POLITECNICO DI TORINO

Master's Degree in AEROSPACE ENGINEERING

Master's Degree Thesis

Development of ORCA: On-ground Risk Casualty Assessment

Supervisor
Prof. MANUELA BATTIPEDE
Technical Supervisors
Ing. MICHELE MARCONE
Ing. CHRISTOPHE ROUX

Candidate
Giulia Chinappi

April 2024

Abstract

According to European and American regulations, whenever a company is capable of bringing something into space, it has to provide data about the collective risk bounded to the operations. The total risk shall cover the entire mission, from launch to disposal, and the accepted value of risk is usually stated in a set of regulations and requirements according to multiple criteria (casualty risk, risk to critical assets, toxic risk). With this aim, the objective of the present thesis work is to develop and introduce a dedicated tool, called ORCA (On-ground Risk Casualty Assessment), capable of propagating the trajectory and evaluating the casualty risk at launch and at reentry, with a low computational cost. A 3 DOF model is developed for the reentry and launch phase in order to compute the collective risk on ground. The tool enables the choice between two modules, one for the risk during launch and one for a generic reentry condition. The user can set the quantities in the input file and the tool is completely autonomous. With a low computational cost, it is able to provide the impact points of the reentring fragments, given as input, considering also the uncertainites related to their ballistic and aerodynamics properties. The results showed in this work have the scope of validation, they have been compared with the ones generated by other similar tools on the shelf. In these simulations the nominal and the scattered initial conditions, for both the modules have been considered. The final aim of this tool is to develop a flexible and indipendent inner tool to be used in safety submissions.

Table of Contents

List of Tables

List of Figures

Acronyms

ORCA
: On-ground Risk Casualty Assessment

LEO
: Low Earth Orbit

SSO
: Sun Syncronous Orbit

FSOA
: French Space Operation Act

FPA
: Flight Path Angle

FAA
: Federal Aviation Administration

CA
: Casualty Area

ISA
: International Standard Atmosphere

NRLMSISE
: US Naval Research Laboratories Mass Spectrometer and Incoherent Scatter Esosphere

ECI
: Earth Central Inertial

ECEF

Earth Central Earth Fixed

LVLH

Local Vertical Local Horizon

Chapter 1

Introduction

1.1 The risk problem

If a nation has the capability to launch something into orbit it has also the duty to protect the population on ground during the operations. There are various way to compute the risk and avoid consequences on population:

- simple analysis on the impact points on ground, linked to one single corridor;
- failure analysis, followed by a trajectory analysis and structural loads for the debris generation;
- risk assessment linked to toxic emissions, nuclear dangers or radiations.

These analysis culminates with risk assessment and definition of mitigation actions in order to minimize the consequences. Usually there are two kind of mitigations:

- considering the solution of the risk implementing a flight termination system that allows the control on the impact points;
- consider a design capable of reducing the risk, such as implementing redundancies on every critical system.

In this work two different kind of risk will be considered: the collective and the individual risk. The collective risk is the number of individuals statistically expected to be exposed to a specified injury. The individual risk is the probability that the maximally exposed individual will suffer the specified injury[1]. As it can be noticed the two risk are slightly different and they depends on a protection factor taken into account, in these simulations this number has been set to one in order to stay conservative, the risk can be reduced if the protection factor is taken into account. After the definition of the dangerous area the authorities must be informed and

they will proceed to the isolation of that area, it is usually done in the launch pad proximity, when the launcher flies over habitated areas is necessary to implement mitigation action in order to respect the risk limits imposed by the regulations.

The main regulation istitutions are the FAA for the United States territories and the ESA for Europe, but the reference regulation in this case is the French one, FSOA. This difference is due to the launch pad position in French Guiana, that is a French territories and, as consequence, it is under the French legislation.

The FAA documentation regarding the on ground risk connected to the launching phase is the 14 CFR 123 [2] and 14-CFR 101 [3]. Attached to these documents there are also the advisory circulars that gives some guidelines in order to respect the requirements of the standards. In particular, here are listed the requirements, according to the 14 CFR 450.101:

1. The collective risk, in terms of expected number of casualties, must not exceed $1 * 10^{-4}$ for both re-entry events and launching ones.

2. The individual risk, in terms of probability of casualty, must not exceed $1*10^{-6}$, for both reentry and launching phases.

The computation of these numbers are based on a formula indicated in the AC 450.101, that depens on:

- the fragment casualty area
- the population density of the impact area
- the protection factor bounded to the fragment's impact energy, period of the year and local time
- reliability of the launcher, in case of the launching risk module

The ESA regulation follows the French regulation, due to the fact that the launch site is based in a French region, French Guiana, as said before. The reference regulation is the FSOA [4], published in 2008, in which there are indicated the requirements on the values of the risk. In the technical part there are the following risk limitation:

- $2 * 10^{-5}$ for the collective risk during launch, that is defined in the opposite way previous exposed;
- 10^{-7} per nominal elements fallout, that are planned to reentry before the orbit injection;
- $2 * 10^{-5}$ for the atmospheric controlled reentry;

- 10^{-4} in the case of uncontrolled reentry of launcher fragments after their orbit injection.

The value presented above are referring only to the collective risk, in order to consider also other types of risk such as the toxic or the explosive one, other analysis have to be performed. In particular the FAA underlines the limits also in the other cases, but ORCA computes only the risk connected to the casualty one and not to the expolsion or to the indirect risk. The regulations provides also information about how the outcomes and evidence have to be presented. In particular all the trajectories have to be provided, considering the nominal simulation and the one with uncertainties, then it is necessary to include also the motivations of the choices and the mitigations actions.

1.1.1 The reference tool

Currently, in the company, the computation of the casualty risk relies on an external software, used in the European environment. This is a tool made up of different modules with a lot of post-processing capacities: it can display impact points on a map, it can extract trajectories. For the porpouse of this work only few modules of this software has been used: the launching and the controlled re-entry module.

The tool's interface is very clear and allows the user to set the quantities for the simulations. First of all the user has to select which module he wants to use, then it is necessary to set all the parameters for the simulation. After that the tool is able to simulate the impact point and compute the risk that are displayed in a text file.

The main challenges of this software, besides the time consuming pre-processing of the data, is that it has been developed only for the European launchers and it is based on the European standards, in this way it can not be used for others aims. It is also not completely clear where the data that it uses are taken from: this work, instead, operates only with replicable and easy to find data.

Concerning the pre-processing steps, there are some of them that are crucial. For example, for the risk during launch, the software needs the trajectory file as a complicated file, a tailored document very different from a text file that usually is an output of the trajectory tools, or it needs the fragmentation model as an xml file, different from the excel file typically assembled by other tools.

About the output, here there are two example of the text file generated by this software. As it can be noticed the layout, shown in figure 1.2, related to the launching risk, is bit chaotic: in this case there are a lot of information for each fragment on each impact point, thus the file could be very intricated and unreadable. The other figure, 1.1 is the outcome generated by the other module, but by its nature it results less complicated.

```
4.541630555770792  -118.160137948101550 25957.06327445471724  3353.912887569900700     9.401777159377783    17.270000000000000
```

Figure 1.1: Example of the output file of the reference tool

```
350 1 1 0 1 1    30.788902111177010   -56.719355112806950 18262.01257973249646  1086.888887694239900    13.223177726983094
350 1 2 0 1 1    31.735090156134902   -56.909387842696816 18262.0097426682355   841.766535547076500    42.033335403665120
350 1 3 0 1 1    31.945498995982295   -56.953553338791544 18262.00866555521887   748.703970910719600    99.223831673392080
350 1 4 0 1 1    31.809618156222328   -56.924541354500570 18262.00954378268698   824.582824155373600    48.688904641881680
350 1 5 0 1 1    31.898982258096550   -56.942961964014934 18262.00923797756292   798.161261435945400    62.417117531435770
350 1 6 0 1 1    31.879320561894623   -56.938857741634266 18262.00932206604568   805.426506346798100    58.214549067951770
350 1 7 0 1 1    31.900243625761590   -56.943227992191750 18262.00923157941507   797.608461462277400    62.746974808064984
350 1 8 0 1 1    31.917960405481320   -56.947030226226010 18262.00911647661188   787.663579266214800    68.925642119447870
350 1 9 0 1 1    31.894583528773403   -56.942040222091700 18262.00925801522968   799.892515844541200    61.383073975199640
350 1 10 0 1 1    31.445695715373400    -56.851116073854320 18262.01044473012293   902.424682621298400    27.608311596987246
350 1 11 0 1 1    31.688875985837086    -56.900045707729840 18262.00985386950724   851.374325425883500    38.914666451959580
350 1 12 0 1 1    31.042297293616926    -56.770357490042265 18262.01156034690149   998.813972288559900    17.532027987497500
```

Figure 1.2: Launch event output file of the reference tool

1.2 Goals and structure of the thesis

The purpose of this thesis is the development of a tool similar to the reference tool, called ORCA. It is able to propagate the trajectory of the involved body considering it as a 3 DOF object, after that it computes the impact points and, consequently, the risk. It is made up of two modules: the first is for the computation of the risk during launch and the second is for the de-orbiting risk. The validation of ORCA is done through two reference tools, in order to have different points of view. The tool is written in *MATLAB* language, following a modular scheme in a way that every module can be used separately. The development started from the study of the physical problem, bounded to the ballistic re-entry, after that some others models have been developed with different hypotesis, then the uncertainties analysis has been performed and validated.
This thesis is organized in the following chapters:

- Chapter 1: general introduction on the risk problem and to the thesis.
- Chapter 2: company's general description.
- Chapter 3: the physical model, where there is the explanation of the hypotesis and the forces that act on the body.
- Chapter 4: the propagator module, it discusses the integration methods implemented in the code.
- Chapter 5: the chapter dedicated to the simulation modules, where the working of the modules are explained.
- Chapter 6: this one is completely dedicated to the validation of the tool, there are the comparisons between the results of ORCA and others tools and, more in general, the ORCA's results.

- Chapter 7: the analysis of the analytic problem. This has been developed in order to speed up the main code, allowing the computation of the propagation beginning after the first instant of the trajectory file.
- Chapter 8: the final chapter with a summary of the work and its future improvements.
- Appendices: they contain the necessary files for the correct execution of the tool.

Chapter 2

Company's description and its missions

2.1 Avio and its business

Avio is an italian space company, based in Colleferro, near Rome and its main business is the launch systems production. The Avio's main space products are launchers, they are three, two of them operatives and the third in a project phase: Vega, Vega-C and Vega-E. Vega is the first launcher, its inaugural flight was in 2012,

Figure 2.1: Vega's family [5]

it is the oldest of the family and it is capable of taking to a LEO orbit 1.5 tons of payload. Vega-C is the evolution of Vega, it has more capacity: it can take to the same orbit 2.3 tons of payload. Its first flight was in 2022 and it is the only active Avio's launcher, considering that 2024 will be the year of the last flight of Vega. At the end there is Vega-E, the launcher in the development phase. It designed for light satellites, with reduced operational costs and low environmental impact

because of the propellant used in the last stage's engine. Avio has also a tactical division for solid propellant missiles and a commercial project, in collaboration with other space companies, called Space Rider.

2.1.1 Vega

Vega is the consolidated launcher of Avio, it is a four stage rocket, three of them have solid propellant motors and the last one has a liquid propulsion system, with dimethil hydrazine(UDMH) and nitrogen tetroxide(NTO). From the figure 2.2, the main parts of the launcher can be identified:

- first stage equipped with P80 solid motor;
- second stage powered by zefiro 23 (Z23);
- third stage with zefiro 9 (Z9);
- fourth stage propulsed by AVUM, the liquid propellant engine

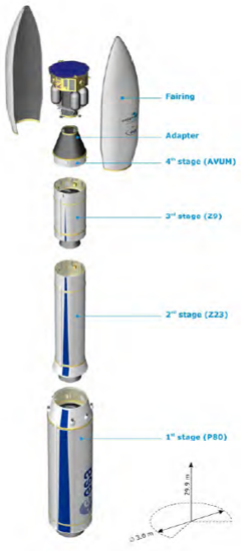

Figure 2.2: Vega's structure

2.1.2 A typical Vega mission

A common Vega mission is to SSO orbit and, in this case, three phases can be identified:

- the first one is the ascent, in order to reach the required orbit;

- the second one is the maneuvers one, for the satellite positioning;
- the last one is the last stage deorbiting;

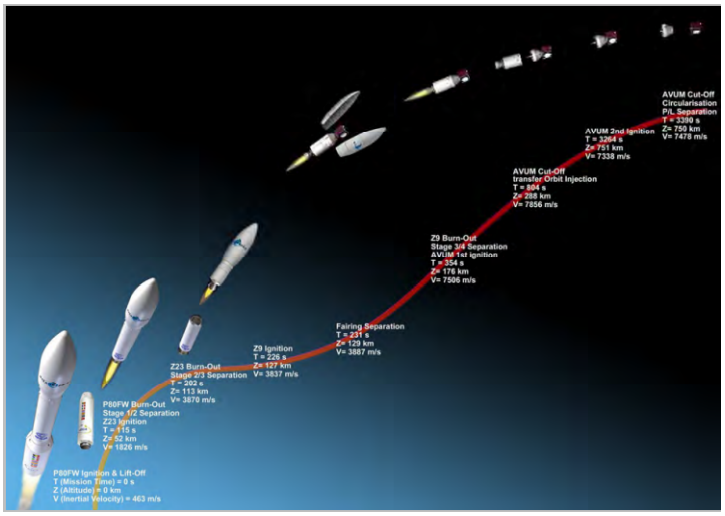

Figure 2.3: Ascent profile of a typical Vega mission[6]

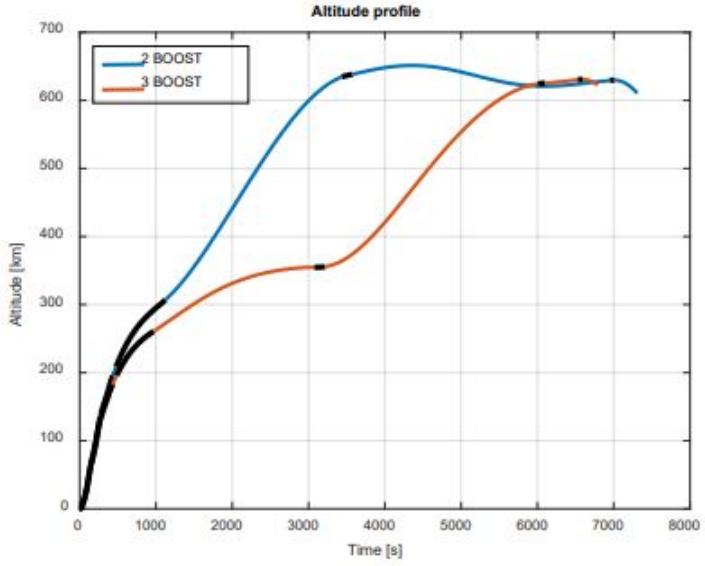

Figure 2.4: Vega ascent profile, comparison between 2 or 3 AVUM's boosts [7]

As it can be noticed by the figure 2.3, a typical ascent phase is characterized by an initial vertical ascent with P80, after that there is the zero-incidence flight phase

performed by Z23. At the end of the second stage, there is the ignition of Z9 and eventually, the AVUM phase, in order to transfer the payload to an intermediate orbit. AVUM, usually, performs three ignitions:

- the first one for reaching the intermediate orbit;
- the second one for the circularization;
- the last one for the deorbiting or disposal;

The typical ground path of the Vega can be seen in the figure 2.5, where all the ignitions are indicated. In particular, the three AVUM boosts can be identified and the coasting phase between the second and the last ignition, FC11 and FC12 respectively. Vega, and in a similar way also Vega-C, is capable to carry more

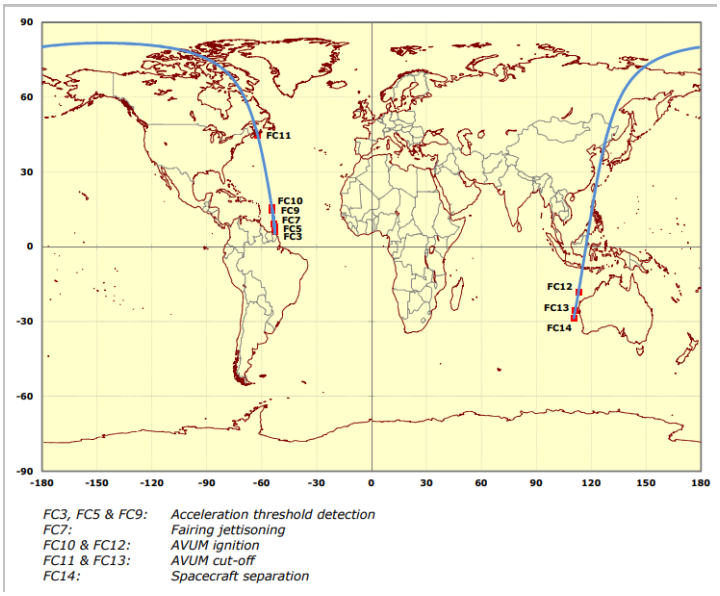

Figure 2.5: Ground path Vega mission[6]

than one single payload in orbit through a new system deveolped by AVIO called SSMS, the Small Spacecraft Mission Service [8]. Thanks to this system, the Avio launchers are capable of carrying multiple payload and put them in different orbit. The SSMS considers different configuration in the arrangement of the satellites on the adapter, in the figure below some of them are shown. As it can be noticed there is always a main payload and the others are smaller: considering the satellite's dimensions, the more suitable configuration can be choosen.

