomaly-Test 5.

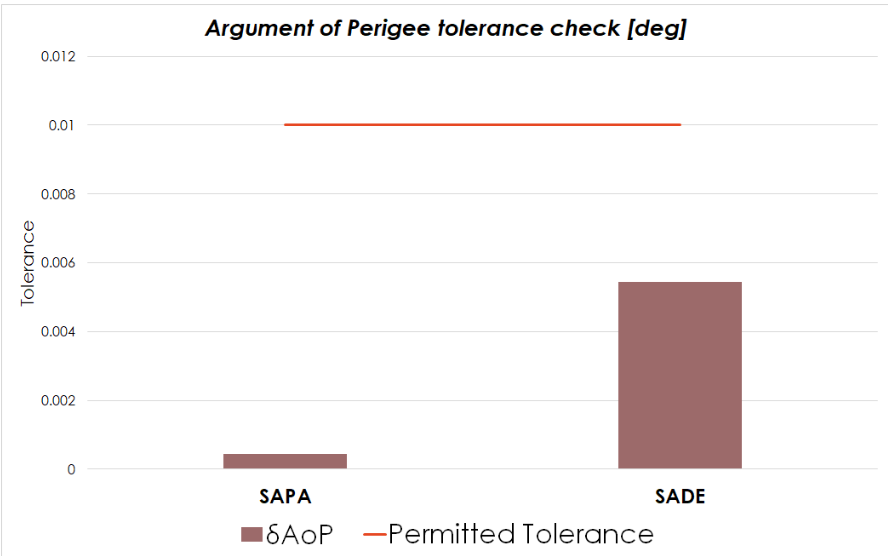

Figure 4.34: Checking tolerances for Arg. of Perigee-Test 5.

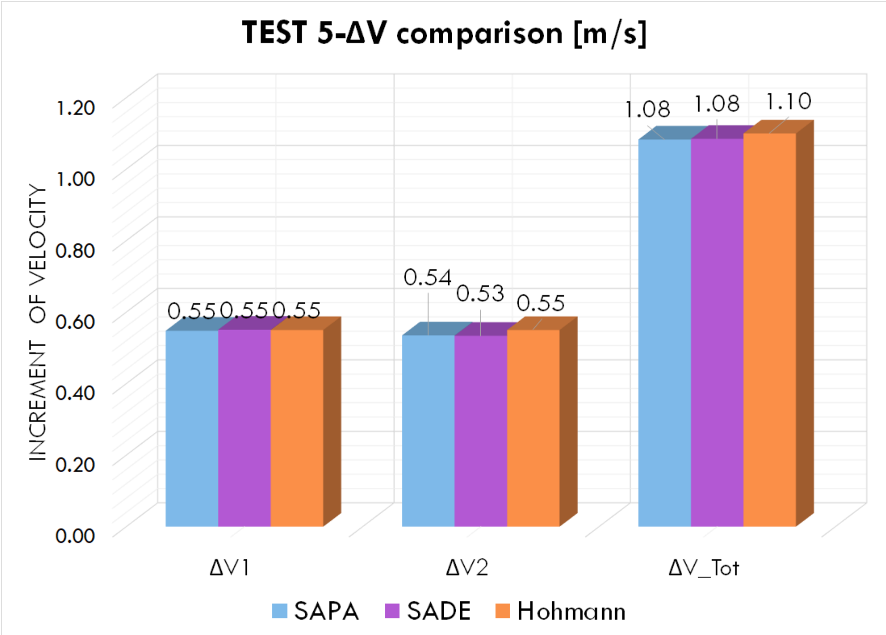

Figure 4.35: Comparison between ΔV for combination cases Test 5.

In Fig. 4.36 manoeuvre epochs distributed along the time axis are represented. It can be noticed that manoeuvre events occurs roughly at same epochs irrespective of the chosen optimisation method. Both ΔV and epochs results are evidence of PA efficiency, since its optimisation process comes up with same DE solution.

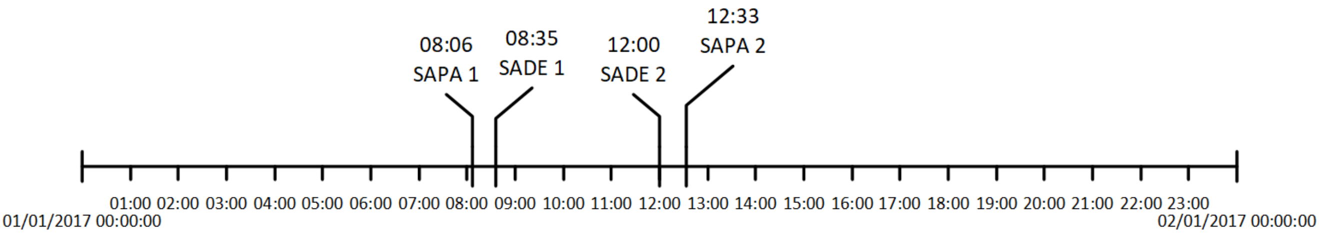

Figure 4.36: Manoeuvre events on propagation timeline Test 5.

In Fig. 4.37 elapsed time for SARROTO execution cases with SAPA and SADE options is illustrated.

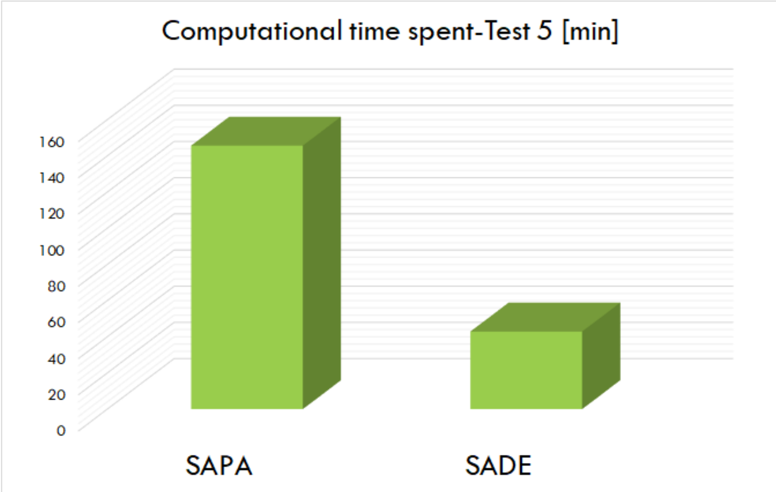

Figure 4.37: SARROTO execution time spent-Test 5.

4.5 General results

Actually, as it has been demonstrated in Test 4 and 5 of sections 4.4.4 and 4.4.5, SARROTO allows the introduction of a wide number of target elements. The choice to define a main target state vector composed of three main elements is to perform a more realistic orbit propagation. Despite what said at the beginning of this section, in Test 4 and 5 other keplerian elements, in addition to the main three defined at initial point, have been taken into account in order to show the possibility to introduce a variable number and types of state vector parameters. Related to that, true anomaly and Arg. of Perigee have been used as extra target elements since their values may change due to manoeuvre events, even if their presence does not affect the optimisation process. Readers can notice that these elements differ in assumed values depending on GOLEM or SAPO propagator selection, although both achieve the expected solution. This difference is due to their way of considering perturbation. In case of RAAN and AOP choices, perturbation introduces a numerical error which leads to different numerical values. For this reason, when eccentricity and inclination are closer or equal to zero, it is suggested not to taken into account these variables value since their reliability can not be proved.
Test performed at the end of this project prove that GOLEM allows to achieve good results when introduced into the optimisation phase. Despite its quasi-analytic definition, optimum results in term of total ΔV are achieved in all cases. Even if SAPO leads to results closer to the ones obtained with Hohmann it has to be underlined the computational time it takes to compute the solution. SAPO tests, in particular SAPA combination, have been performed introducing bigger stepsize and narrower searching interval with respect to the ones performed with GOLEM. In Fig. 4.38 total comparison between time spent for different combinations cases is represented. It can be seen that SAPO combined with PA or DE, returns greater elapsed time with respect to

other cases. On the other hand GOLEM return excellent results in shorter computational time.

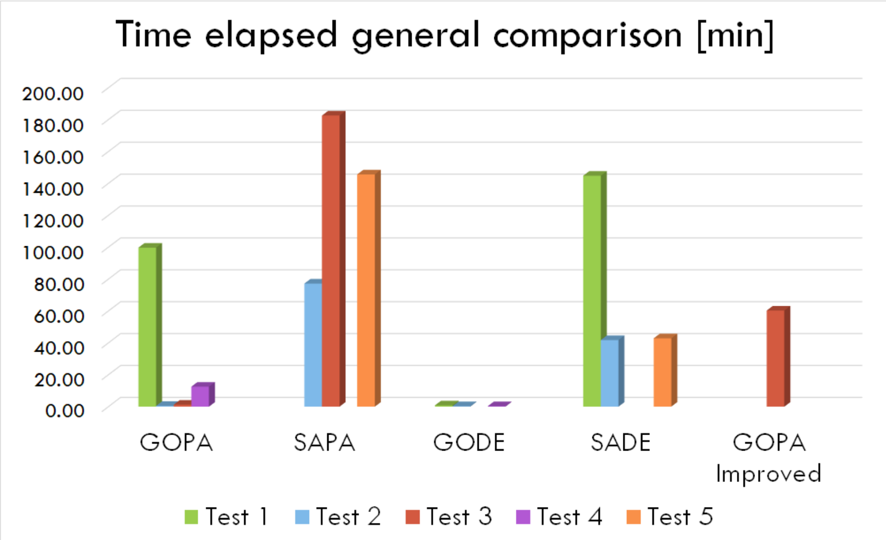

Figure 4.38: SARROTO execution time spent-General comparison.

So it can be concluded that, even if SAPO comes up with optimum results this is obtained at the expense of number of tested values. This leads to the following remark: a good solution can only be obtained with an initial guess value around which looking for.

For what concerns test 3, readers could find interesting the output results obtained with the selected tool option, consisting in optimising manoeuvres number. As it has been possible to appreciate, for both two or three as maximum number of manoeuvres permitted, it results that two events are the best optimised result.Focusing on the optimizer, these tests may prove the PA reliability compared to DE which is a tested algorithm. Even though optimisation with PA generally leads to longer spent time, it allows to perform tests cases with particular option, as the one selected in this test. In addition good results are achieved.

Remarking fundamental points, these tests have been performed in order to prove and compare the PA and DE functionalities. In particular, although the component of the velocity takes values not exactly equal to the Hohmann ones for the single increment of velocity of each manoeuvre, the total ΔV is roughly equal to the Hohmann one for all tested cases.
In addition the achievement of the target state vector has been obtained in all tests. Indeed this was the main requirement of this final test phase: the achievement of the target state vector with an optimised ΔV.