The introduction of this system modified the mission profile because of the impossibility to put each payload on different orbit. Thus AVUM passes from 3 to 5 ignition, each of them allowing little changes on the orbit parameters in a

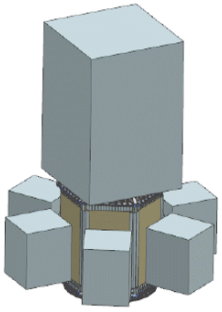

Figure 2.6: HEX-1 configuration

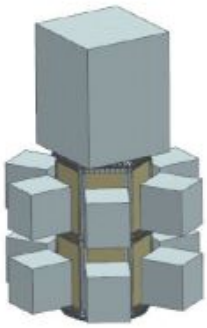

Figure 2.7: HEX-2 configuration

Figure 2.8: Plat-2 configuration

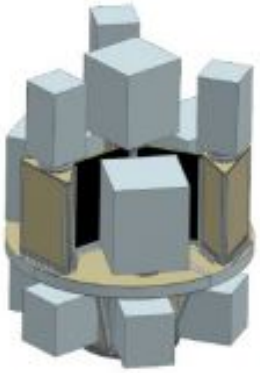

Figure 2.9: FLEXI-3 configuration

way that the payload will not collide. In the figure 2.10 it can be noticed that the ascent profile is a little different from the one of a classic Vega mission, shown in figure 2.4. First of all, as expected, the duration is longer in the first case because it has to perform more ignition and the mission duration is obviously longer, there is also some difference in altitude but, in general, there are not so many variations allowing the great advantage of a multipayload system like this.

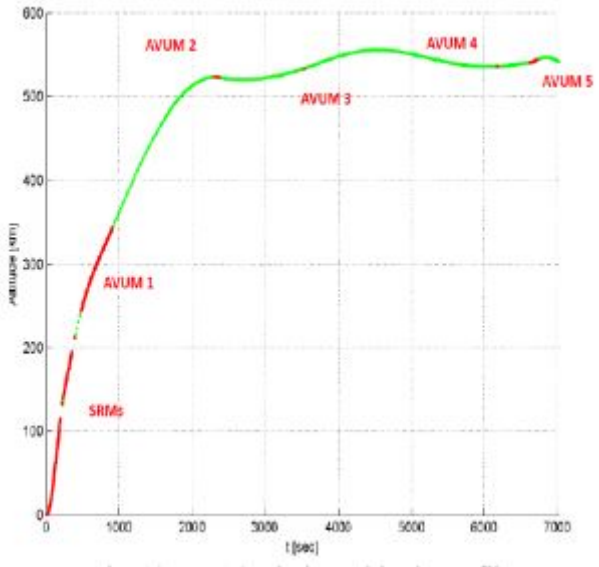

Figure 2.10: Vega altitude profile with SSMS [8]

Chapter 3

Physical model

The implemented tool is based on the physical model of a 3-DOF ballistic reentry, so it is not considered the thrust, that can be modeled with some assumption in the 3 DOF model, and the asset control. The simulator has to model and predict the fragments trajectories in order to find the on ground footprint and compute the risk for the population that lives in that area. The fragments can be generated by a disruptive event or by the burning of the main body in the atmosphere and they fall on ground according to the Newton's dynamic second law. Before examining the details of the equation of motion, the hypotesis, taken into consideration, are shown below.

1. The effect of the gravity is implemented through the J2 coefficient, which considers Earth as a geoid. All the others harmonics have not been taken into account because of their negligibility in this type of problem.

2. The athmospheric model that is undertaken is the *NRLMSISE-00* one, but all the three models have been implemented on the previous versions of the tool and they are: the *NRLMSISE-00*, the *ISA* model and the *CIRA-86* one.

3. Regarding the C_d, it is included in the ballistic coefficient(C_b) and is considered constant during all the path, except for that part of the trajectory where the motion becomes subsonic, here the C_b is evaluated as half of its value. This hypotesis is due to the impossibility to know the exact C_d during all the trajectory with the variation of Mach. It is important to note that the ballistic coefficient is defined, in all this work, as follows, but it has to be also underlined that in the input file the ballistic coefficient has to be defined as the opposite:

$$C_b = \frac{C_d S}{m} \tag{3.1}$$

where the S and the m are, respectively, the reference surface of the fragment and its mass.

4. This model includes the lift and the possible bank angle, taken into account as constant input if a nominal trajectory is implemented, in other cases the bank angle is a random value.

3.1 Reference frame

First of all it is necessary to define in which reference frame the tool will work. There are mainly three reference frames, as shown in figure 3.1, that need to be taken into account in order to understand how the tool works:

1. the ECI reference frame
2. the ECEF refereence frame
3. the ECI Avio reference frame

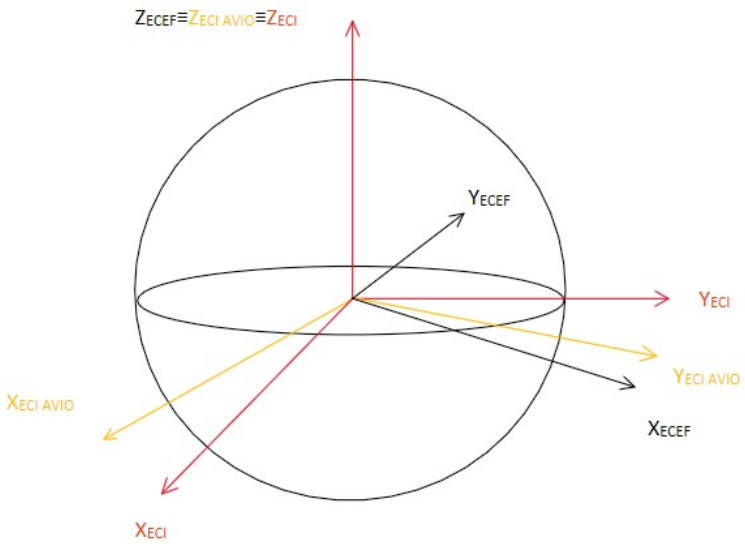

Figure 3.1: Reference frames

The first coordinate system is the ECI one, the Earth Centered Inertial reference frame, the x axis points towards the Aries constellation, the z one is defined with outward from the north pole, while the y completes the triad. It is the reference

frame most used when the quantities are determined, the tool uses the state vector defined in ECI Avio, a special ECI reference frame fixed to 2 seconds, this means that, with respect to the standard ECI, the reference frame is rotated of minus two seconds compared to the direction of aries constellation. At the end, the most useful reference frame for the purpose of the reentry tool is the ECEF one. The ECEF is the Earth Centered Earth fixed reference frame, is defined fixed with the Earth: the x axis is fixed on the Greenwich meridian, the z axis is the same as the ECI one and the Y completes the triad, in this way the ECEF coordinate system rotates with the Earth.

3.2 Equation of motion

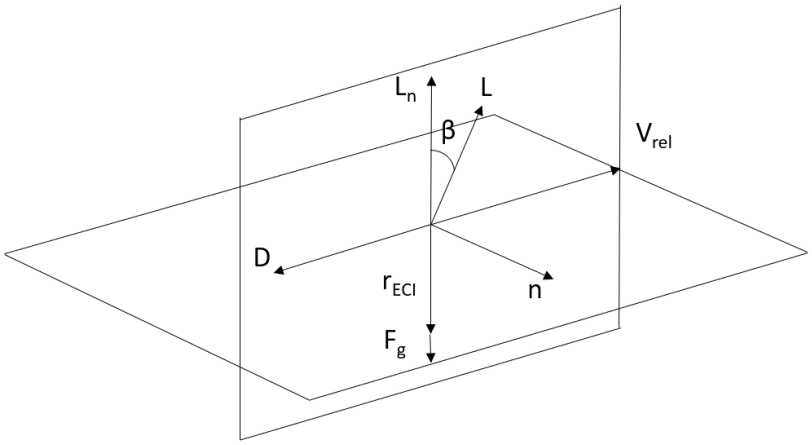

Figure 3.2: Free-body diagram

In the figure 3.2 the forces involved in the simulation are shown. First of all it has to be noticed that the diagram is oriented along the direction of r_{eci}, in that direction there is the gravity force while the lift is perpendicular to the relative speed in ECI reference frame, if the bank angle is not considered the lift is on the opposite verse with respect to the r_{eci} and eventually the aerodynamic drag in the opposite verse of the relative speed in the ECI reference frame. The equation that has to be integrated is the Newton's second law here illustrated:

$$\ddot{r} = \frac{F_g}{m} + Drag + Lift \tag{3.2}$$

where Drag is the aerodynamic force due to the atmospheric drag per unit of mass

defined as

$$Drag = -\frac{1}{2}\rho V^2 C_b \frac{\vec{\mathbf{V}}}{V} \tag{3.3}$$

While for the drag the definition of the direction is straightforward, the direction of the lift is more complicated, specially because, in a problem of this kind, the wind axes are difficult to identify. The situation is shown in figure 3.2. Here, it can be seen that the unit vector $\vec{l_n}$ is out of the plane identified by the velocity vector in ECI reference frame and the vector n, that is the normal to the plane identified by the relative velocity vector and the position vector, in particular they can be written as:

$$\vec{n} = \vec{r}_{ECI} \wedge \vec{v}_{rel_{ECI}}$$

while

$$\vec{l_n} = \vec{V}_{rel_{ECI}} \wedge \vec{n}$$

After these passages, it is important to notice that the lift vector can be on plane in every direction, as shown in figure 3.2: in order to identify it, it will be necessary to define the bank angle. Owing to this definition, the lift vector direction can be found out:

$$\vec{l} = Mat * \begin{bmatrix} 0 \\ sin(\beta) \\ cos(\beta) \end{bmatrix}$$

where Mat is defined as the matrix which columns are the velocity vector, the n direction and the l_n direction.

On the same basis, the lift has been defined as:

$$\vec{L} = -\frac{L}{D} * Drag\vec{\mathbf{l}}$$

The last force that has to be analaysed is the gravity one. This one is defined through the WGS-84 model. The model was updated in 1984 by the US Department of Defense[9], in order to have more accurate data from the navigation satellites. It defines a new coordinate system based on the NNSS one, in particular it considers two rotations and one traslation on the z axis of this last reference system. This model consider also an update of the main Earth parameters: the semimajor axis, the gravitational constant, the second zonal harmonic and the angular velocity. This part is the one that is taken into account by the GRAV function and by the geodetic routine inside the tool. The model considers the Earth as an ellipsoid with a semimajor axis of 6378137 meters, a gravitational constant of 3986001 m^3/s^2, the normalized second degree zonal gravitational coefficient as $-484.16685 * 10^{-6}$, defined as $\frac{-J_2}{\sqrt{5}}$, where $J_2 = 108263 * 10^{-8}$ and, eventually, the angular velocity, that has been undertaken in all reference frame rotations, equals to $7292115 * 10^{-11}$.

3.3 Resolution of equation of motion

The objective of the tool is computing the risk for the population through the computation of latitude and longitude of the impact point. It is necessary to integrate the equation of motion in order to obtain the state vector in a specific instant of time, then it will propagate the results on the whole trajectory till ground. The integration has to be done through numerical method, the choise is between a customize integrator or ODE function of Matlab, after some consideration the choise went on the first option. This decision is mainly due to the modularity requirement and to the easily debugging of a custom integrator, evidently it is accepted the numerical error introduced. This last evidence will be shown in the validation chapter.

The numerical integration can be performed in various ways, there are some methods that are explicit and others that are implicit. The one choosen for this tool is the most used of the explicit methods, the order 4 Runge-Kutta one. Comparing this method to other it results to be the optimum between stability, computational cost and precision: all Euler's method are much less accurate while the other Runge-Kutta method are too expensive in terms of cost.

Chapter 4

Propagator module

As mentioned in the previous chapter now it will be a part dedicated to the module of the propagator. It is defined as a matlab function, in this way it will be available also for other application. It has been implemented in two different ways: the first one is for the propagation before the fragmentation, the second for the propagation of the fragments. The difference between the two routine is a flag that is switched on when the fragmentation altitude, written in the input file by the user, has been reached. Both the routines use the numerical integration method of Runge-Kutta. From a theoretical point of view, the Runge-Kutta method is an explicit numerical method that computes the approximate solution of the considered function. In this case the differential equation that has to be integrated is the equation of motion while the solution is the state vector in a specific instant of time. In general the method integrates follwing steps below:

- First of all, the time-derived of the state vector has to be defined, that is function of the time and the state vector it self: $\dot{S} = f(t, S)$
- Then it is necessary to define the initial state vector $S(t_0) = S_0$
- The solution at the ith instant of time can be approximated as: $S(t_i) \simeq \omega_i$
- The following quantities are the slopes of the solution in the specific point, where h is the integration step:

$$k_1 = h * f(t_i, \omega_i) \tag{4.1}$$

$$k_2 = h * f(t_i + \frac{h}{2}, \omega_i + \frac{k_1}{2}) \tag{4.2}$$

$$k_3 = h * f(t_i + \frac{h}{2}, \omega_i + \frac{k_2}{2}) \tag{4.3}$$

$$k_4 = h * f(t_i + h, \omega_i + k_3) \tag{4.4}$$

- At the end, the approximation of the solution is: $\omega_{i+1} = \omega_i + \frac{1}{6} * (k_1 + 2k_2 + 2k_3 + k_4)$

In the code these are the steps that are followed in order to find the state vector at any time of the time interval. In particular the equation of motion is called in the Runge-Kutta routine five times, four of them are for the integration and the last one is for the definition of the other quantities such as the altitude, Mach and the dynamic pressure, that were used for validation and control of the results. One of the most important variable to set in an integrator is the managment of the time step,here called h: it is fundamental for the precision of the result. By default is fixed to 0.5, but it can be reduced in order to obtain more accurate results. The most simply way is consider the step as fixed, but when it arrives in proximity of the ground it is possible that it will not stop the execution exactly to zero but maybe it will become minor. In order to avoid these result, it has been decided to consider a variable step that will be thicker at low altitudes, in this way it will stop when it reaches altitute equal to zero. However the amplitude of the step can be decided by the user, as an input, along with the starting time and the end time of the simulation.
The variable step is an altitude control when the fragments reach the ground and also whene the main body is near the fragmentation altitude, in order to be more accurate. There is also another kind of step control depending on the acceleration of the object, if it is too high the step will be reduced.

4.1 DYN routine

The core function of the integrator is the so called DYN function. It is the module that represents the equation of motion and it computes all the forces that are involved in the model. This function takes as input the state vector, the INPUT structure, that contains all the information about times, the atmosphere information, tables mainly, and the information about fragments,through the FRAG structure. It returns as output a six dimension vector, made up of acceleration and velocity bounded to the integration step.

In "DYN" function there is a subroutine, called geodetic, that implements the altitude, the longitude and the latitude of the current state, and, thanks to these information, the density can be computed through various model of the atmosphere. This function has as input the position vector in the ECI reference frame and the Earth structure, where there are all the information about the involved central body, in this case the Earth. This function is also called outside the routine when the impact points have to be computed. In this module the forces described in the previous chapter have been implemented and computed.

4.2 Atmosphere models

There are a huge amount of atmosphere model available, however there are three of them that are the most implemented ones: the ISA model, the NRLMSISE-00 model and the CIRA-86 one. The atmosphere configuration employed in the final version of the tool is the NRLMSISE-00 one. A file is created from the NRLMSISE-00, considering each month of the year and split it on latitude, height and longitude. The file obtained is a MATLAB file, *.mat*, under the form of a matrix of dimensions: 37x19x601, respectively the longitude, the latitude and the heights. Also a text file can be generated, but, due to the dimensions, it would be necessary averaging the quantities on the longitude band.

The explanation of choosing this type of model, instead of one of the others defined in the following sections, is linked to the limitations of them and to the fact that, in order to not use the NRLMSISE00, over the 120 km of altitude the atmosphere was considered absent. The only constraint is bounded to the tough hypotesis on longitude values and selected year:

- the assumption on longitude considers the same condition on every longitude degree, if the text file is selected, in case of the mat file this assumption is not made;
- the year considered is 2023, apart from the year entered in the input file: this could lead to error if the year considered has a huge solar activity that could influence the atmospheric quantities;

The use of the combination of model has been also investigated, but where two models are bound up, some discontinuities could be underlined. These are due to the fact that the models have to be linked to each others and the values in the link are not the same. In conclusion, the result is that is more convenient to choose only one atmosphere model, in this way some discontinuities can be avoided.

4.2.1 NRLMSISE-00

The most accurate, but also the most expensive in terms of computational cost, is the NRLMSISE-00 model[10]. This is an empirical model that is able to compute the denisites of the single components of the atmosphere and also the temperature at each altitude, from the exosphere to ground. The NRLMSISE-00 model database looks upon space weather data, that can be available on the NOAA site. These data are shown in a text file formatted as in figure 4.1. Specifically there are for each day of each year, from 1957 to today, the values for Kp,Ap and F10.7, there are more measures because they are considered every three hours, from midnight of the current day to the after one. The so called Kp, is the planetary index which

considers the magnitude of the geomagnetic storm and it indicates the disturbances of the terrestrial magnetic field. The Ap is the planetary equivalent amplitude and derives directly from the Kp index, it measures the amplitude of the magnetic activity. The last index, the F10.7, is the solar radio flux at 10.7 cm, it means at 2800MHz wave frequency, it indicates the intensity of the solar activity in a specific day.

```
#
#                                                                                         Adj     Adj   Adj   Obs   Obs   Obs
# yy mm dd BSRN ND Kp Kp Kp Kp Kp Kp Kp Kp Sum Ap  Ap  Ap  Ap  Ap  Ap  Ap  Ap  Avg Cp C9 ISN F10.7 Q Ctr81 Lst81 F10.7 Ctr81 Lst81
# ----------------------------------------------------------------------------------------------------------------------------
#
NUM_OBSERVED_POINTS 2130
BEGIN OBSERVED
2018 01 01 2515 21 33 37 23 23 27 10 10 13 177  18  22   9   9  12   4   4   5  10 0.6 3   0  66.8 0  69.3  70.8  69.1  71.4  72.5
```

Figure 4.1: Space weather data example[11]

4.2.2 CIRA-86

The Cospar International Reference Atmosphere (CIRA) model[12] is a collection of table that allows to compute some atmospheric parameters between 120 km and 0 km of altitude. The version implemented in the tool is the one of the 1986, it is made up of three tables for each month of the year: the mean temperature one, the mean zonal wind and the mean pressure as function of the latitude and longitude. Regrettably, the CIRA-86 model has some limitations:

- the latitude range: it considers data only between -80° and +80°;
- some data are not available, specially the ones in proximity of the range limit latitude.
- there are no pressure data available between 20 and 0 km of altitude.

Although this limitations, the model is more accurate than ISA model and less expensive, from the computational cost point of view, than NRLMSISE-00 one, with which it bounds up with at 120 km. Due to the caractheristic of CIRA model, it was implemented in the tool only between the 120 km and 20 km, out of this range the ISA model is considered. This solution is not the definitive one, but one of the first version of ORCA.

4.2.3 ISA

The International Standard Atmosphere is the model developed by the ISO and it is valid up to 80 km. In the first versions of this tool the model has been implemented up to 20 km, The International Standard Atmosphere is the easiest model based on a linear distribution of the temperature through the various level

of the atmosphere, as it can be seen on the figure 4.2, while the pressure is related to the geopotenital altitude. The ISA table is defined by ICAO and it includes, for each altitude, pressure, density, temperature and the sound velocity. The code is able to read these quantities after the identification of the altitude. In the tool the model is implemented in order to define the density under 20 km of altitude and above 120 km in substitution of NRLMSISE-00. From the perspective of the computational cost it is the less expensive and it is also the solution that requires less input: it is not necessary to know the day of the year in which the simulation will be performed.

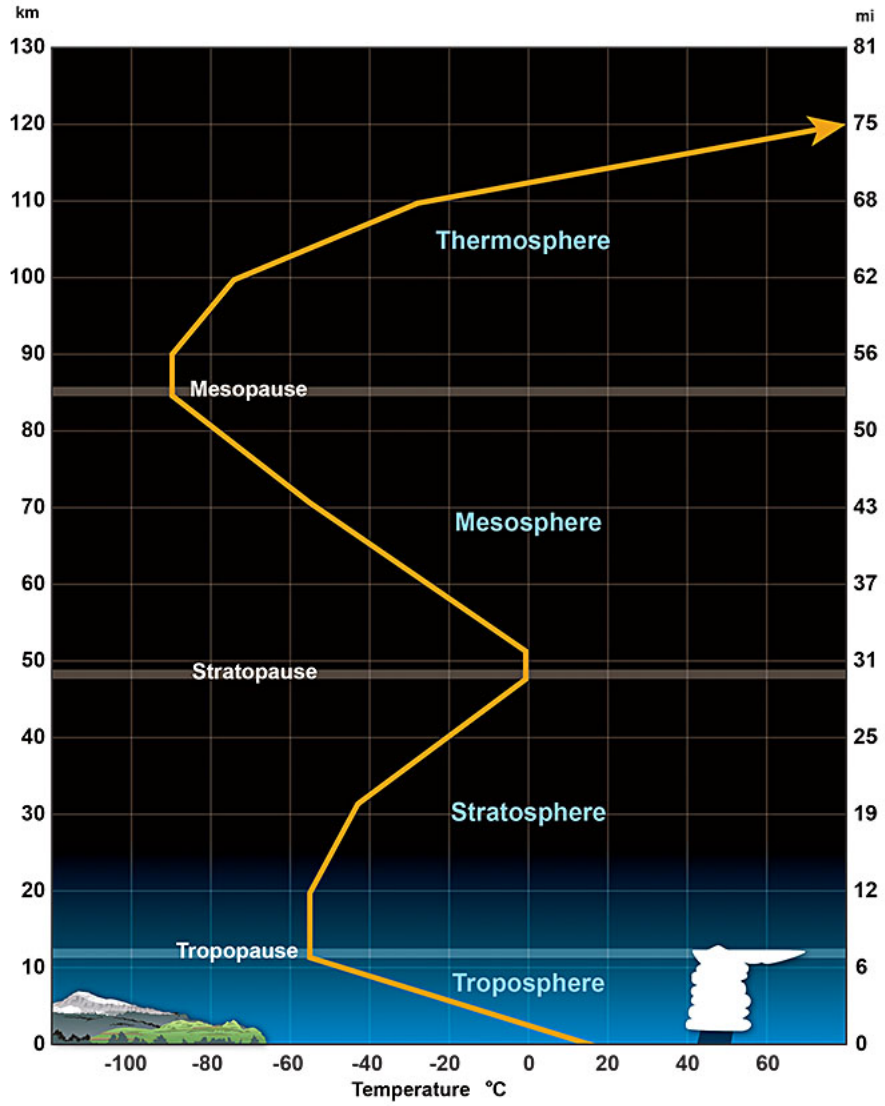

Figure 4.2: Temperature trend and atmosphere levels [13]

4.3 Fragmentation model

In a re-entry problem there is a body that return on Earth surface, when it reaches the densers layers of the atmosphere it is under high thermal and structural loads and some pieces may break away from the main structure. These pieces, that will be called debris, can reach the terrain and could represent a risk for the population. The tool has to take into account the break-up scenario in order to control the on-ground risk, thus some lines of the code have been implemented to consider the quantities related to the fragmentation event, together with the features of each fragment that are indicated in the input file called fragment file. These quantities are:

- the fragmentation altitude: usually, during the re-entry, the altitude at which the loads become critical is around 120 km, in the tool this quantity can be set in the input file by the user. Till this altitude is not reached, the tool propagates the trajectory of the initial object whose features are in the input "before fragmentation" file. This quantity is relevant only for the de-orbiting module, the re-entry object in the launching one is implemented as a collection of fragments.

- the ejection velocity: is the value related to the velocity with which the debris are ejected from the main body. The break up indeed can occur because of a failure that can lead to an explosion, in this case the debris will be ejected with an initial velocity that can be set, also in this case, by the user in the fragmentation file. The direction of this velocity is not set because, after the blast, it is randomic and it is taken into account in the Monte Carlo analysis as variable.

Chapter 5

Simulation modules

The tool is divided in two modules, that allows to run two differents simulations. They are the launch module and the reentry one capable of computing the collective risk. The steps of the simulation depends on which kind of propagation should be run. There are two different choices: the nominal propagation, without the uncertainties, and the scattered simulation, for each module. in the last version of the tool two other choises can be made in order to not consider the initial state vector scattering. Starting from the nominal simulation, the various steps are listed:

1. the propagation of the trajectory of the main body, before the fragmentation altitude;

2. the propagation of every fragment's trajectory that is generated from the primary body, for both the modes;

3. eventually, the part that computes the on ground risk from the impact points;

If the scattered simulation has to be run, in both cases, the MC simulation has to be chosen. The steps of this kind of simulation are:

1. first, the nominal propagation of the primary body with the covariance matrix propagation on the initial state vector, if it is taken into account;

2. when the cloud reaches the fragmentation altitude, the previous simulation stops and starts the Monte Carlo analysis until all the state vectors touch the break up altitude;

3. at the end there is the propagation of each fragment, with scattering on the atmospheric and aerodynamics quantities, until the hitting of the ground.

Obviously, both the modules, whatever is the simulation chosen, are capable of computing the on-ground risk.

Considering how the code operates, first of all, the tool reads the input files that the user has to provide in the format specified in the Appendix A. There are different types of file that differ from each other depending on the chosen module. The input files that the user have to provide are:

- the path file in which there are the file paths for the simulation.
- the simulation file, where there are defined the quantities for the propagation, for example times, the different kind of altitudes or the number of Monte Carlo.
- the atmosphere files, the fragment's one, the population density file, the reliabilty file, if the launch module is chosen, the file with informations about the primary body before the fragmentation.

The features of the bodies are used to compute the reentry trajectory in the propagator module or in the function ode, that is the equivalent of the DYN function of the integration module. The tool performs several propagations because of the uncertainties bounded to some quantities: in the moments before the break up, the uncertainties are linked to the initial state vector, they will be propagated through the covariance matrix, in this way the computational cost remains acceptable. After the break-up the uncertainties are tied, mainly, to the atmospheric quantities, pressure and temperature, and they are managed with the Monte Carlo simulations.

After the integration, an impact point is identified and the computation of the risk can be carried out. The on ground point allows the identification of the population density in that specific area and, consequently, the computation of the risk as indicated in the introduction of this work. The tool has also a post processing capacity of generating the density map of the population, given the file from SEDAC described in the Appendix C. Here it is reported the density population map used in the tool. It must be noticed that the legend bar is on a logaritmic scale and the red zones are the most densely populated ones and they correspond to the north of India and China, as expected. From the post processing, the impact points can be dislpayed on the Earth map, the risk by latitude can be also shown in order to identify the regions with a more higher risk.

In the following sections the different module will be analyzed.

5.1 Risk during launch

This module is able to compute the collective risk on ground as consequences of a failure event during launch operations. It considers as input a trajectory file, in text

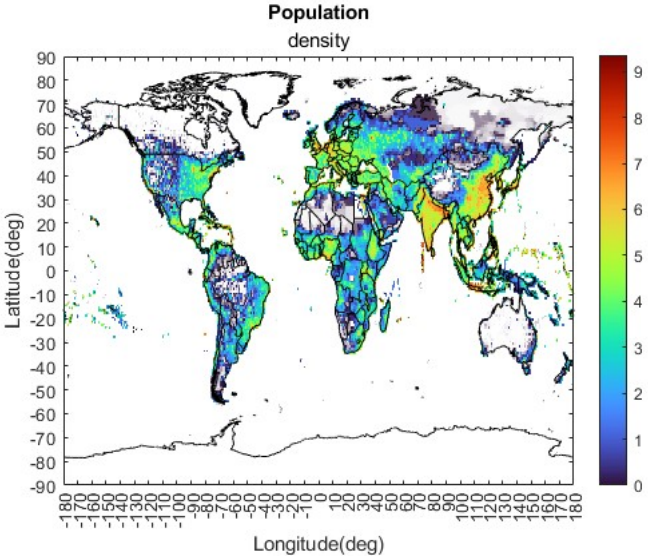

Figure 5.1: Density map referred to 2000

format, along with the fragmentation file and the file with simulation parameters. The tool is capable of reading each row of the file and propagate the associated state vector, if the altitude is under the fragmentation altitude it propagates the fragmented body directly, otherwise it propagates the main body till the break-up altitude and then its fragments. After that it computes the points on ground with their associated risk, taking into account the unreliability in that specific instant, given in the form of excel table from the user.
The integrator of this module is the same as the de-orbiting one, but it can be implemented also with the ode23s MATLAB's function, or ode45, and with Runge-Kutta 4 with control on numerical error. The integration module considers a variable step propagation with a control on the altitude and on the acceleration. It was noticed that, in specific cases when the deceleration was high, the propagator starts to diverge: a particular numerical control was necessary. Thus it was decided to introduce the control on the step bounded to the acceleration.
Both the checks have a coefficient that allows the computation of the necessary step, and then they controls that this step is not less then the minimum one, set by the user in the input file. The stop control is given by two different conditions:

- the final time has been reached;
- the reference altitude has been reached;

If one of these conditions are attained the propagation stops and gives the outputs, it is important to notice that if the final time is reached the outputs could be faulty,

in this case the tool is not capable of identify the faulty points but it could be improved adding lines that can isolate these points.

In this ORCA's module, some hypotesis have been introduced:

- the simulation stops when the perigee's altitude is 20 km. After this value, indeed, the body can be considered as orbital and it does not make sense to propagate its re-entry because there is a very high uncertainty, it is necessary to consider an indirect reentry.