Chapter 5

From SARROTO ground level to future work

As it has been said throughout this thesis, SARROTO is a software which development has started in this internship. Its implementation involves a wide number of features, and nowadays just few of them have been introduced and tested due to the great amount of time that its code development needs.
Actually, it has to be said that the project has come up from an initial idea which has been put into practice at the beginning of the 2017 summer internship offered by GMV company. The SARROTO's basis created during this first training period can be resumed as follows

- ▷ Considering just fixed number of manoeuvres
- ▷ Implemented GOLEM analytic propagator with just two type of perturbations:
  - – Solar pressure perturbation
  - – Non-spherical Earths effects
- ▷ Just along-track manoeuvre are considered, the other ones are skipped
- ▷ Only GEO orbit are propagated
- ▷ Introducing the Differential evolution optimizer
- ▷ State vectors defined in keplerian coordinates

Indeed, at the end of this summer training, a list of possible future working had been composed, with the main aim of introducing these requirements into SARROTO within the next internship request, which correspond to this project itself. The list of first features required at the end of the summer period has been drawn up below.

- Introducing new optimizers (i.e. Parametric Analysis, NLPQLP)
- Introducing new propagators (i.e. SAPO, PROPAG)

- Introducing new constraints features such as
  - Forbidden periods of manoeuvres
  - Manoeuvres allowed only outside eclipses
  - Manoueuvres allowed only inside stations visibilities
- Optimizing numbers of manoeuvres
- Selecting coordinate system between the listed ones:
  - Cartesian coordinates
  - Equinoctial coordinates
  - Geodetic coordinates

Many progress have been made during these months, and many others have been planned for next GMV's development. All the features introduced in these months have been catalogued below.

✓ Defining the target state vector in all the possible coordinates type, from keplerian to geodetic ones.

✓ Possibility to choose the desired combination of target state vector elements: selecting their numbers and type among the list which contains most of the existing coordinates.

✓ Developing and introducing into SARROTO the parametric analysis optimiser.

✓ Transforming the GOLEM propagator from an analytic to a quasi-analytic propagator. Considering the manoeuvre effects separately from the first raised equations where these ones were included inside themselves (i.e please refer to section 2.1.2).

✓ Introducing SAPO propagator into SARROTO. Firstly it has been necessary to adapt its main code to general program usage.

✓ Possibility to optimise, in addiction to other parameters, the manoeuvres number which can be considered no more just a fixed value.

✓ Giving the possibility to the client to define just a single searching interval for all the manoeuvres which it would like to be optimised. This requirement has been introduced both in case of fixed or variable number of manoeuvres.

Continuing, it is possible to find the ones that are still lacking, indicated with a $\times$ symbol.

$\times$ Defining the initial state vector in different coordinates systems, at the moment just keplerian ones are implemented.

$\times$ Introducing numerical propagators such as Propag propagator, and others.

$\times$ Introducing other constraints features such the ones mentioned in previous optional constraints list.

× Introducing NLPQLP optimiser.

× Executing Optimisation cases taking into account SAPO propagation with mean elements.

× Improving and enriching test phase looking for best solution in larger searching interval with smaller step size.

Chapter 6

Conclusion

From the working point of view, this project has required considerable efforts due to the needing to mix theoretical knowledge with the computational one. Indeed logic processes of development have been used due to the necessity to coding with Fortran 90 language.It has to be mentioned that each option implemented during the project evolution has came up day-by-day. It has been a dynamic project, where ideas has been discusses step by step. The first initial plan has been adapted following requirements coming up from the introduction of other tools, such as the main propagators and optimizers.
Results show that it is possible, implementing the tool in reasonable way, obtaining the target state vector with the defined tolerance values. These results can be obtained both using the Parametric analysis and Differential evolution optimiser, and their solution are quite equivalent.
Despite these great achievements, it has to be said that at the moment there are some limitations which have been come up during tests execution. In particular the taken time of computation when SAPO and Parametric analysis options are selected, which as it can be seen in Tab. 4.6 and appears clearly in Fig. 4.38, is the greater among all the possible choices. Clearly this is a big disadvantage since SAPO results are the ones which best fit the client's requirements in terms of representing real case, considering more perturbations events.
On the other hand, as it has been possible to appreciate within the results chapter 4, GOPA combination comes up with results, although not perfectly accurate, close to the expected. Therefore a good option to reduce SAPA computational time may be using GOLEM, or other generic analytic propagator, with PA in order to find a first approximate solution. Then the next step may be executing the same test with SAPA looking for the best solution round the result obtained with the analytic propagator. An other strategy which could be implemented might consist in introducing a different optimizer such as NLPQLP, and to give as input to this last one the approximate results obtained with PA. This in order to avoid great execution time required for higher optimization level.

SARROTO project is far away from being finished. It is born to be a great project, whose main aim is involving many existing tools to satisfy every client requests. In fact, nowadays, many flight dynamics problems requires specific software implementation in order to be resolved, tested and compared with similar past problems, producing waste of time and resources.

Thanks to a project like the SARROTO one, all these different possible solutions could be performed without the needing of moving from a tool to an other, simply selecting the desired options.

Today SARROTO has not yet been designed to be sold to a specific user. The main aim is to improve this solution in order to satisfy future real client requirements. This will allow to make SARROTO a business product at the same level of previous GMV's FDO products such as ***focussuite*** interface, or propagators like PROPAG. To make it possible, other improvements will be made and new interfaces will be added, as mentioned in the future work in chapter 5.

Appendices

Appendix A

Astrodynamics features

A.1 Two-Body Problem

In this section it will be illustrated the theory which characterises most of the flight dynamics problems. In particular it is the start point for both the propagators mentioned and described within this thesis. In fact the two body problem is one of the principal subject of the celestial mechanics, describing, for example, the behaviour of a satellite orbiting around a planet, or two stars orbiting each other. Many others similar examples which can be found commonly in nature (i.e electrons around atomic nucleus) can be described through this theory. This problem is the one which describes the mutual gravitational attraction between two bodies. These ones are represented as two point masses, on which acts the mutual force of gravity. In order to solve the main problem, the motion of these two bodies is considered mathematically equivalent to the motion of a single body whose mass is obtained with the formula A.1 written below

$$\mu = \frac{m_1 m_2}{m_1 + m_2} \tag{A.1}$$

The mathematical steps which have to be followed in order to achieve this problem solution involve the conservation and the Newton's third and second law usage. The force on the first body, due to the interaction mentioned before, can be written as

$$\boldsymbol{F}_{1,2} = -F_{1,2}\hat{r} = -G\frac{m_1 m_2}{r^2}\hat{\boldsymbol{r}} \tag{A.2}$$

and recalling the Newton's third law, it results

$$F_{1,2} = -F_{2,1} \tag{A.3}$$

Once achieving these results, applying the Newton's second law, and considering what it has been written above it becomes

$$F_{1,2} = m_1\ddot{R}_1 \tag{A.4}$$

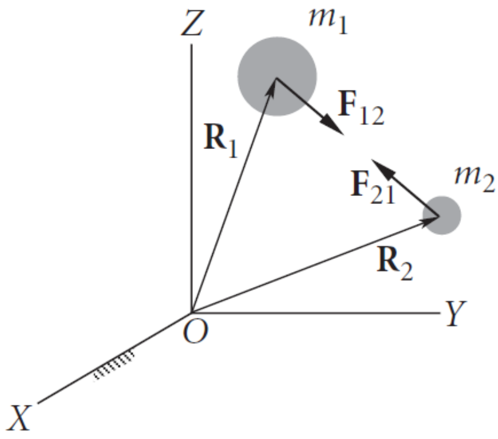

Figure A.1: Two body problem representation.

$$-G\frac{m_1 m_2}{r^2}\hat{\boldsymbol{r}} = m_1 \ddot{R}_1 \tag{A.5}$$

and for Eq. A.5

$$m_1 \ddot{R}_1 + m_2 \ddot{R}_2 = 0 \tag{A.6}$$

It has to be noted that Eq. A.5 can be written in the same way for the second body. So, multiply the Eq. A.5 by m_1 and the similar one obtained for the second body by m_1, computing some mathematical steps, it is possible to get the second differential equation which governs the motion of m_2 with respect to m_1

$$\ddot{\boldsymbol{r}} = \frac{\mu}{r^3}\boldsymbol{r} \tag{A.7}$$

where μ is the gravitational parameter which in this case corresponds to

$$\mu = G(m_1 + m_2) \tag{A.8}$$

with **G** the Universal gravitational constant. This equation is solved considering two separated body problems, with a trivial solution for the first one, while the approach for the second one regards solving the motion of one particle in an external potential.

A.2 Satellite state representation

In order to define the *state* of a satellite in space it is necessary to refers to some variables since a generic orbit needs six elements to be identified. When refers to Cartesian coordinates type, these elements are position and velocity and theirs three-dimensional components, which together represent the ***State Vector***. On the other hand the state can be defined with a set of parameters. The most common and used ones are called orbital, Keplerian or Classic elements of the orbit. Apart from these, other different element sets have been developed such as equinoctial and geodetic orbit elements.

A.2.1 Keplerian orbital elements

To specify the complete motion of a satellite, Keplerian elements known also as osculating or classic elements, can be utilised. These ones define the orbit in terms of scalar magnitude and angular representations, called *Euler's angle*. The mentioned orbital elements are the following:

- $\boldsymbol{a}$, Semi-Major Axis, which defines the size of the orbit;
- $\boldsymbol{e}$, Eccentricity, defining the shape of the orbit;
- $\boldsymbol{i}$, Inclination, which indicates the orientation of the orbit respect to the Earth's equator
- $\boldsymbol{\Omega}$, Right Ascension of the Ascending Node, clarifies the location of the ascending and descending orbit locations with respect to the Earth's equatorial plane;
- $\boldsymbol{\omega}$, Argument of Perigee, defines where the low point, perigee, of the orbit is with respect to the Earth's surface;
- $\boldsymbol{\theta}$, True Anomaly, specifies where the satellite is within the orbit with respect to perigee.

In some cases the semi major axis and the true anomaly can be substituted respectively with angular momentum, $\boldsymbol{h}$, and the mean anomaly, $\boldsymbol{M}$. In Fig. A.2 these elements are shown. An other element which can be defined when speaking about orbital elements is the mean motion $\boldsymbol{n}$, which describes the satellite's average angular rate of the motion over the orbit, and can be calculated as follows

$$n = \frac{d}{P}$$

where $\boldsymbol{d}$ is the quantity of time in one day, in seconds, and $\boldsymbol{P}$, the Earth's orbital period calculated as

$$P = 2\pi\sqrt{\frac{a^3}{\mu}}$$

A.2.2 Cartesian State Vector

The Cartesian Coordinates, also known as state vector, describe the position and motion of satellites using respectively $\boldsymbol{x, y, z}$ and $\boldsymbol{V_X, V_Y, V_Z}$, which are components of velocity. In addition, it is noteworthy, that for GMV's coding library the *Arc*, which is the total amount of the state vectors plus related time epoch along the orbit, coming out from the computation, as to be described in Cartesian Coordinates.

A.2.3 Equinoctial orbital elements

The Equinoctial elements have been developed due to some singularities coming up from the definition of special orbit cases using classical elements, mentioned above. This type of coordinates uses the Earth's center as origin and the plane of the satellite's orbit as reference plane. These type elements are the following:

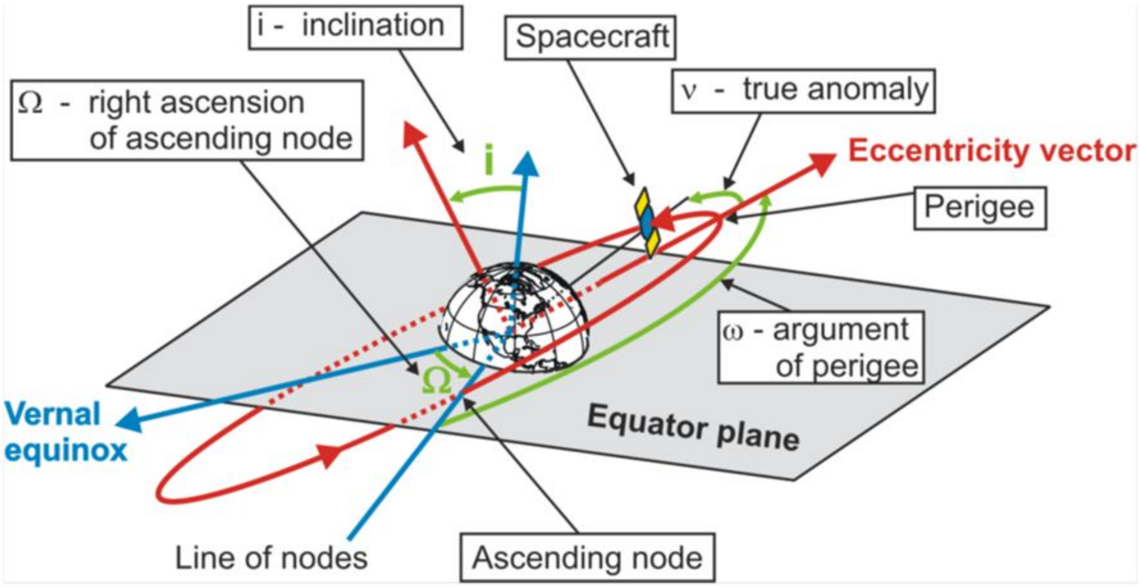

Figure A.2: Classical orbit elements

- ***a***, Semi-major axis, defining the orbit size, same definition of the Keplerian semi-major axis;
- ***h, k***, two elements which describes the shape of the satellite's orbit and the position of perigee, defined as components of the eccentricity vector;
- ***p, q***, specifies the orientation of the satellite's orbit plane, defined as components of the ascending node vectors;
- λ, the Mean Longitude, specifies a satellite's position within its orbit at epoch and equals the sum of the classical Right Ascension of the Ascending Node, Argument of Perigee, and Mean Anomaly.

Actually there are two sets of equinoctial orbital elements: the direct set and the retrograde one which can be applied respectively to direct and retrograde satellites even if, for non-equatorial satellites, the direct one can be used for retrograde satellites and vice-versa. On the other hand, for equatorial satellites, the element set has to be accurately chosen in order to avoid singularities, in fact for equatorial satellites in case of direct elements

$$\lim_{i \to \pi} \sqrt{p^2 + q^2} = \infty$$

while for retrograde ones

$$\lim_{i \to 0} \sqrt{p^2 + q^2} = \infty$$

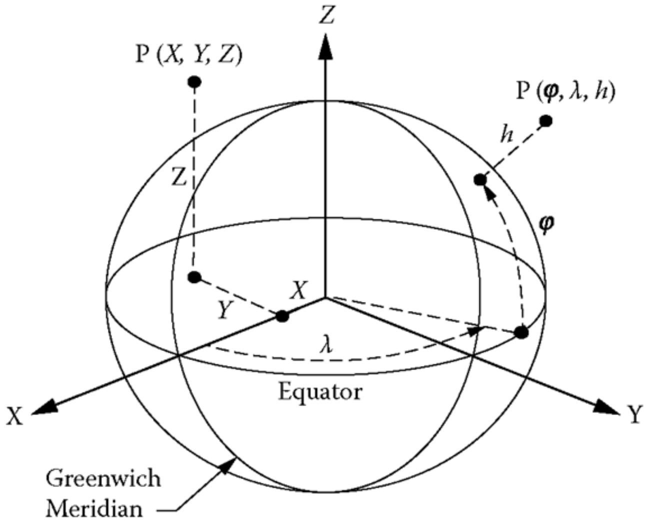

Figure A.3: Geodetic coordinate type system

A.2.4 Geodetic coordinate system

The geodetic coordinate system, also known as ***geodetic datum***, is composed from the following parameters

- Altitude, measured along an outward normal to the surface of the reference ellipsoid;
- Latitude, which is the angle between the normal to the reference ellipsoid and the equatorial plane;
- Longitude, which corresponds to the angle between the projection of the position vector in the equatorial plane and the prime meridian

Differently from the other coordinate system, components taken into account are just three. In fact the other three correspond to the rate of change of these parameters but are not considered in these calculations.

A.3 Coordinates transformations

A.3.1 Conversion from Keplerian elements to Cartesian Coordinates

The algorithm used to convert Keplerian elements in order to obtain the Cartesian State vector is the following:

1. Compute Mean Anomaly using the mean motion parameter seen before and difference between epoch and initial time
$$M = n \cdot (t - T)$$

2. Evaluate the Eccentric Anomaly solving the Kepler's equation with an appropriate method numerically
$$M = EA - e \cdot \sin EA$$
3. Calculate true anomaly
$$\tan\frac{\theta}{2} = \sqrt{\frac{1+e}{1-e}}\tan\frac{EA}{2}$$
4. Compute the radius, **r**, the distance to the central body, using the eccentric anomaly
$$r = a \cdot (1 - e\cos EA)$$
5. Obtain the position and velocity vector, respectively **o** and $\dot{\boldsymbol{o}}$ in the orbital frame
$$o(t) = r\begin{pmatrix}\cos\theta \\ \sin\theta \\ 0\end{pmatrix} \qquad \dot{o}(t) = \frac{\sqrt{\mu a}}{r}\begin{pmatrix}-\sin EA \\ \sqrt{1-e^2}\cos EA \\ 0\end{pmatrix}$$
6. Convert **o** and $\dot{\boldsymbol{o}}$ in the orbital frame to the inertial frame, respectively, **r** and $\dot{\boldsymbol{r}}$
$$r(t) = \begin{pmatrix} o_x(\cos\omega\cos\Omega - \sin\omega\cos i\sin\Omega) - o_y(\sin\omega\cos\Omega + \cos\omega\cos i\sin\Omega) \\ o_x(\cos\omega\sin\Omega + \sin\omega\cos i\cos\Omega) + o_y(-\sin\omega\sin\Omega + \cos\omega\cos i\cos\Omega) \\ o_x(\sin\omega\sin i) + o_y(\cos\omega\sin i) \end{pmatrix}$$

$$\dot{r}(t) = \begin{pmatrix} \dot{o_x}(\cos\omega\cos\Omega - \sin\omega\cos i\sin\Omega) - \dot{o_y}(\sin\omega\cos\Omega + \cos\omega\cos i\sin\Omega) \\ \dot{o_x}(\cos\omega\sin\Omega + \sin\omega\cos i\cos\Omega) + \dot{o_y}(-\sin\omega\sin\Omega + \cos\omega\cos i\cos\Omega) \\ \dot{o_x}(\sin\omega\sin i) + o_y(\cos\omega\sin i) \end{pmatrix}$$

A.3.2 Conversion from Keplerian elements to Equinoctial Coordinates

As said before, it can be useful to convert Keplerian elements to Equinoctial coordinates type. It is sufficient to know keplerian elements $a,\ e,\ i, \Omega,\ \omega,\ \theta$ values in order to execute the analytic transformation. These quantities can be obtained with the following mathematical expression

$$\begin{aligned} h &= e \cdot \sin\tilde{\omega} \\ k &= e \cdot \cos\tilde{\omega} \\ p &= \left[\tan\left(\frac{i}{2}\right)\right]^I \cdot \sin(I\Omega) \\ q &= \left[\tan\left(\frac{i}{2}\right)\right]^I \cdot \cos(I\Omega) \\ \Lambda &= M + \tilde{\omega} \end{aligned}$$

where $\tilde{\omega} = \omega + I\Omega$. The quantity **I** is called for retrograde factor and has two values

$$I = \begin{cases} +1 & \text{for the direct equinoctial elements} \\ -1 & \text{for the retrograde equinoctial elements} \end{cases}$$

A.3.3 Conversion from Keplerian elements to Geodetic Coordinates

Converting keplerian elements to Geodetic ones requires more steps than other transformation type. In fact an intermediate step which involves Cartesian coordinates it is necessary. The transformation process is:

$$Keplerian \longrightarrow Cartesian \longrightarrow Geodetic$$

It has to be mentioned that in addition to this it is required a reference frame transformation from an inertial one to Earth-Fixed. The first transformation from Keplerian to Cartesian has been mentioned in previous paragraph. So assuming known Cartesian values, the second transformation which interests the conversion between Cartesian and geodetic is cumbersome and requires different mathematical steps involved computational methods of resolutions. Among the three variables which are obtained the simplest one to evaluate is the geodetic longitude:

$$\lambda = \arctan \frac{x_G}{y_G}$$

The other two variables are calculated utilising the Newton's iteration to obtain the correct result. At first it is necessary to find the projection of a point h distance away on the reference ellipsoid. The mathematical steps which have to be evaluated are the following, where p_E represents the distance shows in the simplified meridian ellipse in Fig. A.5.

$$p_E = \sqrt{x_E^2 + y_E^2}$$
$$z_E = \frac{b^2}{a^2} N \cdot \sin\phi$$

Through some mathematical steps, which will be not investigated, it is possible to get the system of nonlinear equations which has as main argument the variable z_E, p_E.

$$\begin{cases} f_1(p_E, z_E) = (p_E - p_G)Hz_E + (z_E - z_G)Gp_E = 0 \\ f_2(p_E, z_E) = Gp_E^2 + Hz_E^2 - K = 0 \end{cases}$$

This system can be solved using the Newton's method, which is a iterative process necessary to obtain solution when a system of nonlinear equation is present. Once calculated these two values, the second step is calculating the latitude and the height, respectively indicated as ϕ and h. The latitude can be obtained from the following equation

$$\tan\phi = \frac{a^2 z_E}{b^2 p_E} = H^2 \frac{z_E}{p_E} \tag{A.9}$$

and the height is obtained utilising the Euclidean formula which calculates the distance between two points.

$$h = \sqrt{(p_E - p_G)^2 + (z_E - z_G)^2} \tag{A.10}$$

The Eq. A.10 is valid when $(p_G + |z_G|) < (p_E + |z_E|)$, if this condition is not verified then $h = -h$.