- there is a stop altitude, considered as the ground from the tool, sets to 20 km. This choise enables the speeding up of the code without introducing significal errors: from that altitude the object's falling can be considered as vertical. In order to compute the risk in a correct way the velocity taken into account in this case is not the one to 20 km but the limit speed defined as $V_{lim} = \sqrt{\frac{2*g}{\rho * C_b}}$ where g is the gravity acceleration, ρ is the atmospheric density and C_b the balistic coefficient.

Thanks to these semplification, the tool takes only 5 minutes for running against the 12 before the speeding up. The execution time can be improved by the user through the setting of the interval of time of the integrator, in the simulation file. The user has to be warned that an integration step major of 60 seconds is not advised because the tool can be slowed down by the huge amount of controls on the step's variation and the risk to stumble upon on numerical instability. The table 5.1 shows how the computational time changes with the integration step. It has to be noticed that if the integration step is too high, this means from 60s and forward, the minimum step has to be changed. In order to control the speed of the code the user can change the minimum step, increasing it, up to 2 seconds. The initial

Integration step	Computational time
20 s	549.705 s
40 s	536.258 s

Table 5.1: Differences on the computational time

phase of the launch is not considered in the simulation: the first phase is controlled and if there is a failure, the system will be blown up. Thanks to this operation the computational time can be reduced by reducing the number of simulations. In order to find the starting point, the tool computes firstly impact points using the analytic impact's hypotesis, when an impact point on a populated land is identified the propagation as described so far is performed. Considering the initial control of the launcher during the first instants of the launch, this module uses as first computation an analytic way of identifying the impact points and when there is

a point on populated land the simulator starts the propagation. this method is shown in the chapter 7.
After the definition of the functions some simulations with the MATLAB's profiler have been performed. In these way some unecessaries calling to some functions have been identified and deleted. In the table 5.2 there are the computational time of the launching module after the optimization, considering also the introduction of the step control in the propagator module. As it can be noticed in the table the nominal times reduced of about a half and they can be reduced from the managing of the minimum and maximum step. In this module it is also possible to compute

Maximum integration step	Minimum integration step	Computational time
20 s	0.5 s	230.815 s
40 s	0.5 s	221.255 s

Table 5.2: Differences on the computational time after the speeding up

the risk when the launch veichle is still on the launch pad, in this way it is possibile also to consider the safety area for the launch operations. In order to obtain this results the user has to disable the analytic impact control and he has to run the simulation from the beginning of the trajectory. The tool will not take into account other kind of risk, such as the one linked to explosion or toxic exposure, they need another kind of analysis that has to be performed with other softwares.

5.2 Orbital reentry

Sometimes it is necessary to comupte the risk on ground due to a reentry object and this has to be treated differently than a launch risk. In these cases it is not possible to have a reference trajectory because, most of the times, this kind of reentry are uncontrolled, they usually are the consequences of a failure during the deorbiting manoeuvers. Thus the computation will take into account the ballistic trajectory before the break-up altitude and, then, the fragments will be propagated. The propagation is performed by the Runge-Kutta propagator of the forth order explained in the chapter 4. The results are obtained in the same way of the launching module.
The main difference between the two modules is the input file structure while the working flow is the same. At the beginning of this work the main difference was about the uncertainities: regarding the launching risk module there were considered only the ones on the atmospheric and aerodynamics quantities, while the deorbiting considered also the scattering on the initial state vector. After the validation of the launching module, it was decided to take into account also the initial scattering, this decision causes a growth of the computational time but it can be improved

in further iterations and versions of the tool. The steps of the simulation for the orbital reentry module, before the fragmentation altitude, can be illustrated by a clear representation shown in figure 5.2:

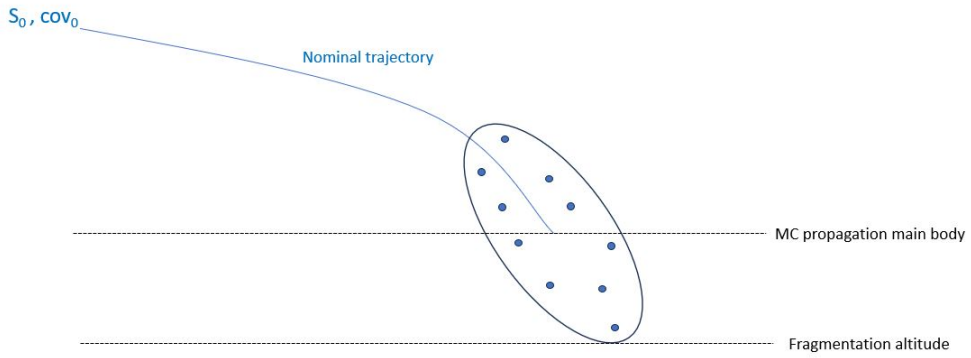

Figure 5.2: Working flow graphic representation

Especially the simulation starts from an initial state vector and the covariance matrix that gives information about the uncertainties on the six components of the vector, then ORCA propagates these quantities until the covariance matrix amplitude reaches the stop altitude, decided by the user. At this point of the simulation the nominal trajectory is above that altitude, in this case it is at 240 km, the scattered vectors are generated and they are propagated up to the fragmentation altitude. In the figures shown below, the altitude of the state vector randomly generated are displayed. As it can be notice from figure 5.3 the cloud of state vectors is centered about 240 km and its amplitude reaches 78 km, the fragmentation altitude set for this simulation. In particular the end of the covariance propagation is when the 3.1σ reached the fragmentation altitude: this means that there is a probability equals to 99.9% that the randomic state vectors are above the fragmentation altitude. From the figure it can be noticed that there is only one vector below that altitude, as expected. From this point of the simulation, in both the modules, the vectors above the fragmentation altitude are propagated by a Monte Carlo analysis till the fragmentation altitude set by the user in the input file. The outcomes are shown in the figure 5.4. As it can be seen the most of them stops to 78 km, the ones that do not reach that altitude are the ones that needs more time to be propagated. It is rememberd that the maximum number of integration step is managed from the end time of the simulation in the input file.

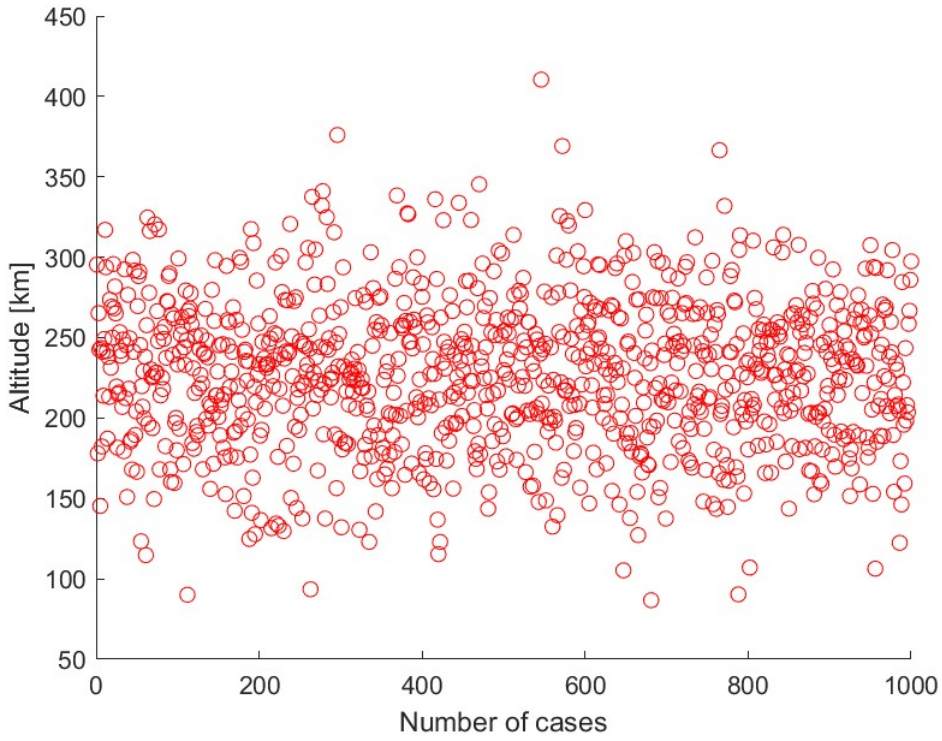

Figure 5.3: Cloud at the end of the covariance propagation

5.3 Covariance propagation

Regarding the error propagation, the deorbiting module is more complicated than the launching one, this is linked to the uncertainties on the initial state vector. In the de-orbiting module the user has to give a text file with an initial uncertainties deriving from the running of 1000 Monte Carlo's simulation, in this way the initial state vector, written in the input file, has an initial scattering. In order to speed up the code, it was decided to propagate the covariance matrix, through the computation of the transition state matrix. The selected method to propagate the state tranistion matrix is the Merkley's one [14]. The method considers two instants, with a little difference such as 0.5 s, and linearized the problem: it works well only when the problem can be linearized, infact it has been choosen to apply it only in the upper part of the reentry trajectory, where there is no atmosphere and the problem is linear. This method considers the Taylor series expansion stopped to the second order and it takes into account the effect of Earth's flattening through J_2 factor. The expression taken into account in this way becomes:

$$\Phi(t, t_0) = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} 0 & I \\ G_0 & 0 \end{bmatrix} \Delta t + \begin{bmatrix} G_0 & 0 \\ \dot{G}_0 & G_0 \end{bmatrix} (\Delta t)^2/2 \tag{5.1}$$

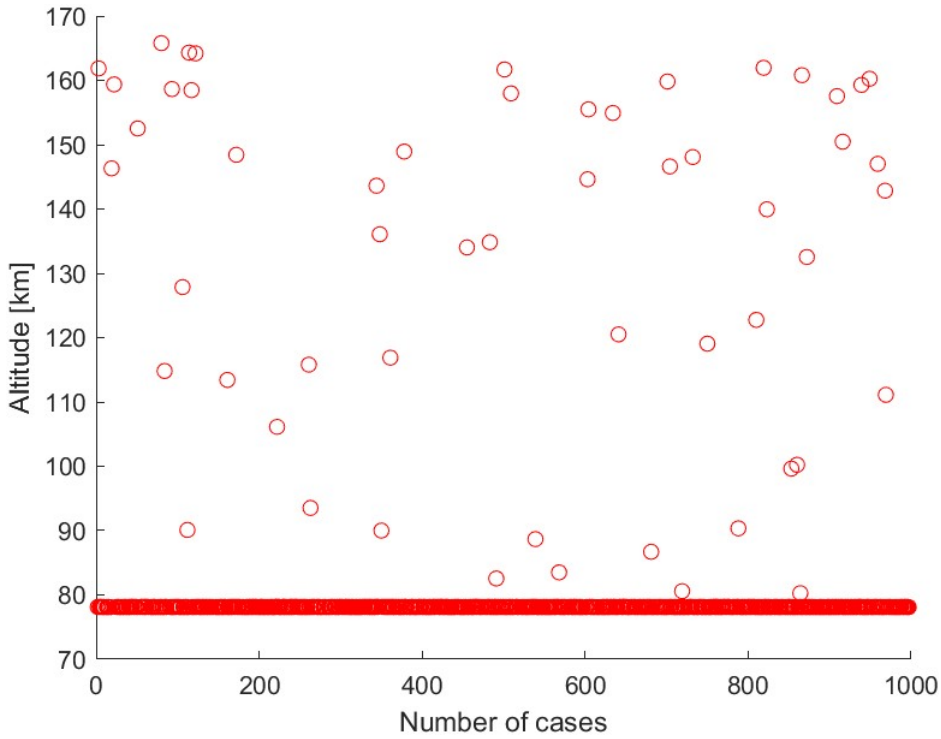

Figure 5.4: Altitude of the state vectors propagated through the Monte Carlo analysis

From 5.1 the components of Φ can be pointed out as follow:

$$\Phi_{11} = I + (2G_0 + G)\frac{(\Delta t)^2}{6} \tag{5.2}$$

$$\Phi_{12} = I\Delta t + (G_0 + G)\frac{(\Delta t)^3}{12} \tag{5.3}$$

$$\Phi_{21} = (G_0 + G)\frac{(\Delta t)}{2} \tag{5.4}$$

$$\Phi_{22} = I + (G_0 + 2G)\frac{(\Delta t)^2}{6} \tag{5.5}$$

The quantities that appears in the previous equation are:

- the identity matrix, I;
- the gradient matrix, G, of the acceleration, considering only the effect of J_2 and the gravity's one.
- the interval of times between two instants of integration, Δt

The speeding up of the code can be noticed in the calculation of G matrix: it uses quantities already computed in previous integration steps, so it does not require any further calculation. In particular, the components of G [15] are the partial derivates, in x,y and z, of the quantities:

$$a_x = -\frac{\mu x}{r^3}[1 + \frac{3J_2R_{eq}^2}{2r^2}(\frac{5z^2}{r^2} - 1)] \tag{5.6}$$

$$a_y = \frac{y}{x}a_x \tag{5.7}$$

$$a_z = -\frac{\mu z}{r^3}[1 + \frac{3J_2R_{eq}^2}{2r^2}(3 - \frac{5z^2}{r^2})] \tag{5.8}$$

where μ is the mass parameter of the considered planet, in this case the Earth, R_{eq} is the equatorial radius of the Earth and r is the position vector. The partial derivatives has been computed through the MATLAB's symbolic toolbox and then they are considered in the G matrix in the following way:

$$G = \begin{bmatrix} \frac{da_x}{dx} & \frac{da_x}{dy} & \frac{da_x}{dz} \\ \frac{da_y}{dx} & \frac{da_y}{dy} & \frac{da_y}{dz} \\ \frac{da_z}{dx} & \frac{da_z}{dy} & \frac{da_z}{dz} \end{bmatrix} \tag{5.9}$$

Then the equation 5.2 are used to compute the state transition matrix at any istant of time of the propagator module. In particular the transition matrix has the following structure:

$$\Phi = \begin{bmatrix} \phi_{1,1} & \phi_{1,2} \\ \phi_{2,1} & \phi_{2,2} \end{bmatrix}$$

After the computation of the state transition matrix some attempts have been made. Firstly the matrix has been propagated until the break-up altitude was not reached, at that altitude the covariance matrix was computed in order to obtain the scattered state vector, necessary to run the Monte Carlo simulation on the fragments.
The covariance matrix is defined as a 6x6 matrix where on the diagonal there are the square of the variances and, off-diagonal, there are the covariances:

$$Cov = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}$$

Obtained this last matrix as[16]:

$$Cov = \Phi' Cov_0 \Phi + Q_d \tag{5.10}$$

where Cov_0 is the initial covariance matrix from the initial scattering on the state vector, Φ is the state transition matrix at the fragmentation altitude and Q_d a noise factor taking into account the little uncertainites.
But doing that, some discrepancies with the Monte Carlo analysis were observed, thus some other solutions were investigated. First of all, it was noticed that if an orbital problem was considered, the covariance propagation works quite well. This happens because of the hypotesis behind the state transition matrix propagation listed before. The major problem was that the last covariance matrix, corresponding to the nominal trajectory reaching the fragmentation altitude, was too large and, consequently, whene the state vector were randomly generated, they reached very low altitude, some of them also the negative ones. Afterwards, the idea was to intersect the covariance cloud with the plane at the fragmentation altitude, in this way only the state vector belonging to that plane would be taken into account and, eventually, they could be compared with the Monte Carlo results stopped at that altitude. The result was that they did not coincide again, so it has been supposed that it is wrong to propagate in time the covariance matrix and then pass it to the space domain. The solution adopted at the end is the following:

- the matrix propagation is run until the 3.1σ reaches the fragmentation altitude and the nominal one is obviously above;
- the randomic state vectors are generated, the number can be set by the user in the input file, at the Monte Carlo's number section;
- Until the fragmentation altitude is not reached, the complete body is propagated from the state vectors generated through the Monte Carlo method;
- When the break up altitude is caught up, their propagation starts.