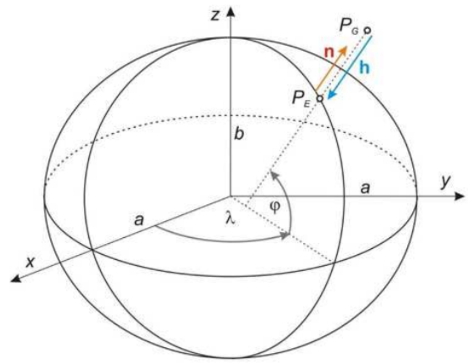

Figure A.4: Geodetic coordinates for ellipsoid system

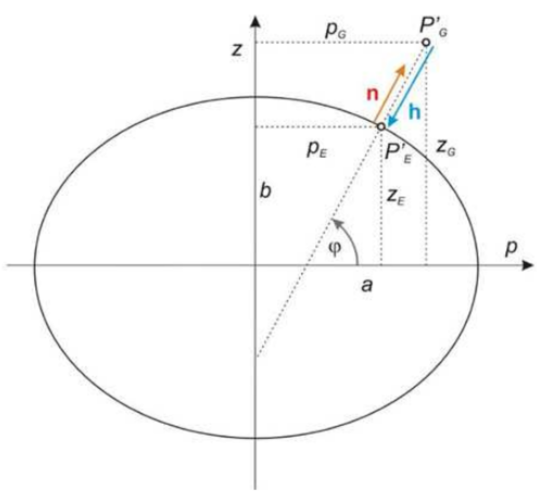

Figure A.5: Geodetic conversion for meriadian ellipsoid

Appendix B

Orbital Manoeuvres Definition

Speaking about geostationary spacecraft it is necessary to discuss about orbit manoeuvres. Within the thesis, the satellite increment of velocity due to manoeuvre event has been mentioned many times.
First of all, manoeuvres in GEO orbit are used in different situations:

- to allow the insertion of the satellite in the geostationary orbit. This process is called *station acquisition* ;
- to contrast perturbations which act on the satellite during its life and which lead to a non geostationary orbit, the *Station Keeping* manoeuvre;
- to manoeuvre the satellite in a orbit fewer kilometres under or above the geostationary one at the end of its life. This manoeuvre is defining *Re-orbiting* process.

For a satellite in a geostationary orbit thrusts can be performed both in the orthogonal direction to orbit plane and in the tangential direction, but it has to be said that thrusts in the radial direction is rarely used. In Fig.B.1 it is represented how velocity components of spacecraft during a manoeuvre can be splitted.
In particular the two manoeuvres which commonly take place are the orthogonal and tangential one. The first one is named inclination or North-South manoeuvre, and it is a out-of-lane thrust. The other one is known as west-east or longitude manoeuvre, also called along-track thrust. This is the manouevre type which has been mentioned in this project and to which usually refers when speaking about transference between two coplanar orbit.

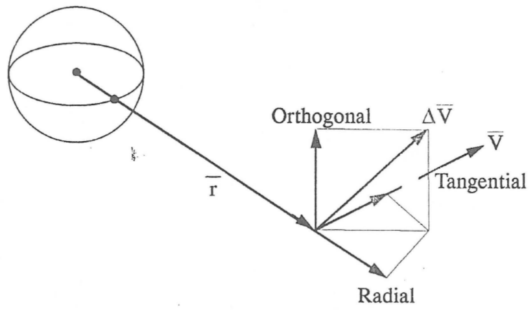

Figure B.1: Thrust Manoeuvre with velocity components with respect to the orbital plane.

In fact these manoeuvres effects can be resumed as follows:

- The inclination manoeuvre is the one used when a changing in the plane of the orbit is required. In mathematical terms means that the inclination vector and the keplerian elements i, Ω can be changed with this manoeuvre type.
  A classical representation of this manouevre can be found in Fig. B.2

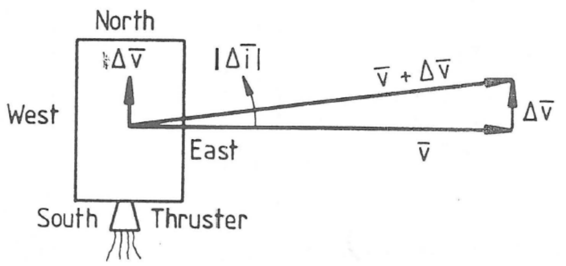

Figure B.2: North-South Thrust Manoeuvre.

- The East-West Manoeuvre is used when a change in the longitude drift and orbital eccentricity is required. As previously mentioned, this is the type of manoeuvre which is used when a coplanar orbit would be obtained. This manoeuvre allows changes in semi-major axis and eccentricity orbital values. In particular combination of two of these, defined as a multiple in-plane thrust sequence, is performed when drift rate has to be changed without introducing changes in the eccentricity vector. This multiple sequence is also called two-burn longitude manoeuvre. Performing subsequently two tangential manoeuvres like the ones above described allows to obtain a transfer from a circular orbit to an other circular orbit with major or minor height. This type of multiple set of manoeuvres, as shown in Fig. B.3 is commonly known as Hohmann transfer, and it will be better explained in next section.

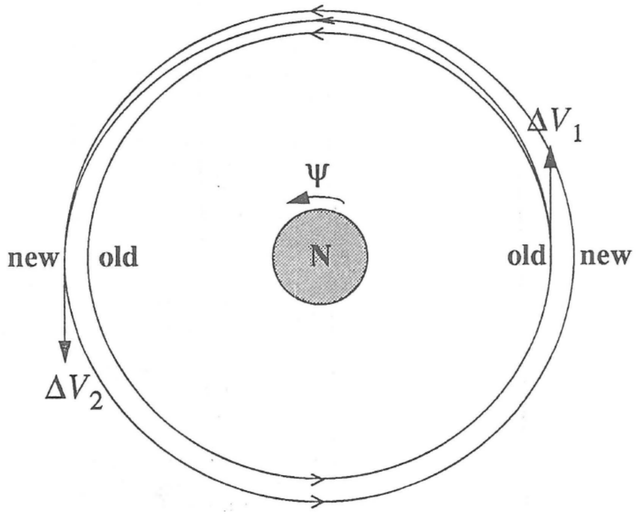

Figure B.3: Multiple East-West Thrust Manoeuvres.

B.1 Hohmann transfers

One of the most required mathematical solution is how to transfer from one orbit to an other one or to a interplanetary trajectory. In order to compute this manoeuvre it is necessary to assume that spacecraft velocity changes. This is done generating an impulse, assumed to be instantaneous, with the spacecraft rockets. The main aim, when performing a manoeuvre, is to achieve the desired position with the minimum ΔV, so as the better energy efficiency.

In this sense, one of the existing most efficient transferring manoeuvre is the Hohmann transfer, developed by Hohmann in 1925. It is a two-impulse manoeuvre which allows transferring a spacecraft between two coplanar circular orbit which share a common focus. This can be defined as an elliptical orbit, represented in Fig.B.4 with red color, tangent to both circular orbits. Actually, it is known that real orbit are not perfectly circular. For this is reason it necessary to generalize the definition and enrich this last one introducing transfer manoeuvre between elliptical orbits that are coaxiales. This transference works in such a way that starting from an initial point, defined in Fig. B.4 with *1st burn*, it is required an increment of velocity Δv_A to boost the vehicle in the elliptical trajectory. Once the point defined as *2st burn* is achieved, an other velocity increment Δv_B is required to place the spacecraft in this second orbit. The total Δv necessary to perform this transference is obtained added the first Δv_A to the second one.
In case of circular orbit the constant orbital speed can be calculated as follows in Eq. B.1

$$v = \sqrt{\frac{\mu_{Earth}}{Radius}} \tag{B.1}$$

Within this thesis it is assumed that the first orbit has a radius which is inferior to the geostationary orbit height. The radius itself corresponds to the satellite's initial position represented by the

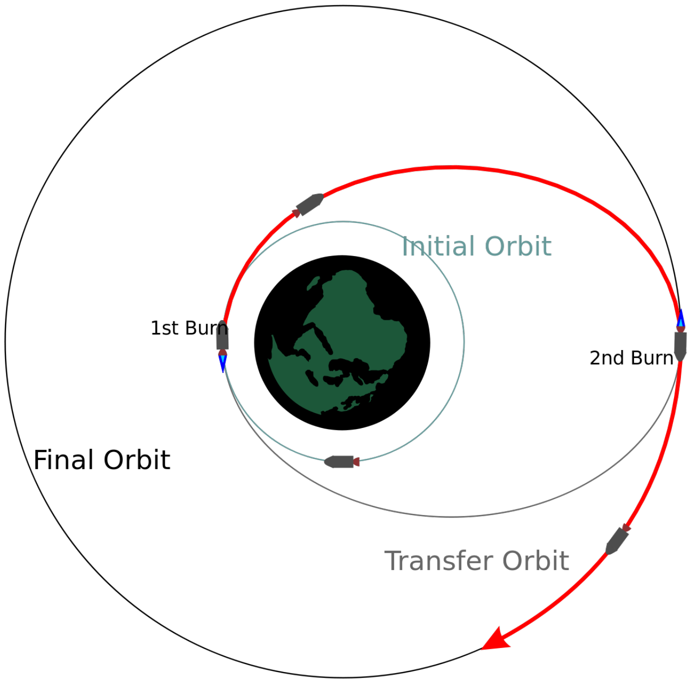

Figure B.4: Hohmann transfers model.

semi-major axis value and renamed as R_{ISV}

$$V_{<geo} = \sqrt{\frac{\mu_{Earth}}{R_{ISV}}} \tag{B.2}$$

where $R_{ISV} = SMA_{ISV} + R_E = 42134\ km$. Computing this calculation it results a velocity equal to $\Delta V = 0.55\ m/s$. It is expected to obtain a value from the optimisation which approximately is similar to this, and as it can be verified in chapter 4 this requirement is checked.

Appendix C

focussuite GMV's software

focussuite can be resumed in few words as *the ultimate flight dynamics solution for satellite operation*, as described in [17]. It arises from the GMV's desired of satisfying customers needs and requirements as it regards the flight dynamics field of interest. Its functionalities can be found listened above:

▷ Keen on obtaining solutions when multi-mission, multi-satellite flight dynamics problems for flight dynamics satellite control are setted, offering reliability, flexibility and user friendliness.

▷ Offering a wide collection of plug-and-play components which give the customer the possibility to satisfy his requests.

▷ Providing an opened framework which allows other GMV's products development and evolution.

▷ Coming up with full life-cycle solution in the fields of satellite's dynamic operations thanks to its library of proven missions.

focussuite major goals are the ones obtained thanks to the possibility to reuse existing software, designing with a friendly client architecture, a database driven system, a graphical interface with high capabilities, portability which translated means different platform (i.e. Windows, UNIX) implementation and others which can be read in the GMV's ***focussuite*** web page as in [17].

Appendix D

Differential Evolution Optimiser

Optimization has always been one of the major tasks in the scientific field of expertise. The main objective pursued by the scientific community is optimising certain properties of a system by pertinently choosing the system parameters. It is usual representing these parameters in a vector.
In particular this optimiser has born from the attempt of Ken Price, one of the member of the International Computer Science Institute, to solve the Chebychev Polynomial fitting Problem.
The first problem to the differential evolution optimiser theory is to define an objective function which can take into account all the problem constraints. Actually an optimiser, to be called with such name, has to accomplish some requirements:

- ✓ Handling non-differentiable, nonlinear and multimodal cost functions.
- ✓ Providing convergence properties
- ✓ Easy to use

The DE satisfies all these requests and in particular can be defined as a parallel direct search method which utilises NP parameter vectors as population for each generation G.

$$x_{i,G},\ i = 1, 2, ..., NP - 1 \tag{D.1}$$

Initial population parameters can be selected arbitrary. Indeed the working method of this optimiser is based on computing trial parameters vector. These new parameter vectors are generated by adding a weighted difference vector between two population members to a third member. This mathematical step is called mutation. This computed vector is compared with the previous one and if the objective function acquires minor value with respect to the other, this last is overlapped with the new one. This process is computed for each generation in order to save the best vector and check its value absolutely.
A wide number of strategies related to this optimiser algorithm have been developed, but among all of them the two most promising schemes for this optimiser will be illustrated in next lines in greater detail.

▷ Scheme 1
The new trial vector is generated according to formula (D.2)

$$\vec{v} = \vec{x}_{r_1,G} + F \cdot (\vec{x}_{r_2,G} - \vec{x}_{r_3,G}) \tag{D.2}$$

where $r_1, r_2, r_3 \in [0,\ NP-1]$ integer and mutually different, with $F > 0$. F is a real and constant factor which controls the amplification of the factor difference variation presents into commas. Once that these values have been calculated, next step takes the name of *crossover*. It consists in performing the crossover operation, applying this one to each pair of the target vector $x_{i,G}$ and its corresponding mutant vector $v_{i,G}$ to generate a trial vector $u_{i,G}$. The most known crossover operation is the binomial one, which can be written as

$$u^j_{i,G} = \begin{cases} v^j & for\ \ j = \langle n \rangle_D, \langle n+1 \rangle_D ..., \langle n+L-1 \rangle_D \\ x^j_{i,G} & for\ \ all\ \ other\ \ j \in [0, D-1] \end{cases}$$

where the acute brackets $\langle\rangle_D$ denotes the module function with modulus D.

▷ Scheme 2
This second scheme works at the same way of the first one but the vector $\vec{v}$ is calculating according to

$$\vec{v} = \vec{x}_{r_1,G} + \lambda \cdot (\vec{x}_{best,G} - \vec{x}_{i,G}) + F \cdot (\vec{x}_{r_2,G} - \vec{x}_{r_3,G}) \tag{D.3}$$

simply introducing an additional control variable λ. This last mentioned variable is a mean to take into account the calculated current best vector $\vec{x}_{best,G}$. The steps which follows this first are the same of the previous said scheme 1.

Appendix E

PROPAG satellite's orbit propagator

Propag is the last generation numerical propagator developed in GMV's FDO space section. Its main objective is computing the evolution of an orbit forward and backward in time. Manoeuvres can be implemented in this software, both impulsive and long ones. The main time frame selected for this propagator is J2000.
As said countless time, propagation is one of the basis of all the satellite's flight dynamics. In particular propagating an orbit means to solve and integrate the equation of motions. This propagator implements an 8th-order Adams-Bashforth/Adams-Moulton prediction- correction method to numerically solve a system of **n** ordinary first order differential equations with initial values. The main equations which have to be integrated are the following

$$\frac{dy_1}{dx} = f_1\left(x, y_1, y_2, ..., y_n\right) \qquad ; \qquad y_1(x_0) = y_0^1 \tag{E.1}$$

$$\frac{dy_2}{dx} = f_2\left(x, y_1, y_2, ..., y_n\right) \qquad ; \qquad y_2(x_0) = y_0^2 \tag{E.2}$$

$$..... \tag{E.3}$$

$$\frac{dy_n}{dx} = f_2\left(x, y_1, y_2, ..., y_n\right) \qquad ; \qquad y_n(x_0) = y_0^n \tag{E.4}$$

where x_i is the independent variable and y_i is the dependent one. In order to ingrate these equations, which means evaluate y_i in successive points, it is necessary to start from an initial value that in this case is x_0. So the system can be rewritten as

$$\frac{dy}{dx} = f\left(x, y, ...\right) \qquad ; \qquad y(x_0) = y_0 \tag{E.5}$$

The *predictor-corrector method* previously mentioned is based on the assumption that the value y_{i+1} is predicted using the previous 9 points. Then introducing the first 8 points with this last one in a specific formula the predicted value is corrected, and this process is executed until the convergence is achieved. The differential equations are integrated using a single-step 8th-order Runge-Kutta method.
This propagator can implement real cases due to the presence of different perturbation models which have been introduced during its development. The force models considered are listed above:

- ✓ Solar radiation pressure
- ✓ Atmospheric drag
- ✓ Perturbation due to Earth Gravity Field
- ✓ Ocean Tides Perturbation
- ✓ Solid Tides Perturbation
- ✓ J_2/Moon Interaction
- ✓ Relativistic effects
- ✓ Third body gravitational force

In order to clarify readers ideas few words about these ones will be say. Atmospheric drag and solar pressure radiation are quite known: the first one is produced by the interaction of electromagnetic particles on the main vehicle and the second one is due to incident gas molecules on the satellite. Ocean and solid tides are phenomena which originate due to gravitational attraction of Moon and Sun. These tidal perturbations can be quite significant, reaching several seconds and so for precise computation have to be taken into account. The third body gravitational force is the acceleration on a body vehicle which orbits around the Earth, produced by the presence of Moon and Sun. The interaction with the Moon can not be studied with the three-body perturbation equation due to its proximity to the Earth, and perturbation value could be affected by the non-spherical mass distribution. Relativistic effects are the ones related to curvature in time space which can cause heavy body acceleration.
In next lines panels configuration of PROPAG's propagator will be illustrated.

- ► PROPAG main option panel
  In this section, likely it has been done for SAPO and GOLEM, main options are configured. In Fig. E.1, it is shown how it appears in the GUI tool.

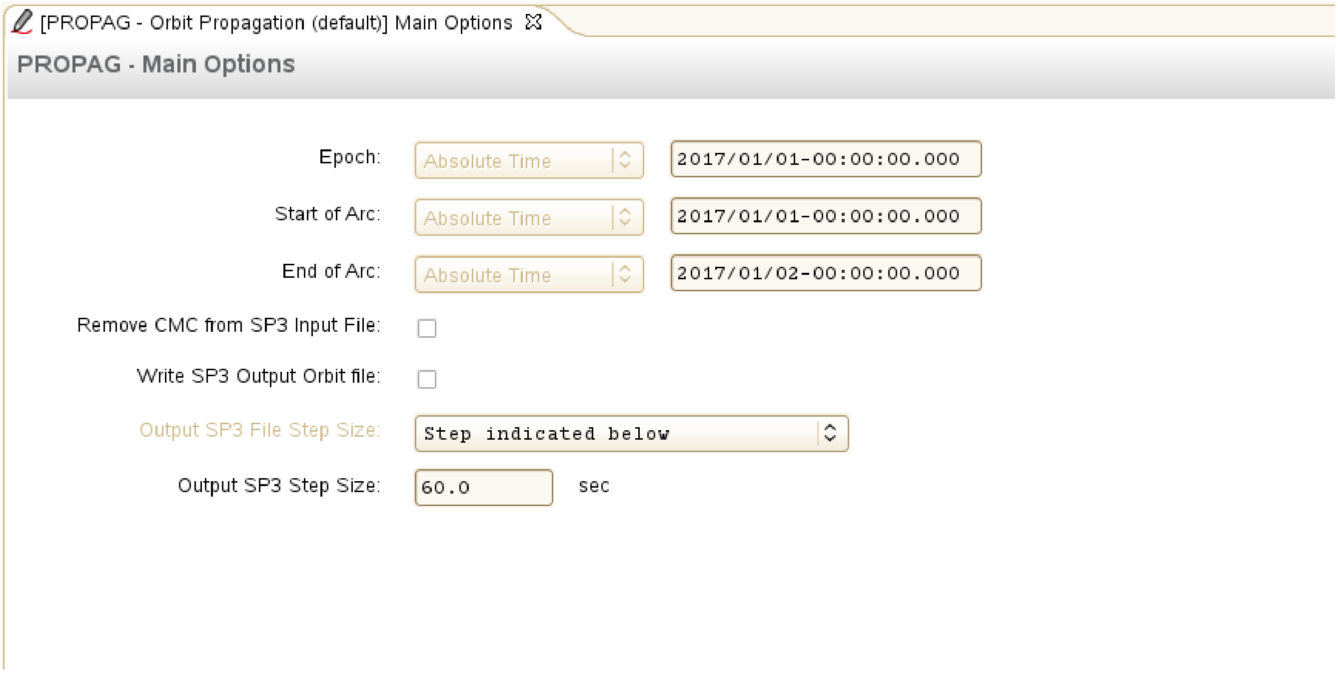

Figure E.1: PROPAG main options panel.

- PROPAG perturbation parameter panel
  This orbiter propagator could be configured so that real cases, with all effects which can perturb a satellite in space, can be analyzed. A representation of that can be seen in Fig. E.2. It is possible to see that the non-spherical Earth effects are considering thanks to the first two options present in the list of Fig. E.2. With the *Degree of gravity field* and *Order of gravity field* refers to the zonal harmonic polynomials used for definition of the Earth geopotential field.