It is important to notice that the value of the amplitude of σ is set from the user and is advised to not change this value owing to the preservation of the best connection between the accuracy and the computational time. Choosing 3.1σ allows to set the probability accuracy of having an outcome vector, with the desired features, equals to 99.99%. In order to validate this method, both the simulations have been run and their results are based on 1000 state vectors. In the following histograms the obtained results are shown. With the aim of obtaining these results, the following data were used:

- the minimum time step is set to 1 s;
- the initial time step is set to 30 s;
- the end of simulation at 10000 s;

- the initial state vector set by an internal trajectory file, chosen according to the reentry necessary condition on the perigee altitude.

It is important to notice that these simulations consider only scattering on the initial state vector and not on the other quantities, as opposed to the Monte Carlo described in the following section.

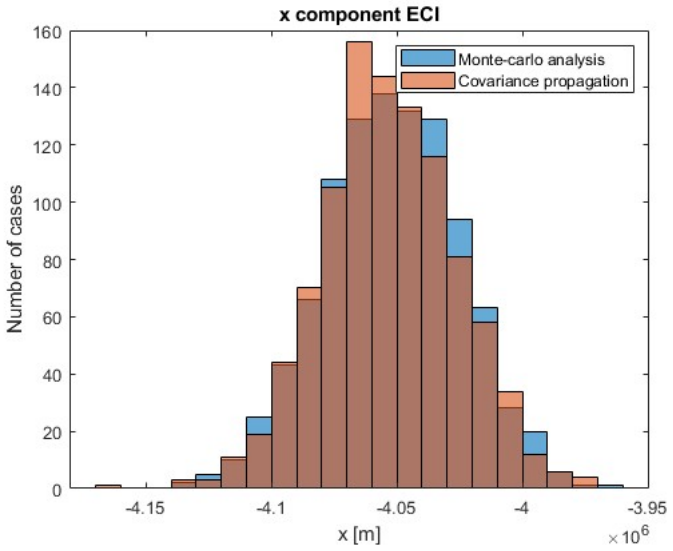

Figure 5.5: Variance on the x component of the state vector

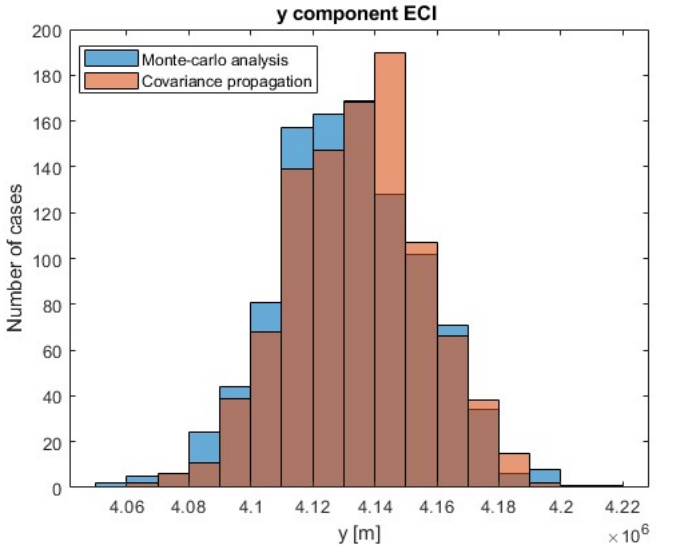

Figure 5.6: Variance on the y component of the state vector

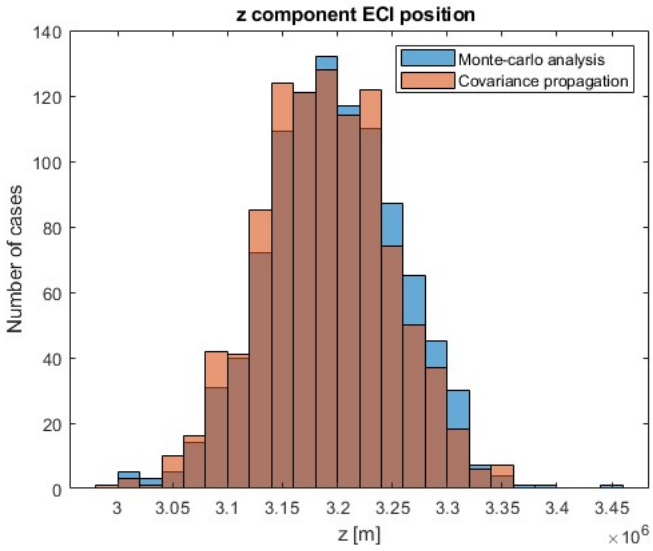

Figure 5.7: Variance on the z component of the state vector

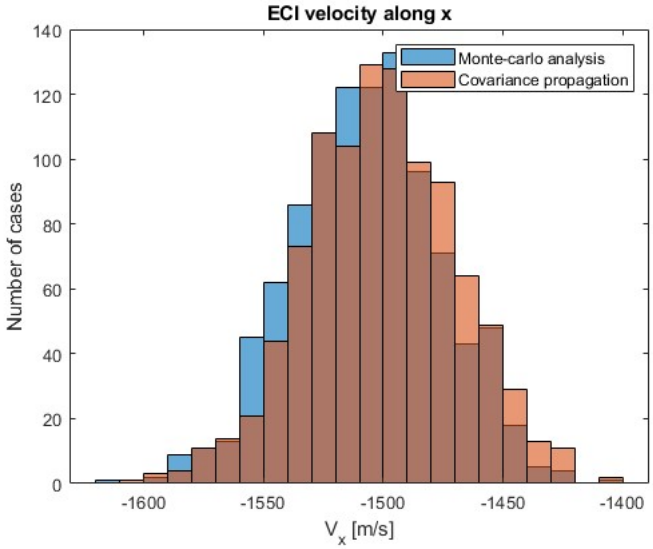

Figure 5.8: Variance on the v_x component of the state vector

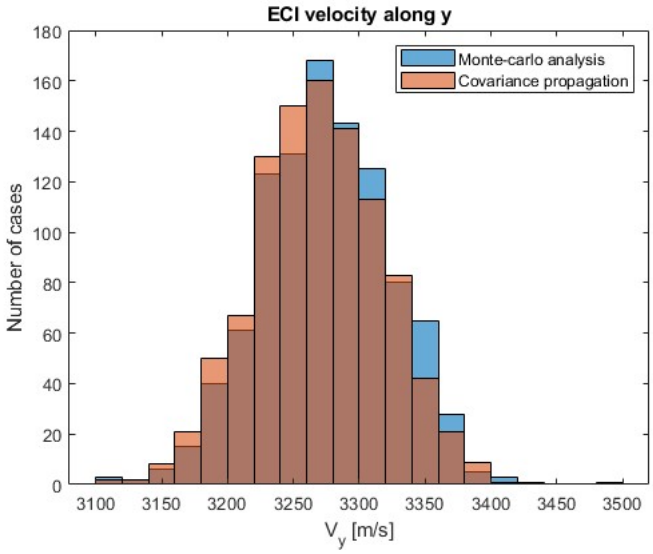

Figure 5.9: Variance on the v_y component of the state vector

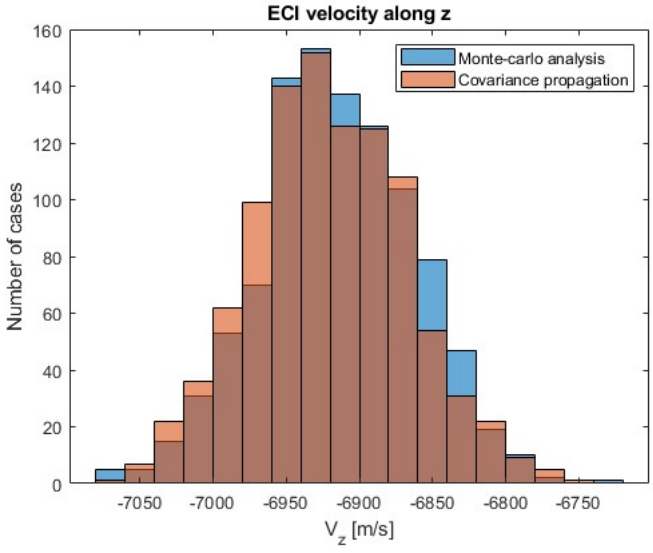

Figure 5.10: Variance on the v_z component of the state vector

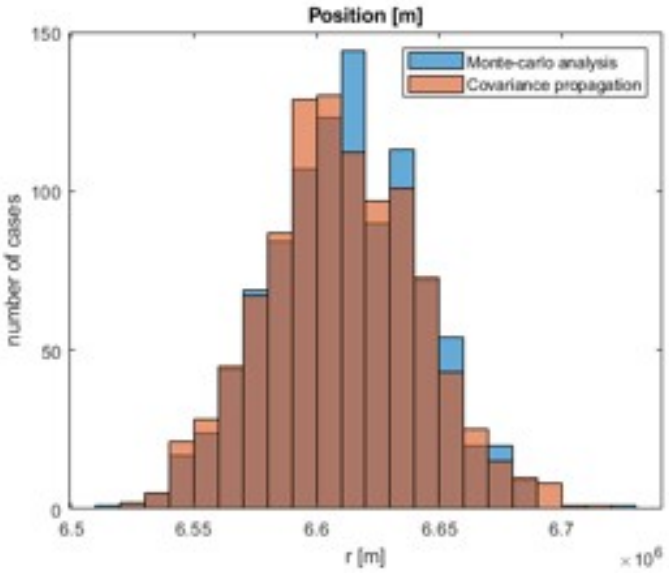

Figure 5.11: Variance on the position vector

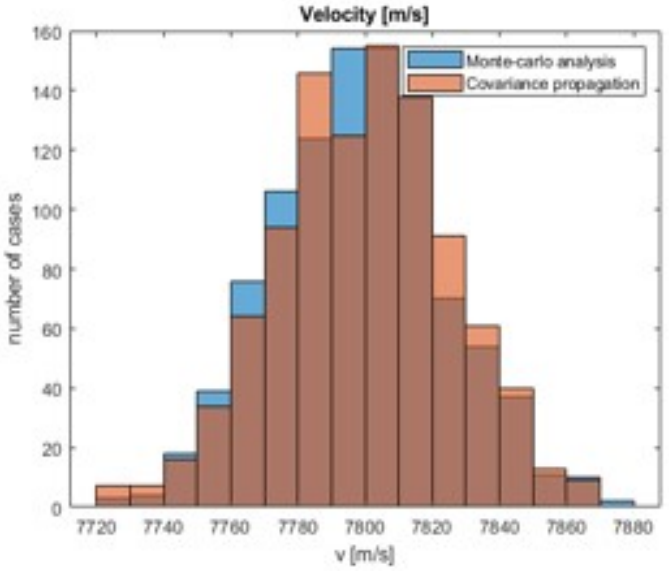

Figure 5.12: Variance on the velocity vector

First of all, it can be noticed the high consistency of the data deriving from the two different analysis. The first six graphs represent the six components of the state vector and the other two the modules of the velocity and the position of the object. These histograms justify the use of the covariance propagation method

instead of the Monte Carlo running, allowing a time saving about eight minutes on the reentry state vector. As it can be noticed, on the components of the state vector, the simulations present the same variances, demostrating that the two method can be switched. This approach can also be used in orbit determination problem. According to the hypotesis listed before this particular approximation of the problem is suitable especially for this kind of problem, considering that the problem in this case can be always considered linear. Here, the histograms for the orbital situation are shown.

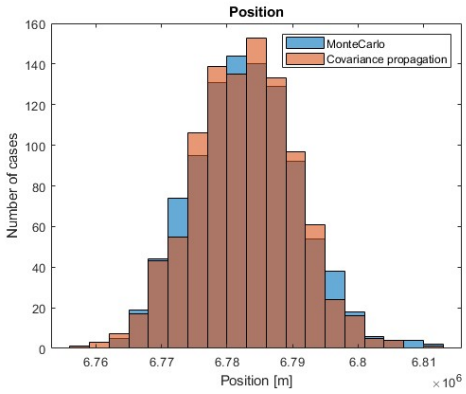

Figure 5.13: Variance on position, circular orbit

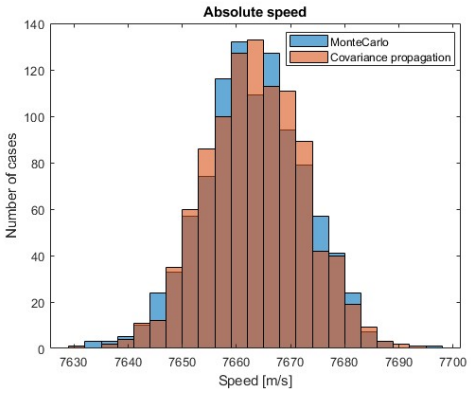

Figure 5.14: Variance on speed, circular orbit

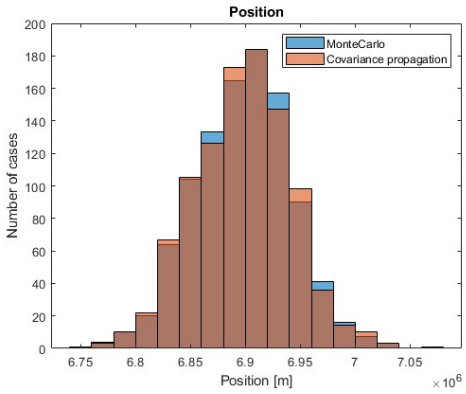

Figure 5.15: Variance on position, elliptical orbit

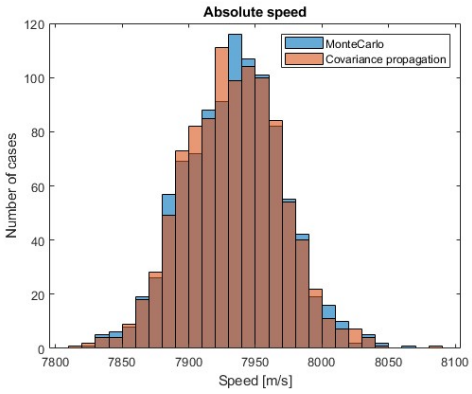

Figure 5.16: Variance on speed, elliptical orbit

From the orbital graphs, it can be noticed the high confidence of the results that, also in this case, allows the use of the covariance propagation instead of the Monte Carlo simulation. Theis particular outcomes allows the use of this method also in others tools in order to replace the Monte Carlo simulations.

5.4 Monte Carlo simulation

The Monte Carlo analysis is runned, as said before, before the fragmentation altitude, for a short time, and on the scattering of the fragments: now this second case will be analyzed. Firstly it must be underlined the uncertainties considered:

- uncertainties on the atmosphere: in particular there are some on the density and on the temperature, given by a table that the tool takes as input.
- the scattering on the bank angle, it is mainly due to the fact that the reentry is un-controlled, so the direction of the lift can not be defined in advance.
- the L/D value, on the input file the maximum value of lift to drag ratio is defined but, actually, it can change between 0 and that value;
- the last uncertainty is on the direction of the ejection velocity, if an explosion occurs. This event can be choosen by the user and the velocity value can be set in the fragment file.

All the uncertainties are randomized, thus, every Monte Carlo has one different value for each of the scattered quantities. The number of the simulation can be choosen by the user in the input file, commmonly it is set to 1000, in order to find a good compromise between the accuracy, that increases with the number of simulations, and the computational cost that in this kind of simulation is very high and, in general, increases with the number of simulations.
There are some strong hypotesis on the Monte Carlo simulation due to the altitude control. For both the modules is necessary to control the altitude of some state vector in order to avoid the unexpected stop of the running. in particular there are the following controls, they are separated for the two different modules, respectively the orbital reentry and launch risk:

1. After the propagation of the covariance matrix, there is a perigee altitude control, in order to decide if the body is able to reach the ground, if the perigee altitude is major of 100 km or below the fragmentation altitude, thus that state vector is ignored;
2. After the Monte Carlo analysis before the fragmentation altitude there is another control on the altitude of the state vector generated by the simulation, if that vector is major of 5 km than the fragmentation altitude, it will not be considered;

Here the hypotesis on the second module are listed:

1. control on the apogee and perigee altitude of each initial state of the trajectory file, in this way the tool is able to define the end of the simulation and the

minimum altitude that has to be used, chosen between 0 km and the one set by the user in input simulation file.