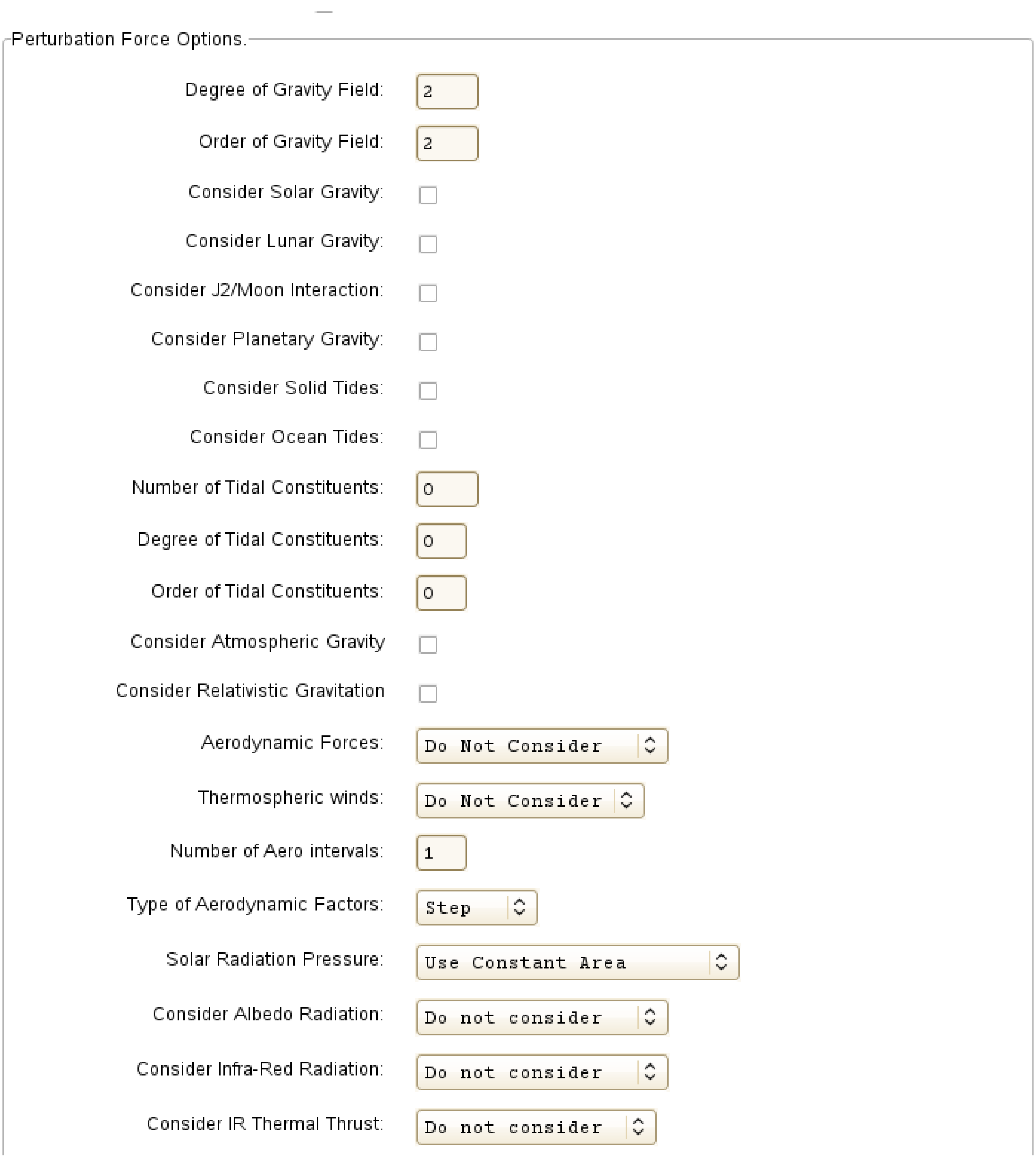

Figure E.2: PROPAG perturbations panel. Gravitational field and solar radiation pressure perturbation are just considering and signed with a check symbol.

- PROPAG parameters
  In this last significant panel, parameters are specified. In particular the initial state vector with all the perturbations constant which user would like to use, have to be introduced in this panel.

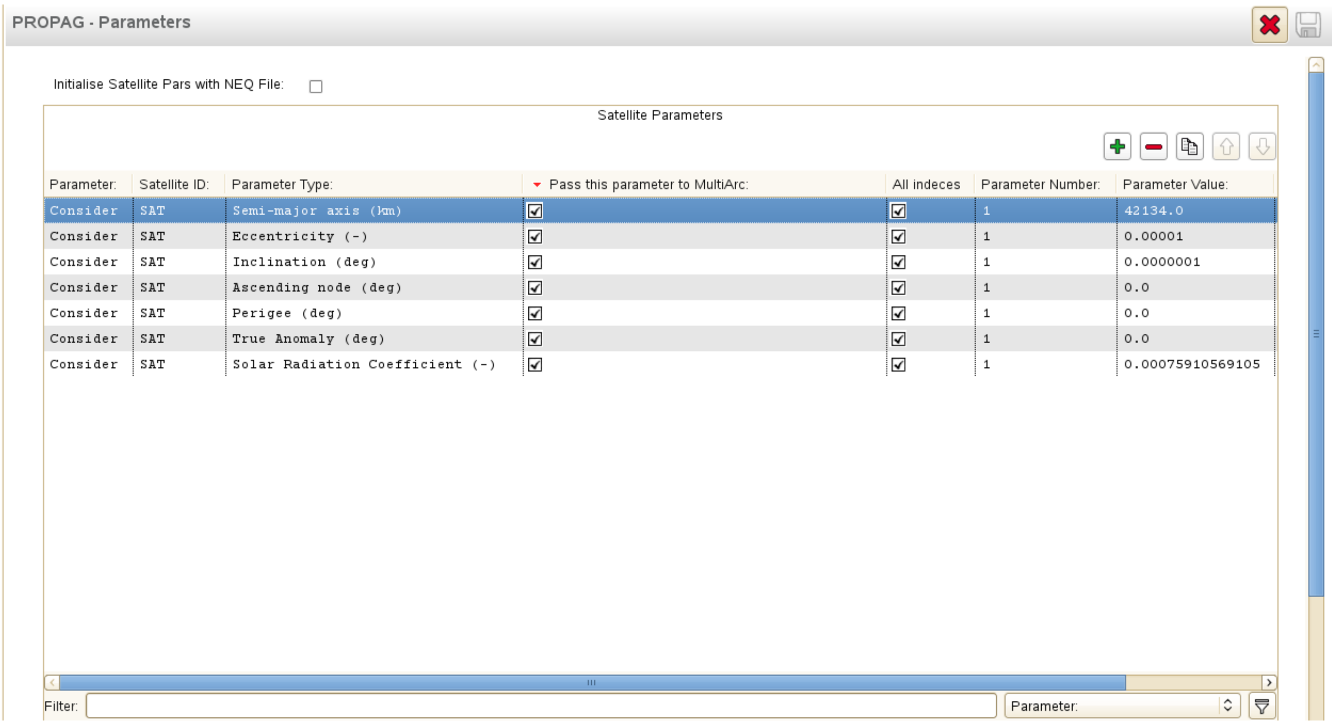

Figure E.3: PROPAG parameters panel.

Appendix F

Semi-analytic satellite theory for SAPO implementation

In this section of appendices, it will be described in grater detail the semi-analytic theory. A first approximated description has been made in 2.1.1. In next paragraphs this theory will be analysed focusing in all the mathematical aspects, from the equations to the operators used, which leads to the development of SAPO propagator.

F.1 First Aspects

The semi-analytic theory is used to represent the state of a satellite through the usage of the equinoctial elements, defined in appendix section A.2.3. Among the six, the first five are slowly varying in time while the last one, which corresponds to mean longitude, varies rapidly. One of the main characteristics of this theory is the difference between mean and osculating elements. Osculating elements, as already mentioned, describe the orbit followed by a body when perturbations are suddenly turned off, just leaving the effect of the central body. On the contrary the mean elements are the ones used to define an orbit which varies in time.
In particular this distinctions is important in order to apply the equation F.1, where the osculating elements are obtained as the sum of the mean ones plus a fast periodic 2π term which is referred to the mean longitude component

$$\hat{a}_i = a_i + \eta(a_i, a_2, ..., a_6, t) \tag{F.1}$$

The subscript i refers to the six elements defined in the state vector, the term $\hat{a}_i$ refers to the osculating elements and the terms $\hat{a}_i$ to the mean ones. The behaviour of these last can be explained using the differential equations written below in Eq. F.2, where letter n indicates the mean motion, ∂_{i6} is the Kronecker delta (this quantity can be described as $\partial_{i6} = 0$ if $i \neq 6$ and in the other case, ∂_{66}, it is equal to 1) and **t** represents time. The term A_i describes the temporal changes of the mean elements.

$$\frac{da_i}{dt} = n\partial_{i6} + A_i(a_i, a_2, ..a_5, t) \tag{F.2}$$

The short periodic variations, on the other hand, can be described thanks to the Fourier series

$$\eta_i = \sum_{j=1}^{\infty} \left[B_i^j(a_1,..,a_6,t)\sin j\lambda + C_i^j(a_1,..,a_6,t)\cos j\lambda \right] \tag{F.3}$$

where λ is the mean longitude, as described in appendix section A.2.3.
Among all the possible coordinate system, the equinoctial ones have been selected to develop this theory since they are nonsingular for all orbits to which the generalised method of averaging, used in this theory, is applied. Clearly, in order to be appreciated in all the coordinates, this theory can be used making transformation from the equinoctial components to the others.
In appendix section A.3, it has been described transformation process used to convert keplerian into other coordinates system. These conversions have been necessary to develop the GOLEM propagator.
In this case, the mathematical steps made to transform the equinoctial will not be discussed. Interested readers can find more information in [7].
As seen in 2.1.1, the VOP equations, Variation-of-Parameters, are the result of the introduction of the satellite orbit elements rates of change, obtained as function of themselves. This transformation has been necessary since this is a requirement of the generalised method of averaging.

F.2 Mathematical Operators

In this section it will be described, in general terms, mathematical operators useful in order to solve the **VOP**. As already mentioned in section 2.1.1, this method has been described making usage of the Cartesian elements, in the form which takes the name of **Lagrangian VOP equations**.
As said in the previously cited section 2.1.1, the SAPO propagator uses the equinoctial elements to compute the desired results. For this reason, the Poisson brackets have to be introduced into this context, since they are necessary to describe how the variation of the orbital parameters are obtained.
Clearly, this mathematical operator makes usage of partial derivatives applied to equinoctial elements with respect to position and velocity. The fisrt partial derivatives, the ones with respect to position can be described as follows

$$
\begin{aligned}
\frac{\partial a}{\partial \vec{r}} &= \frac{2a^2\vec{r}}{r^3} \\
\frac{\partial h}{\partial \vec{r}} &= -\frac{ahB\vec{r}}{(1+B)r^3} + \frac{k(p\dot{X} - q\dot{Y})\vec{w}}{AB} - \frac{B}{A}\frac{\partial \dot{\vec{r}}}{\partial k} \\
\frac{\partial k}{\partial \vec{r}} &= -\frac{akB\vec{r}}{(1+B)r^3} - \frac{h(p\dot{X} - q\dot{Y})\vec{w}}{AB} + \frac{B}{A}\frac{\partial \dot{\vec{r}}}{\partial h} \\
\frac{\partial p}{\partial \vec{r}} &= -\frac{C\dot{Y}\vec{w}}{2AB} \\
\frac{\partial q}{\partial \vec{r}} &= -\frac{C\dot{X}\vec{w}}{2AB} \\
\frac{\partial \lambda}{\partial \vec{r}} &= -\frac{\dot{\vec{r}}}{A} + \frac{(p\dot{X} - q\dot{Y})\vec{w}}{AB} - \frac{B}{A(1+B)}\left(h\frac{\partial \dot{\vec{r}}}{\partial h} + k\frac{\partial \dot{\vec{r}}}{\partial k}\right)
\end{aligned}
\tag{F.4}
$$

On the other hand, the ones calculated with respect to velocity are

$$
\begin{aligned}
\frac{\partial a}{\partial \dot{\vec{r}}} &= \frac{2\dot{\vec{r}}}{n^2 a} \\
\frac{\partial h}{\partial \dot{\vec{r}}} &= -\frac{(2\dot{X}Y - X\dot{Y})\vec{f} - X\dot{X}\vec{g}}{\mu} + \frac{k(qY - pX)\vec{w}}{AB} \\
\frac{\partial k}{\partial \dot{\vec{r}}} &= -\frac{(2X\dot{Y} - \dot{X}Y)\vec{g} - Y\dot{Y}\vec{f}}{\mu} + \frac{h(qY - pX)\vec{w}}{AB} \\
\frac{\partial p}{\partial \dot{\vec{r}}} &= -\frac{CY\vec{w}}{2AB} \\
\frac{\partial q}{\partial \dot{\vec{r}}} &= -\frac{CX\vec{w}}{2AB} \\
\frac{\partial \lambda}{\partial \dot{\vec{r}}} &= -\frac{2\vec{r}}{A} + \frac{(qY - pX)\vec{w}}{A} + \frac{1}{(1+B)}\left(k\frac{\partial h}{\partial \dot{\vec{r}}} - h\frac{\partial k}{\partial \dot{\vec{r}}}\right)
\end{aligned}
\tag{F.5}
$$

where the A, B and C terms present in the equations can be calculated using mathematical relations written below

$$
\begin{aligned}
A &= na^2 = \sqrt{\mu a} \\
B &= \sqrt{1 - h^2 - k^2} \\
C &= 1 + p^2 + q^2
\end{aligned}
\tag{F.6}
$$

It has to be mentioned that X and Y are the components which describe the position of the satellite in the equinoctial reference frame while their derivative, $\dot{X}$ and $\dot{Y}$, are the components of the velocity. These ones can be calculated as follows:

- the position derivative

$$\begin{aligned}
\frac{\partial X}{\partial h} &= -\frac{k\dot{X}}{n(1+B)} + \frac{aY\dot{Y}}{AB)} \\
\frac{\partial Y}{\partial h} &= -\frac{k\dot{Y}}{n(1+B)} - \frac{aX\dot{Y}}{AB)} - a \\
\frac{\partial X}{\partial k} &= -\frac{h\dot{X}}{n(1+B)} + \frac{aY\dot{X}}{AB)} - a \\
\frac{\partial Y}{\partial k} &= -\frac{h\dot{Y}}{n(1+B)} - \frac{aX\dot{X}}{AB)}
\end{aligned} \tag{F.7}$$

- the velocity derivative

$$\begin{aligned}
\frac{\partial X}{\partial h} &= \frac{a\dot{Y}^2}{AB} + \frac{A}{r^3}\left(\frac{akX}{1+B} - \frac{Y^2}{B}\right) \\
\frac{\partial Y}{\partial h} &= -\frac{a\dot{Y}\dot{X}}{AB} + \frac{A}{r^3}\left(\frac{akY}{1+B} + \frac{YX}{B}\right) \\
\frac{\partial X}{\partial k} &= \frac{a\dot{Y}\dot{X}}{AB} + \frac{A}{r^3}\left(\frac{ahY}{1+B} + \frac{YX}{B}\right) \\
\frac{\partial Y}{\partial k} &= -\frac{a\dot{X}^2}{AB} - \frac{A}{r^3}\left(\frac{ahY}{1+B} - \frac{X^2}{B}\right)
\end{aligned} \tag{F.8}$$

Poisson demonstrated that it is possible to apply the method of variation of the parameters to the potential contribution of the disturbance using the following expression,

$$\frac{\mathrm{d}a_i}{\mathrm{d}t} = -\sum_{j=1}^{6}(a_i, a_j)\frac{\partial R}{\partial a_j} \tag{F.9}$$

The term (a_i, a_j) is named Poisson brackets. The general expression of this operator is defined as,

$$(a_i, a_j) = \frac{\partial a_i}{\partial \vec{r}}\frac{\partial a_j}{\partial \dot{\vec{r}}} - \frac{\partial a_i}{\partial \dot{\vec{r}}}\frac{\partial a_j}{\partial \vec{r}} \tag{F.10}$$

Poisson brackets are independent of time and it is immediately evident that,

$$\begin{aligned}
(a_i, a_i) &= 0 \\
(a_i, a_j) &= -(a_j, a_i)
\end{aligned} \tag{F.11}$$

Therefore, only fifteen Poisson Brackets are independent. Among all the fifteen, four are equal to zero, while the others acquire specific values:

$$
\begin{aligned}
(a,h) &= 0 \\
(a,k) &= 0 \\
(a,p) &= 0 \\
(a,q) &= 0 \\
(a,\lambda) &= -\frac{2}{na} \\
(h,k) &= -\frac{B}{A} \\
(h,p) &= -\frac{kpC}{2AB} \\
(h,q) &= -\frac{kqC}{2AB} \\
(h,\lambda) &= \frac{hB}{A(1+B)} \\
(k,p) &= -\frac{hpC}{2AB} \\
(k,q) &= -\frac{hqC}{2AB} \\
(k,\lambda) &= -\frac{kB}{A(1+B)} \\
(p,q) &= -\frac{C^2}{4AB} \\
(p,\lambda) &= -\frac{pC}{2AB} \\
(q,\lambda) &= -\frac{qC}{2AB}
\end{aligned}
\tag{F.12}
$$

At this point it could be useful to introduce the direction cosines, (α, β, γ), although they can not be considered as mathematical operators, they will be discussed in this section. They are commonly used to describe the conservative perturbations and can be calculating as follows

$$
\begin{aligned}
\alpha &= \mathbf{z_B} \cdot f \\
\beta &= \mathbf{z_B} \cdot g \\
\gamma &= \mathbf{z_B} \cdot w
\end{aligned}
\tag{F.13}
$$

where (f, g, w) are the basis vectors of the equinoctial reference frame and z_B is the vector, with unitary module, whose direction goes from the centre of mass to the third body.
It has to be noticed that these cosines are not independent and in particular their relationship is

$$\alpha^2 + \beta^2 + \gamma^2 = 1 \tag{F.14}$$

These variables, at the same way of the basis vector, are defined in function of p and q, but in addition these first depend on time since $\mathbf{z_B}$ depends on it.

F.3 Equation of Motion: from Cartesian to VOP

As it has already been mentioned in Sec. 2.1.1 speaking about theory behind the SAPO propagator, the Cartesian equation of motion which describes the movement of a satellite in an inertial coordinates system is the following

$$\ddot{\vec{r}} = \frac{-\mu}{r^3}\vec{r} + \vec{q} + \nabla\mathscr{R} \tag{F.15}$$

In order to find a solution for this problem it is necessary to refer to the generalised method of averaging, as it will be described in next section. Before computing this step a transformation has to take place, from the Cartesian equation of motion to VOP, variation of parameters. The set of equations which come up from this conversion is represented in F.16, where it is reminded that the terms $(a_1,...,a_6) = (a,h,k,p,q,\lambda)$ indicate *osculating equinoctial elements*. More information about that can be found in [16] and [22].

$$\dot{a}_i = n\delta_{i6} + \frac{\partial a_i}{\partial \vec{r}}\vec{q} - \sum_{j=1}^{6}(a_i,a_j)\frac{\partial \mathscr{R}}{\partial a_j} \tag{F.16}$$

This equation is composed from three terms which define the rate of change of this orbital elements.

- The first term is the one which concerns to the two-body problem

$$\dot{a}_i = n\delta_{i6} \tag{F.17}$$

- The second one refers to the Gaussian or nonconservative contribution to equation

$$\dot{a}_i = \frac{\partial a_i}{\partial \vec{r}}\vec{q} \tag{F.18}$$

- The third, and last one, the Lagrangian or conservative part which can be written as follows

$$\dot{a}_i = -\sum_{j=1}^{6}(a_i,a_j)\frac{\partial \mathscr{R}}{\partial a_j} \tag{F.19}$$

The mathematical operators which appear in the equations written above are:

- The Kronecker delta defined as δ_{i6} whose values are 0 if $i \neq 6$, and 1 if $i = 6$.
- The partial derivative $\frac{\partial a_i}{\partial \vec{r}}$
- The Poisson brackets (a_i, a_j), described in previous section.
- The term $n = \sqrt{\frac{\mu}{a^3}}$ represents the Kepler mean motion.
- $\dot{r} = \frac{d\boldsymbol{r}}{dt}$ indicates the velocity vector.

Focusing on the last of these three terms, the Lagrangian ones, it includes the partial derivatives of the disturbing function $\mathscr{R}$ with respect to p and q. Despite the dependence to these variables, it is more functional to define these derivatives using the direction cosines, mentioned in section related to mathematical operators, together with the following terms (a,h,k,λ). The Chain Rule is the mean thanks to which it is possible to evaluate the partial derivatives of the disturbing function $\mathscr{R}$ with respect to p and q. It results:

$$\begin{aligned}\frac{\partial \mathscr{R}}{\partial p} &= \frac{\partial \mathscr{R}}{\partial \alpha}\frac{\partial \alpha}{\partial p} + \frac{\partial \mathscr{R}}{\partial \beta}\frac{\partial \beta}{\partial p} + \frac{\partial \mathscr{R}}{\partial \gamma}\frac{\partial \gamma}{\partial p} \\ \frac{\partial \mathscr{R}}{\partial p} &= \frac{\partial \mathscr{R}}{\partial \alpha}\frac{\partial \alpha}{\partial p} + \frac{\partial \mathscr{R}}{\partial \beta}\frac{\partial \beta}{\partial p} + \frac{\partial \mathscr{R}}{\partial \gamma}\frac{\partial \gamma}{\partial p}\end{aligned} \tag{F.20}$$

where it may be convinient to describe the cross-derivative operator as below

$$\mathscr{R}_{,\alpha\beta} = \alpha\frac{\partial \mathscr{R}}{\partial \beta} - \beta\frac{\partial \mathscr{R}}{\partial \alpha} \tag{F.21}$$

Referring to the variable A,B, C seen in the previous section, all these procedures lead to the following equation which can be resumes as the Lagrangian part of the VOP equations of motion

$$\begin{aligned}\dot{a} &= \frac{2a}{A}\frac{\partial \mathscr{R}}{\partial \lambda} \\ \dot{h} &= \frac{B}{A}\frac{\partial \mathscr{R}}{\partial k} + \frac{k}{AB}(p\mathscr{R}_{,\alpha\gamma} - q\mathscr{R}_{,\beta\gamma}) - \frac{hB}{A(1+B)}\frac{\partial \mathscr{R}}{\partial \lambda} \\ \dot{k} &= -\frac{B}{A}\frac{\partial \mathscr{R}}{\partial h} - \frac{h}{AB}(p\mathscr{R}_{,\alpha\gamma} - q\mathscr{R}_{,\beta\gamma}) - \frac{kB}{A(1+B)}\frac{\partial \mathscr{R}}{\partial \lambda} \\ \dot{p} &= \frac{C}{2AB}\left[p\left(\mathscr{R}_{,hk} - \mathscr{R}_{,\alpha\beta} - \frac{\partial \mathscr{R}}{\partial \lambda}\right) - \mathscr{R}_{,\beta\gamma}\right] \\ \dot{q} &= \frac{C}{2AB}\left[q\left(\mathscr{R}_{,hk} - \mathscr{R}_{,\alpha\beta} - \frac{\partial \mathscr{R}}{\partial \lambda}\right) - \mathscr{R}_{,\alpha\gamma}\right] \\ \dot{\lambda} &= -\frac{2a}{A}\frac{\partial \mathscr{R}}{\partial a} + \frac{B}{A(1+B)}\left(h\frac{\partial \mathscr{R}}{\partial h} + k\frac{\partial \mathscr{R}}{\partial k}\right) + \frac{1}{AB}\left(p\mathscr{R}_{,\alpha\gamma} - q\mathscr{R}_{,\beta\gamma}\right)\end{aligned} \tag{F.22}$$

Similarly, one can obtain the Gauss form of VOP equations.