2. Before the fragmentation altitude there is the same control, performed in the other module, on the perigee altitude of the current state vector.

5.5 Computational times

In this section the computational time of the various analysis is shown. In particular it can be noticed the enormous advantage sing the covariance propagation, or more in general the uncertainties propagation, with respect to the Monte Carlo analysis.

Number of simulations	MC time [s]	Uncertainties propagation time [s]
100	776.21	32.378
1000	7272.39	32.315

Table 5.3: Computational time for a circular orbital propagation

Number of simulations	MC time [s]	Uncertainties propagation time [s]
100	77.621	3.2378
1000	774.239	3.2475

Table 5.4: Computational time for an elliptical orbital propagation

The reader has to be warned that the differences in the two orbital propagations, indicated in tables 5.3 and 5.4, are due to the different ending time, the circular one has runned for 10000 seconds while the second one only for 1000, there is not any technical or teorical hypotesis behind this change, it is up to the user to decide that time.

5.6 Nominal simulations

In this section the outcomes of the nominal simulation, for both modules, will be analyzed.

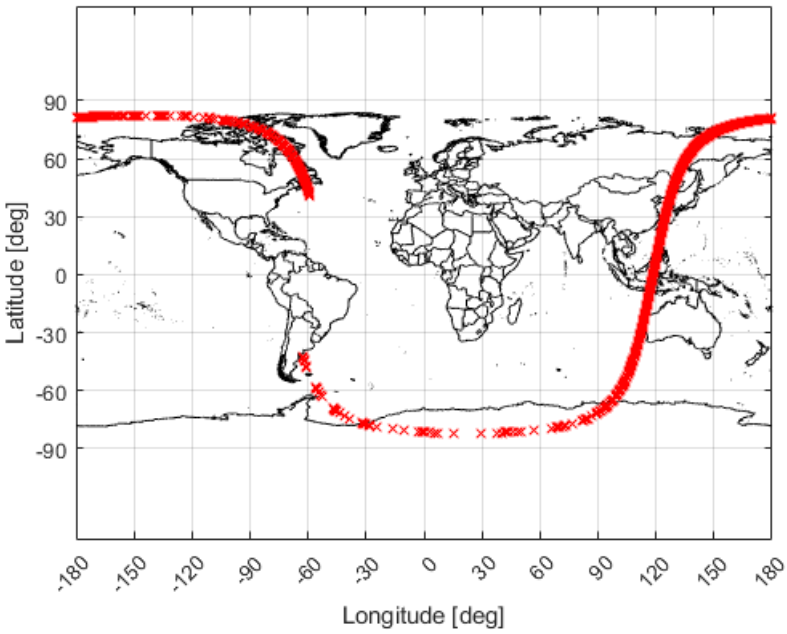

Figure 5.17: Nominal impact points from the launching module

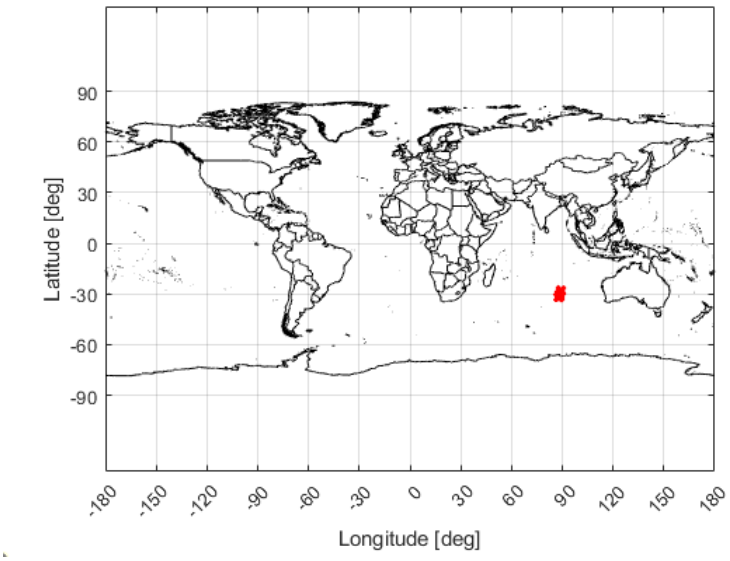

Figure 5.18: Impact points from nominal simulation of orbital reentry

In the figure 5.17 the impact points on ground are shown. Analyzing the

footprint, the following observation can be taken out:

- the computation starts from about 40 degrees of latitude and -50 degrees of longitude, it is continuous until the coasting phase, then continuous again and at the end it has some discontinuities.
- The last part of the groundtrack can not be taken into account because those impact points are not accurate.

The computation starts when one fragment reaches habitated land, as illustrated in the previous chapter.

The figure 5.18 shows the impact points from the nominal simulation of an orbital reentry, considering 12 fragments. As it can be noticed that all the points are grouped in a specific zone and they are less then the ones of the launching module.

5.7 Simulation computational times

In this section the simulation times will be shown, it has to be noticed that these times are different from the one in the section 5.5 that refer to the orbital propagation. Now the times bounded to nominal and scattered simulations will be shown, considering the following hypotesis:

- the minimum step is set to 0.5 s and it is advised to not change this value in order to have a good accuracy on the outcomes. If more accuracy is required the minimum step can be modified;
- considering the launch module the simulation's stop condition on the perigee altitude is -20 km, it is not still in orbit but it has been noticed that there are fragments that are not able to reenter in one orbit.

Module	Simulation's number	ORCA times
Launch	Nominal	5 min
	10 x 12 frag x 915 state vectors	40 min
	100 x 12 frag x 915 state vectors	2.4 h
Orbital reentry	Nominal	1 s
	10	6 s
	100	42 s

Table 5.5: Computational times ORCA

As it can be noticed the computational times are low for the orbital reentry and quite high for the launch module. The launch module can be optimize changing

the trajectory input file, it has to be sampled to one second not to 0.1 seconds as the one used for these simulations.

5.8 Risk computation

The computation of the on ground risk is the same for both modules. After the identification of the impact point there is a module, called PopRisk, whose outputs are the population density in the impacting cell, the probability of making almost one victim and the expected number of victim due to the considered account. The computation of the risk is based mainly on two formulas:

$$E_v = \rho * CA * pf * Q \tag{5.11}$$

$$P_v = 1 - e^{-E_v} \tag{5.12}$$

The 5.11 is reffered to the expected number of victims, in this equation appear:

- the population density ρ in the impact cell;
- the casualty area, indicated as CA, of the individual fragment;
- the protection factor, pf, identifies the probability that a person in the cell dies hit by a debris, It depends from the protection level of the person taking into account some conditions, for example if it is winter or summer or if it is day or night. This parameter is set by the user;
- the reliability factor, Q. This quantity is considered in both the modules, in the case of the launching one it has a value for every instant of the trajectory, in the other one there is one single failure probability set by the user and a general probability, considered as having an uniform distribution(1/100).

The 5.12 equation derives from the poisson distribution. In general the Poisson's distribution is defined as:

$$Pr(N) = \frac{\Lambda^N}{N!} e^{-\Lambda} \tag{5.13}$$

The quantities that appear in 5.13 are N, the number of victim and Λ that identifies the expected number of victim, also called, in a more general way, the form of the probability distribution. In order to obtain the 5.12 some considerations have to be made: the N is set equal to zero, in this way the probability that nobody dies is set and, then, in order to have the probability of almost one victim, the probability that everybody dies, 1, has to be subtracted to the previous quantity. The population density derives from a text file, that identifies a number of people in every cell. The site of the SEDAC supplies a lot of different documents, the main

difference is in the dimension of the cell, ORCA uses a one degree grid, this means that the cell has a one degree step for both longitude and latitude. However the user can decide which precision has to been used. In the figure an example of one degree grid of population is shown. The casualty area is the area that is involved in the impact, it is made up of the section of the debris and the diameter of a person. It is defined as the effective area bounded to the vertical falling of a debris and its risk of hitting a person. In particular the computation of the casualty area is the following:

$$A_c = \pi(d_p/2 + R_i)^2 \tag{5.14}$$

In this equation there are the diameter of a person, usually set to 0.667 m, and the radius of the sectional area of the debris. A clear representation of the casualty is here reported:

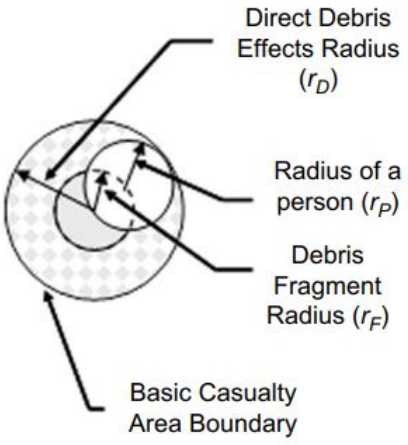

Figure 5.19: Casualty area representation [1]

From the figure 5.20 there are shown the trend of the risk, linked to each fragment, in function of the latitude band. In order to obtain these graphs a mean of the risk on the latitude band has been done. It was considered a latitude band equals to 10 degrees and the risks on that band were averaged. The outcomes shows that the major risk corresponds to 40 degrees of latitude, this was forseeable because that band of latitude is the most dense populated. The graphs are shown only for the first 6 fragments but they are similar also to the others 6. The peak between 0 and 10 degrees is due to the fact that, for launching mode, there is a passage above the asian regions, such as Japan and South Corea, these are very densly populated thus the risk results very high respect to the others. The plot reffered as figure 5.21 shows the collective risk trend per longitude band, it gives a general indication on where the fragment falls down, considering the most densly

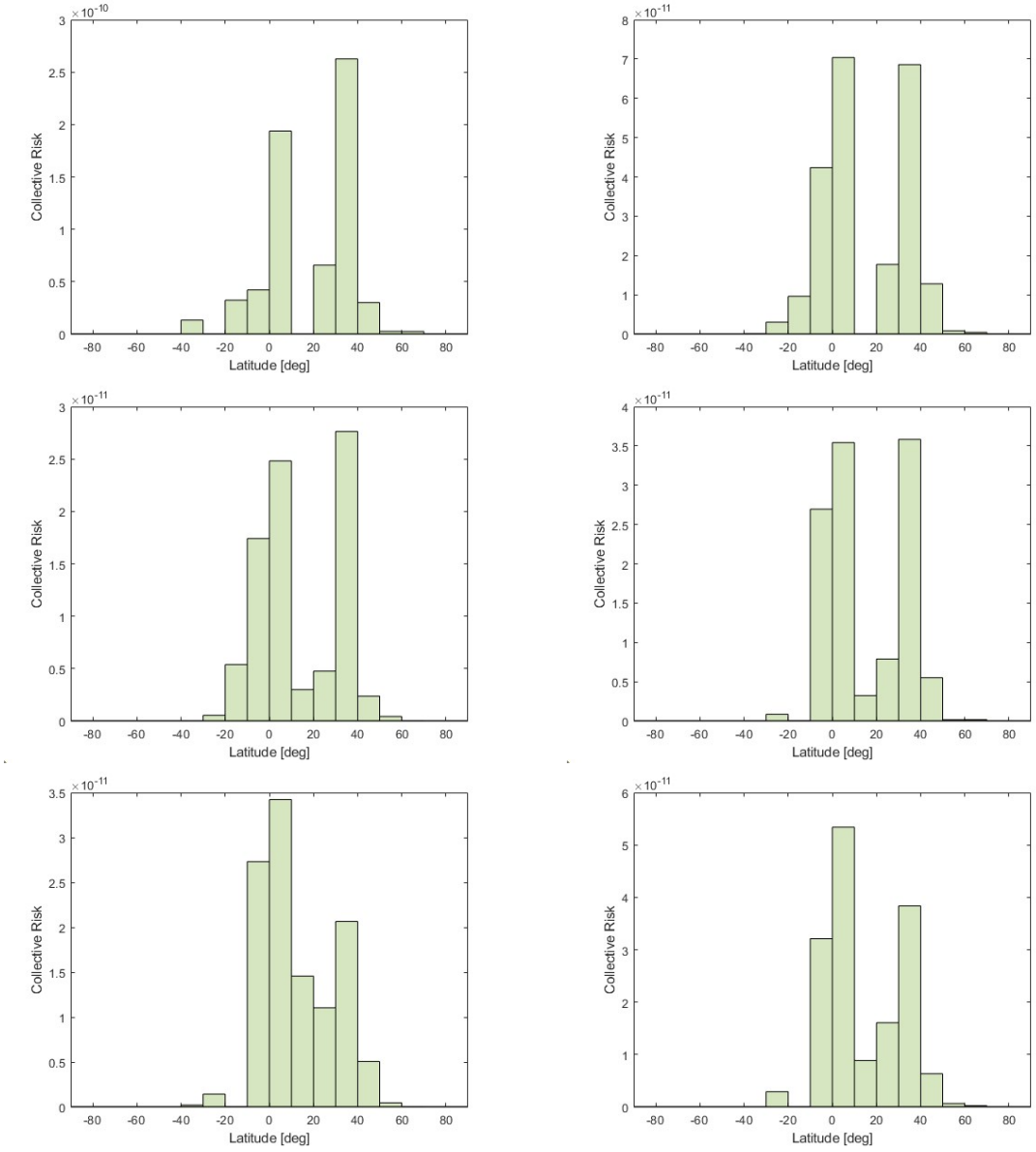

Figure 5.20: Risk for latitude from fragment 1 to 6

populated area. as it can be noticed most of the collective risk is between 120 degrees and 150 degrees, this because the ground track of the trajectory taken into account starts from -50 degrees, over the North America lands, and it finishes over Antartica, in this way it can easily be explained the plot. In this case alle the fragments are not reported becuase the plot are basically the same for every piece.

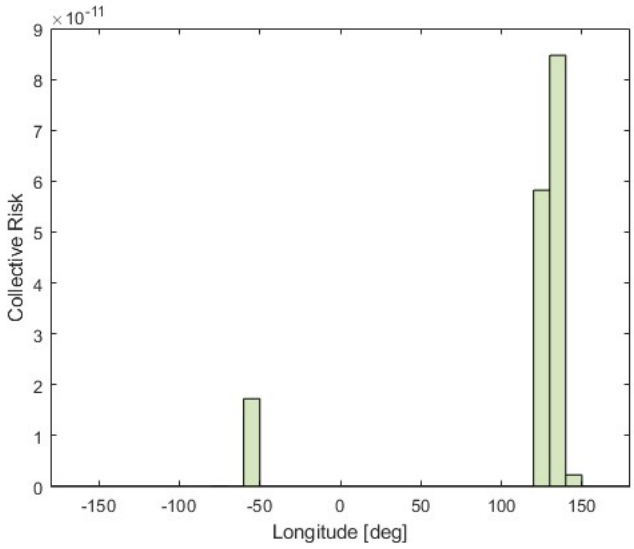

Figure 5.21: Risk per longitude band linked to the first fragment

5.8.1 Reliability factor

The equations indicated in the previous section depend on the Q factor, namely the probability of having a failure during a nominal working of the launcher. This quantity is given in a excel file where, for each instant of time, there is an unreliability factor, but, some considerations have to be done before its use. First and foremost it is noticable that the instants, in which the factor is defined, are different from the ones of the trajectory file, thus the first step should be the interpolation of the unreliability. It is not straightforward to obtain the factors for the trajectory instants because they are many more than the RAMS ones and the threat is that they will not converge to them after the interpolation. In order to solve the problem, the solution that has been adopted is considering a linear interpolation between the instants of the excel file and then, instead of adding up the risks accounted for each fragment, the definition of conditioned probability is applied. Whithin this approach the probability is defined as follows:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{5.15}$$

where A and B are the two events:

- A: the probability that the failure occurs, Q
- B: the probability that the failure, at the previous instant of time, will not occur, $1 - Q_{j-1}$

Considering that $P(A \cap B) = P(A) + P(B) - P(A \cup B)$, where $P(A \cup B) = 1$ due to the fact that the two events are incompatible, the numerator becomes:

$$P(A \cap B) = Q_j + 1 - Q_{j-1} - 1$$

In this way the 5.11 equation, implemented in order to obtain the total value, takes the form:

$$E_{v_j} = \frac{\Delta Q_j}{1 - Q_{j-1}} \rho C_a p f \tag{5.16}$$

where the j stands for the instant of time considered. This process is used for the launching risk module, not for the reentring one. It is due to the fact that the reentry starts from a single state vector, scattered in 1000 state vectors and after that a single occurence probability is assigned to each of them, by default equals to 10^{-3}. This value is the probability of failure due to a failure during the deorbiting manuever. The total risk is considered as the sum of the risk computed by equation 5.16 divided for the number of Monte Carlo taken into account, in case of the scattered simulation.

Chapter 6

Validation

In this section the results obtained in the various step of the development will be shown. In order to validate the core of the tool, the propagator described in the chapter 4, some simulations have been done. The module used to compare the results was the deorbiting one, but the core is the same of the launching one so the outcomes can be defined validated also for it. The state vector taken into account was the first one deriving from a trajectory file, representing the reentry trajectory of the last stage of VEGA, AVUM. It was set also the starting time in the input file, that corresponds to the one indicated in the trajectory file. The obtained results are shown in the figures below.

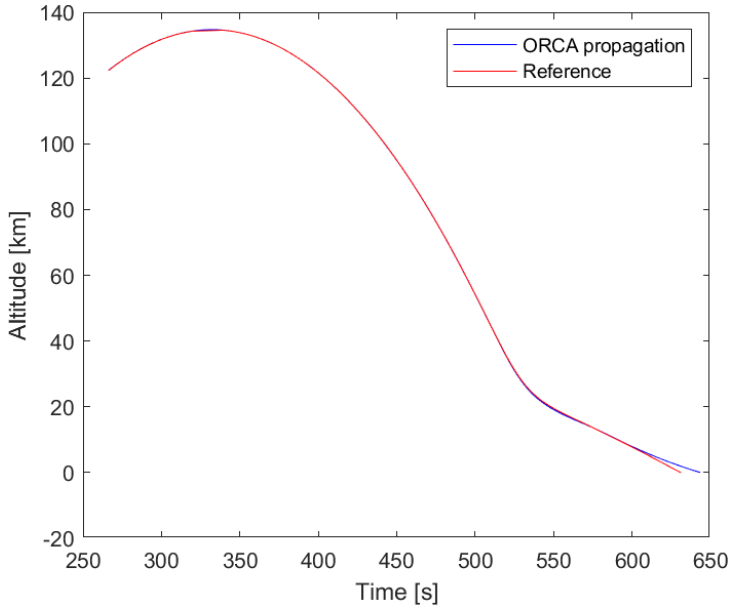

Figure 6.1: Altitude evolution until the hitting of the ground

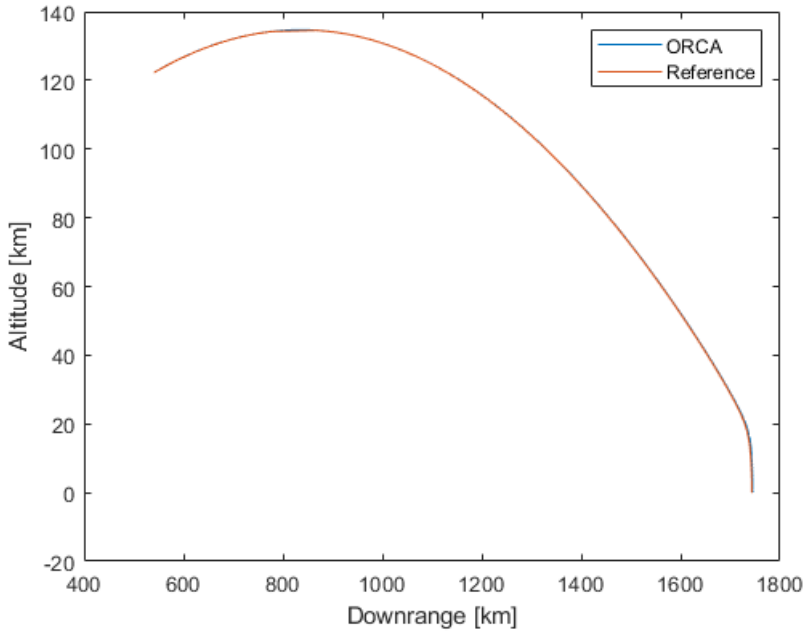

Figure 6.2: Altitude vs downrange with ISA atmosphere only

In this figure the atmosphere considered is the ISA and the lift is not considered. The two propagation are very similar to each other, there are small differences only in the final phase of the reentry, when the drag, starts to be high due to the increasing of the atmospheric density. Another quantity taken into account for this phase of the validation is the flight path angle, shown in the figure 6.3. Also in this case the atmosphere considered is the ISA one, the state vector is the same as before and the same for the other simulation quantities. From the figure the similarity with the trend of the reference file can be noticed, the only difference is in the final phase as expected from the altitude trend. It has to be underlined that these results consider only one fragment, the first one, and it not set the fragmentation altitude: the single fragment is propagated from the starting time of the simulation. After this, other quantities has been plotted. In particular from the downrange, figure 6.2, it can be noticed the high correspondence between the two tools, also considering the difference in the atmosphere models: thus the reference tool uses CIRA-86 with some adjustments in order to delete the limits presented in the pure model.

The second validation that has considered was the comparison with the NRLMSISE-00 atmosphere model. The quantities are the same as before as input. In this second round of results the two trajectories begin to diverge in the figure 6.4, specially when the atmosphere is more dense and the drag effect becomes evident. It has to be noticed that the first fragment is an aluminium sheet so the effect is

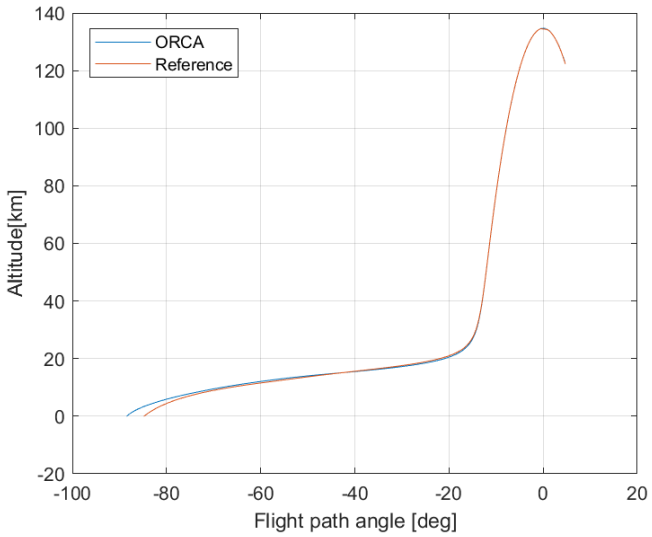

Figure 6.3: Flight path angle with ISA model

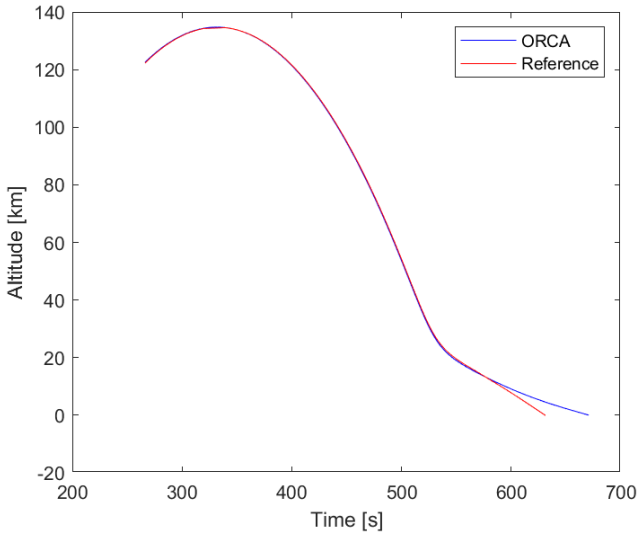

Figure 6.4: Altitude with NRLMSISE-00 model

more evident then on the other fragments. The use of the NRLMSISE-00 model allows a better approximation in the higher part of the trajectory and a general improved accuracy due to the model. Considering the FPA in this second case,figure 6.6 there is the same attitude of the altitude graph. The flight path angle describes the attitude of the fragment, in particular it is the angle between the velocity vector and the local horizon, so it can be seen that is at the beginning positive for a while and after that it starts to be always more negative, reaching the ground with the

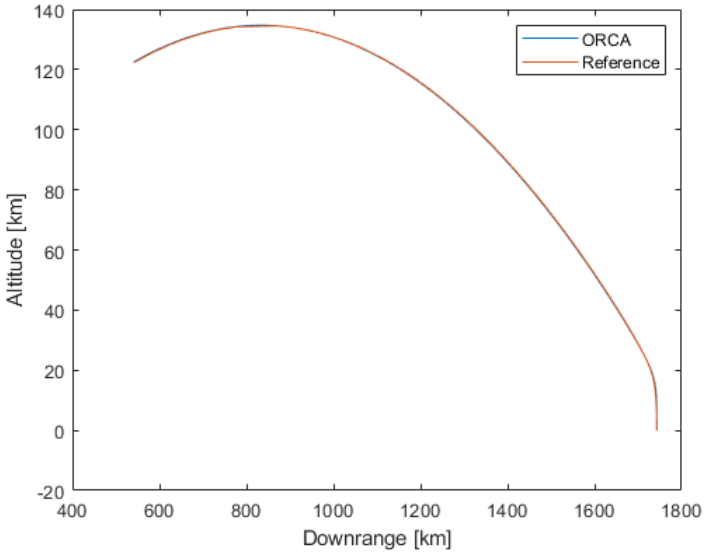

Figure 6.5: Altitude vs downrange with NRLMSISE-00 model

value -90 degrees, this means that it reaches the ground in a vertical descent, as expected.

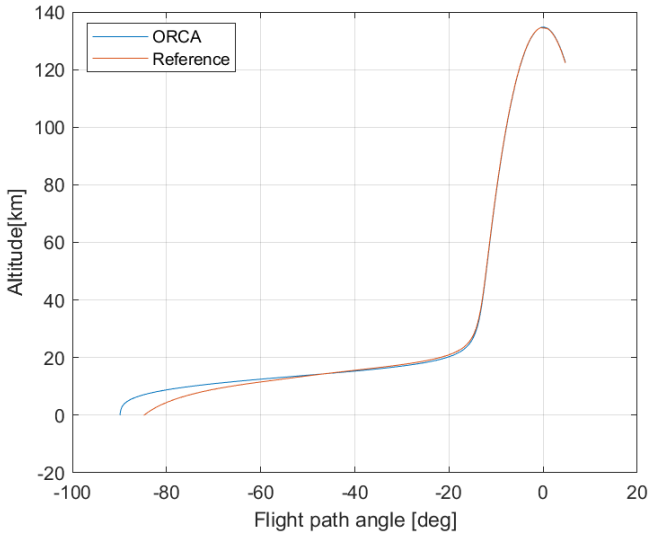

Figure 6.6: FPA trend with the NRLMSISE-00 model

In the graph 6.8 the Mach trend during reentry has shown, it is the outcome of the simulation with the ISA model and it can be noticed that from the discontinuity to 120 km. At that altitude there is a discontinuty in the density value in the

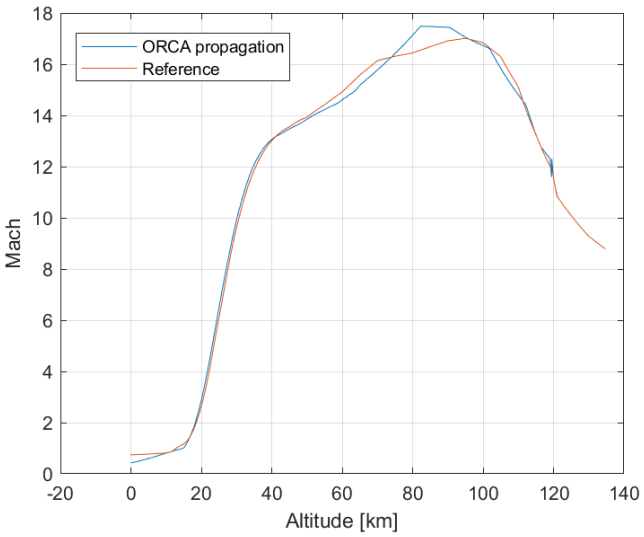

Figure 6.7: Mach trend during reentry ISA model

atmosphere model and this was one of the causes of the model changing, from the ISA one to the NRLMSISE-00. In the figure 6.7 the comparison between the reference trend of Mach and the ORCA output is shown. The first noticable feature is the matching between the two trends: they overlap in the final part of the trajectory, from 40 km to 0 km, and in the first part from 140 km to 120 km. In the middle of the trajectory, the Mach numbers have the same magnitude, reaching peak of 17. The great deceleration between 40 and 20 km corresponds to the dynamic pressure peak shown in figure 6.9, it is due to the increase of the drag, as consequence of the more dense atmosphere layer. From the figure 6.10, it can be noticed the high comparability between the two trends, till 20 km, after that the two curves separate, this can be due to the different models used in this phase, infact the reference tool uses the CIRA-86 that exists upon 20 km, under this altitude it is necessary to change the model or to approximate it. A different situation is the one of the ORCA's output, it considers indeed the ISA model that is very accurate at low altitudes.
There are others two figures which show relevants outcomes. The first one is the figure 6.12, it shows the altitude as function of the downrange considering the nominal simulation and also taking into account the whole pack of fragments. One of the first things that comes out is the different L/D value of each fragments that changes the descending mode, and as consequences, the impact points. The other graphs, the ones from figure 6.13 to figure 6.16, show the altitude trend as downrange's function of 4 differents fragments from the break-up altitude to the ground, in the case of scattered simulation, considering 100 different conditions. The main effect on the fragments is the scattering on the atmosphere's quantities,

in particular on the density. there are small effects bounded to the bank angle but in general the main difference is caused by the density. The other observation could be the one linked to the first and last fragments that has a similar balistic coefficient and their behaviors are also similar, instead of the other ones that have a very high balistic coefficients. The graph shown in figure 6.11 is the whole trajectory considering 100 scattered simulations and all the fragments, as it can be noticed there are some of them that will not reentry but they rise again, these ones are the one that are deleted by the simulation because they would produce wrong impact points.
After the validation of the propagation module with the ISA model , some other considerations had been done. First of all the combination of ISA model and CIRA-86 one has been considered: in figure 6.17 there is shown the altitude trend of the two tools. The main difference is the introduction of the lift in ORCA that caused that discrepancy in the last part of the trajectory, indeed the reference tool has not the lift model. The difference is mainly in the final part of the trajectory when the density effect is more persistent. The effect is more evident in the other graph, in figure 6.18, where the lift effect on downrange is evident in the impact point location.