F.4 Generalized Averaging Equations

The Generalized Method of Averaging is applied to the VOP equations of motion in order to separate these ones into two parts:

- A short-periodic part which can be integrated analytically,

- A slowly-varying part which should be integrated numerically.

A simply overview of this approach will be discussed in these lines, leaving the readers find more information in [10] and [22]. First of all it has to be assumed that the osculating elements, indicated as $\hat{a}_i$, are obtained from the sum of the mean elements a_i and a short period variation of these last ones indicated as η_i. These sum can be represented using the following mathematical relation

$$\hat{a}_i = a_i + \sum_{j=1}^{\infty} \varepsilon^j \eta_{i,j}(a,h,k,p,q,\lambda,t) \tag{F.23}$$

The quantity ε symbolises a little short periodic variation of order j in element i. The magnitude ε is named the *small parameter* and is the main variational parameter fundamental in order to obtain the Equations of Averaging.

There are some requirements in defining the short periodic variations, necessary to allow defining this theory. One of these requirements, is that these variations have to include all of the high-frequency components in the osculating elements, which means that the mean elements should vary slowly with time. These demands can be mathematically translated making usage of the following set of inequalities:

$$\frac{1}{n}\left|\frac{\mathrm{d}a}{\mathrm{d}t}\right| \ll a \tag{F.24}$$

$$\frac{1}{n}\left|\frac{\mathrm{d}a_i}{\mathrm{d}t}\right| \ll 1 \qquad \text{for } i = 2,3,4,5 \tag{F.25}$$

$$\frac{1}{n}\left|\frac{\mathrm{d}\lambda}{\mathrm{d}t} - n\right| \ll 1 \tag{F.26}$$

$$\Delta^{k+1}\left|\frac{\mathrm{d}^{k+1}a}{\mathrm{d}t^{k+1}}\right| \ll a \tag{F.27}$$

$$\Delta^{k+1}\left|\frac{\mathrm{d}^{k+1}a_i}{\mathrm{d}t^{k+1}}\right| \ll 1 \qquad \text{for } i = 2,3,4,5,6 \tag{F.28}$$

The inequalities mentioned above ensure that the second-order effects will be small. The same will be valid for the integrator errors. Once these preliminaries have been discussed it is possible to rewrite the VOP equation of motion as presented above

$$\frac{\mathrm{d}\hat{a}_i}{\mathrm{d}t} \simeq n(\hat{a})\delta_{i6} + \varepsilon\left[\frac{\partial \hat{a}_i}{\partial \vec{r}}\vec{q} - \sum_{j=1}^{6}(\hat{a}_i,\hat{a}_j)\frac{\partial R}{\partial \hat{a}_j}\right] \tag{F.29}$$

and as follows

$$\frac{\mathrm{d}\hat{a}_i}{\mathrm{d}t} \simeq n(\hat{a})\delta_{i6} + \varepsilon F_i(\hat{a},\hat{h},\hat{k},\hat{p},\hat{q},\hat{\lambda},t) \tag{F.30}$$

$$\frac{\mathrm{d}a_i}{\mathrm{d}t} = n(a)\delta_{i6} + \varepsilon A_i(a,h,k,p,q,t) \tag{F.31}$$

F.5 Latest SST comments

More and more can be said about this method. Information about short-periodic variations, conversions method, truncation algorithms which are included in the SST, Semi-analytic Satellite Theory, can be found in [7]. Anyway, it is not of this thesis interest to focus in greater detail on the mathematical aspects which involves the SST theory. A simple overview has been discussed to give reader a general idea of how can be complex to implement this satellite propagation theory and to leave to the braver one the assignment of investigating fully.

Appendix G

General Perturbation Theory

Astronomically speaking, perturbation refers to a wide class of phenomena which can be divided in two major group which are the *secular* perturbations, and on opposite the *periodic* ones. General definition of perturbation refers to forces, generated by other massive bodies, or resistance due to atmosphere, acting on a massive body along is motion. When perturbations are considered in analyzing a general situation, solutions can be obtained only for two-body problem. In case of more than two bodies, or body's irregular shapes, analytic solution does not exist.

Secular contributions refer to perturbation which varies over time continuously. The exact definition can be described as the cumulative effect on a body moving along an orbit, which causes the growing or decrease of an orbital elements. Among all, it has to be mentioned the gravitational attractions of the planets due to whom secular changes in the longitude of the ascending node, longitude of perihelion, and times of perihelion passage can occur.

On the other hand periodic perturbations are the ones which repeats over the course of a time frame of interest, or in other terms that periodically change their direction. These deviations which can occur during the satellite or planet motion on its orbit general do not exceed a century.
In particular it is possible that these last ones superpose to the secular perturbation as, for example, in the case of the perturbation of the longitude of the node. Indeed it is demonstrated by the fact that the rate of secular regression is not constant in time. An other example of short perturbation is the one related to the distributions of continents and oceans, and also to the varying mass densities in Earth's mantle. All these phenomena cause a deviation of Earth's gravitational force field from the axial symmetry. As consequence short period perturbation of low amplitude for satellites positioned near Earth may be generated.

Appendix H

FORTRAN 90

Fortran, which stands for ***Formula Translation***, is a high-level programming language used in the field of numeric computation and scientific computing. It is used in many fields of expertise. Among them can be mentioned some such as numerical weather prediction, finite element analysis, computational fluid dynamics, computational physics, crystallography and computational chemistry.
The first Fortran compiler has been devolved in 1957 at IBM, International Business Machines Corporation, a major US company dedicated to informatics and technology. During the years different versions of this scientific language have been released until the 1990 when the one of the latest version, inclusive of new features, have been updated. Actually, a revised version for Fortran 90 called Fortran 95, has been introduced in 1997. It can be considered equal to the previous one with some resolved issues which affected the 90 version.
It has to be said that Fortran 90 main goals, in opposite to the Fortran 77 version, have been the introduction of modular structures which make the codification easier and less vulnerable to data storage, and the creation of pointers which allow manipulation of dynamic structures. In 2003 a significant revisited version has been developed with respect to Fortran 95. Some smaller changes have been introduced in 2008 and 2015. Fortran main features can be resumed in the list present above:

- ✓ A scientific language easy to learn.
- ✓ Simple way to express mathematics functions, even the more complex ones.
- ✓ Problem orientated language.
- ✓ Possibility to control storage allocation.

Bibliography

[1] Montenbruck Oliver, Gill Eberhard. *Satellite Orbits-Models, Methods and Applications.* Springer-Verlag Berlin Heidelberg, 2000

[2] E.M. Soop. *Handbook of Geostationary Orbits.* Springer Netherlands, December 2010

[3] Marcin Ligas, Piotr Banasik. *Conversion between Cartesian and geodetic coordinates on a rotational ellipsoid by solving a system of nonlinear equations*. Department of Geomatics, AGH University of Science and Technology, 30 Mickiewicza Al., 30-059 Krakow, Poland.

[4] Howard D. Curtis. *Orbital Mechanics for Engineering Student.* Embry-Riddle Aeronautical University Daytona Beach, Florida. Amsterdam: Elsevier Butterworth Heinemann.

[5] D. A. Danielson, C. P. Sagovac, B. Neta, L. W. Early. *SEMIANALYTIC SATELLITE THEORY.* Mathematics Department, Naval Postgraduate School, Monterey, CA 93943

[6] S. Widnall, J. Peraire. *Lecture L17 - Orbit Transfers and Interplanetary Trajectories.* 16.07, Dynamics Fall, 2008, Version 2.0

[7] D. A. Danielson and C. P. Sagovac and B. Neta and L. W. Early. *Semianalytic Satellite Theory.* Tech. rep. Naval Postgraduate School Department of Mathematics Monterey, 1995

[8] Rainer Storn, Kenneth Price. *Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous space.*

[9] Erik Gregersen. *Astronomical Observations: Astronomy and the Study of Deep Space* Binding – December 20, 2009

[10] Nayfeh, A.H. *Perturbation Methods*.Pure and Applied Mathematics, Wiley, 1973

[11] Adrian Perez. *Enhancement, optimization, and implementation of truncation algorithms in a semi-analytical analytical propagator.* 2017

[12] Tamara Casillas. *Semi-analytical Propagator Development: Manoeuvres implementation*. 2017

[13] W. D. McClain. *A recursively formulated first-order semianalytic artificial satellite theory based on the generalized method of averaging. Volume 1: The generalized method of averaging applied to the artificial satellite problem.* NASA, 1977.

[14] D.A. Vallado. *Fundamentals of Astrodynamics and Applications*. Microcosm Press and Kluwer Academic Publishers, 2001

[15] René Schawrz. *Keplerian Orbit Elements to Cartesian*.

[16] Cefola, P. J., Long, A. C., and Holloway, G. *The Long-Term Prediction of Artificial Satellite Orbits*. AIAA Paper 74–170, AIAA Aerospace Sciences Meeting, Washington, DC, January 1974.

[17] https://www.gmv.com/en/Products/focussuite/info_focussuite

[18] https://earthobservatory.nasa.gov/Features/OrbitsCatalog/page2.php

[19] http://www1.icsi.berkeley.edu/∼storn/code.html

[20] Marcel J. Sidi. *Spacecraft Dynamics and Control: A Practical Engineering Approach*. Cambridge Aerospace Series, 1997.

[21] Michel Capderou. *Handbook of Satellite Orbits: From Kepler to GPS*. Springer, 2014.

[22] W. D. McClain. *A recursively formulated first-order semianalytic artificial satellite theory based on the generalised method of averaging*. Volume 1: The generalised method of averaging applied to the artificial satellite problem. NASA, 1977.