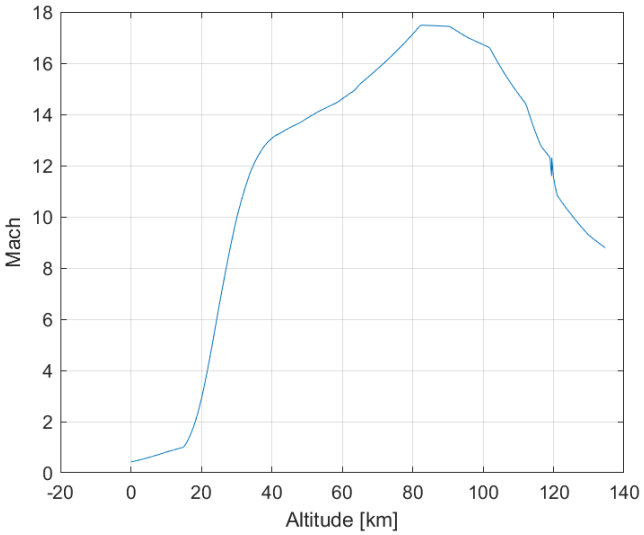

Figure 6.8: ORCA propagation of the Mach

6.1 Impact points validation

After the validation of the propagation module, the location of the impact points has been checked. In order to obtain these results the reference tool described in the

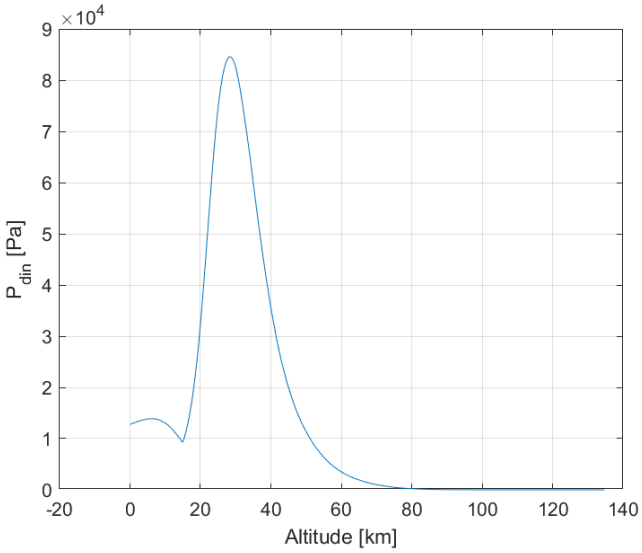

Figure 6.9: Dynamic pressure during reentry

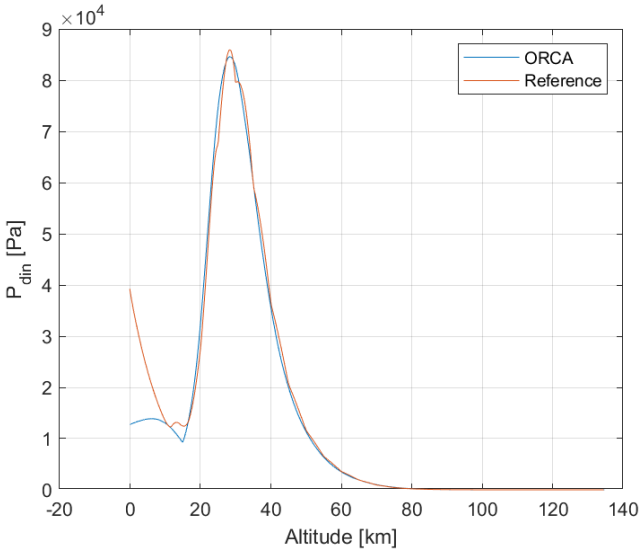

Figure 6.10: Dynamic pressure comparison

chapter 1 has been used together with the ODE function of MATLAB. Considering the reference tool, the outcomes showed are generated from the uncontrolled reentry module, considering the same initial state vector. These results consider the nominal conditions because is not possible to compare two different Monte Carlo analysis from different tool owing to the fact that there is a randomic generation of number in them.

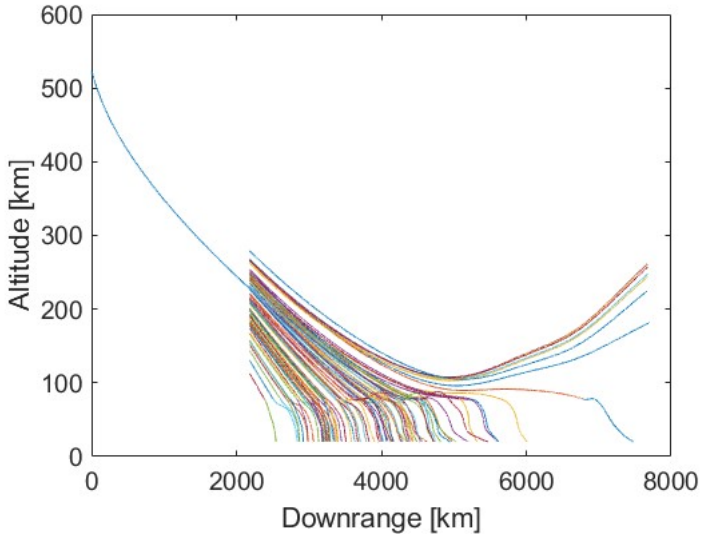

Figure 6.11: Altitude vs Downrange

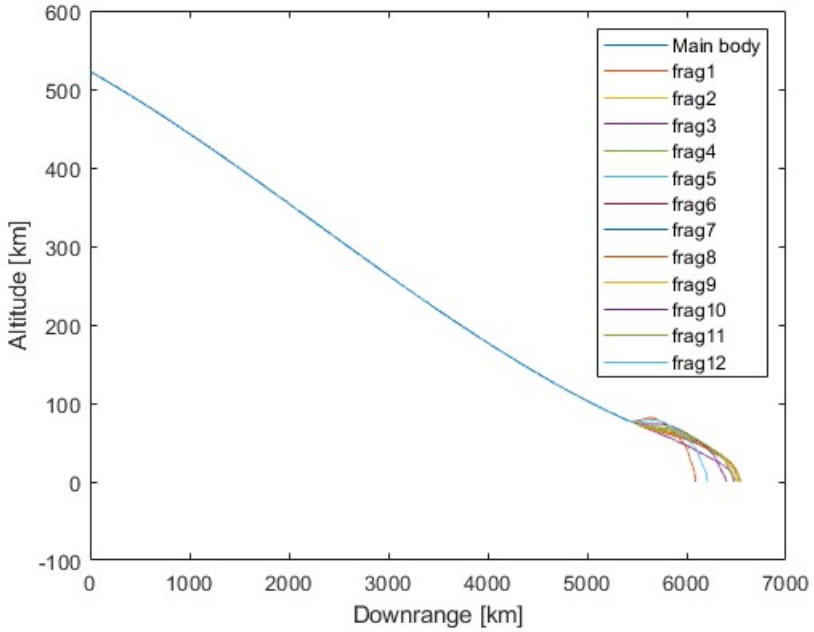

Figure 6.12: Nominal trajectory

The results shown in the table 6.1 deriving from the nominal simulation of ORCA, reentry module, starting from the same state vector, but with different aerodynamics data: the first results are without lift and bank angle; the second

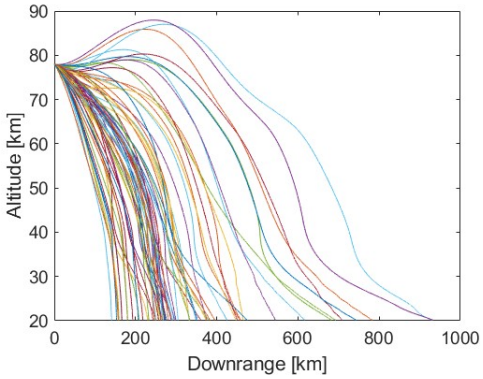

Figure 6.13: Altitude vs Downrange, fragment 1

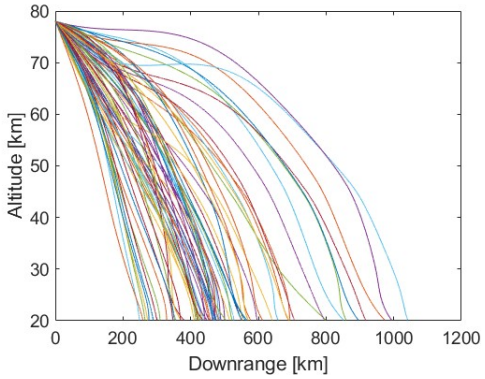

Figure 6.14: Altitude vs Downrange, fragment 2

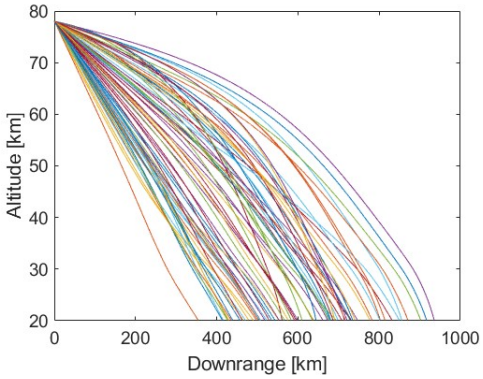

Figure 6.15: Altitude vs Downrange, fragment 3

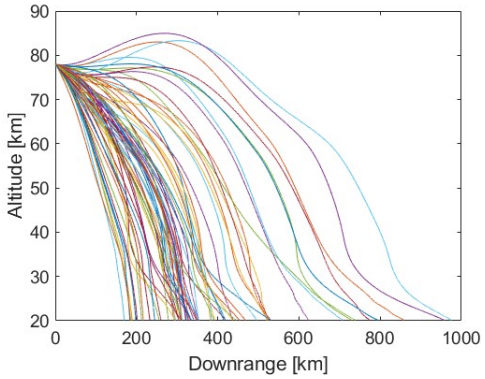

Figure 6.16: Altitude vs Downrange, fragment 12

	Reference tool	ORCA	ODE45
Longitude [deg]	-121.7235	-121.5780	-121.8724
Latitude [deg]	4.25	4.26	4.24
Longitude [deg]	-118.2815	-118.0081	-118.3507
Latitude [deg]	4.53	4.54	4.52

Table 6.1: Comparison between the impact points deriving from both the tools

ones considers L/D ratio equals to 0.15 and the bank angle sets to zero. Another comparison has been done in order to consider the simulation with ODE 45. As it can be noticed the ODE45 results are slighty different from the other, but it can be motivated by the major accuracy of the integration method, it is a Runge-Kutta

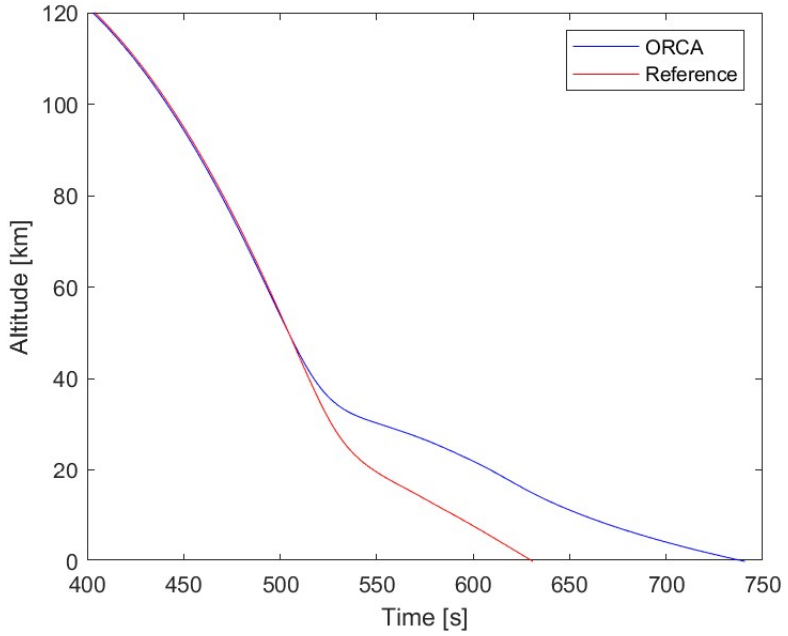

Figure 6.17: Altitude trend considering a combination of ISA and CIRA-86

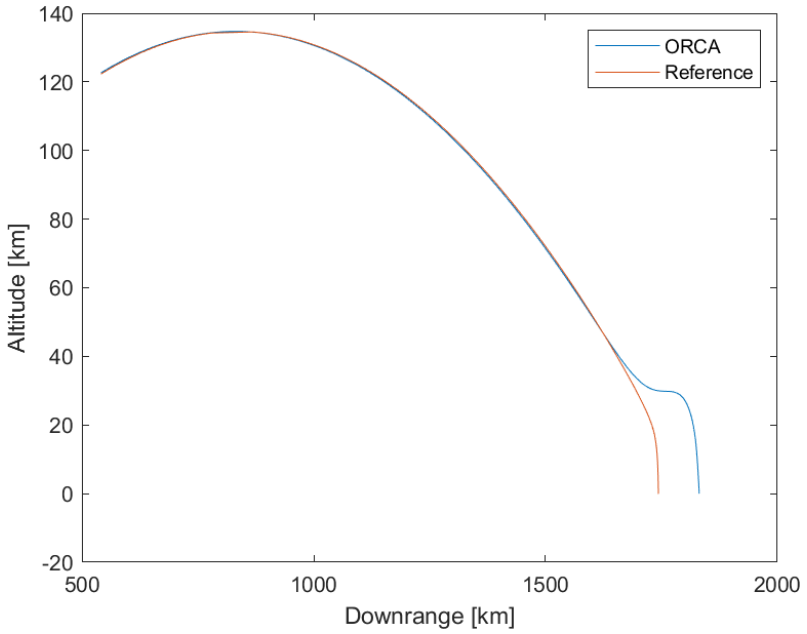

Figure 6.18: Altitude vs downrange with combination of ISA and CIRA-86

45 integrator, this means that it computes the derivatives for five times and then it controls the numerical error on the fifth. Another kind of table is the one shown in

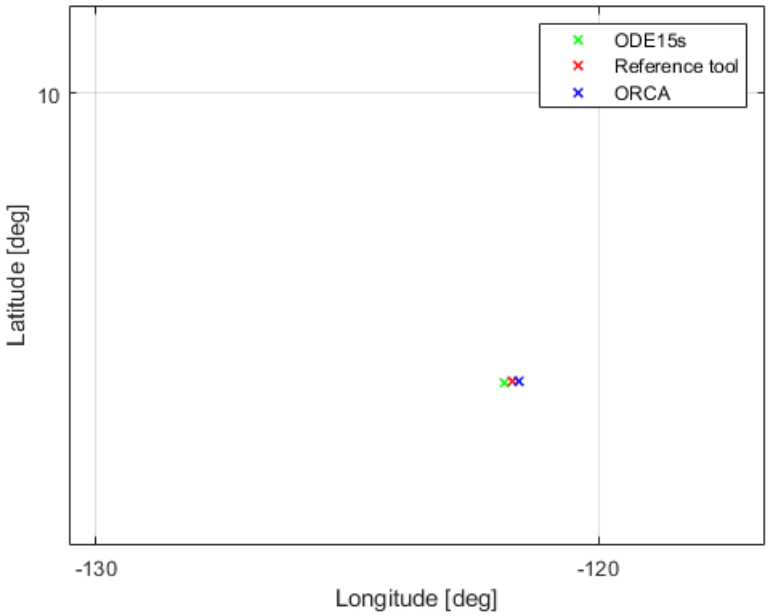

Figure 6.19: Impact points on map

6.2, it is a comparison between the final version of ORCA and the reference tool. As it can be noticed there is also the information about the distance of the impact points in terms of kilometers: thus the impact points can be considered validated from the moment this distance is below the 2 km. The main difference is due to the fact that the reference tool uses a different integration method and another kind of atmosphere's model.

Point's number		Reference tool	ORCA	Δd [km]
1	Longitude [deg]	-103.81	-103.82	1.5
	Latitude [deg]	31.25	31.26	
2	Longitude [deg]	93.55	93.52	3
	Latitude [deg]	23.41	23.41	
3	Longitude [deg]	-109.69	-109.70	1.5
	Latitude [deg]	30.19	30.18	

Table 6.2: Comparison between the impact points

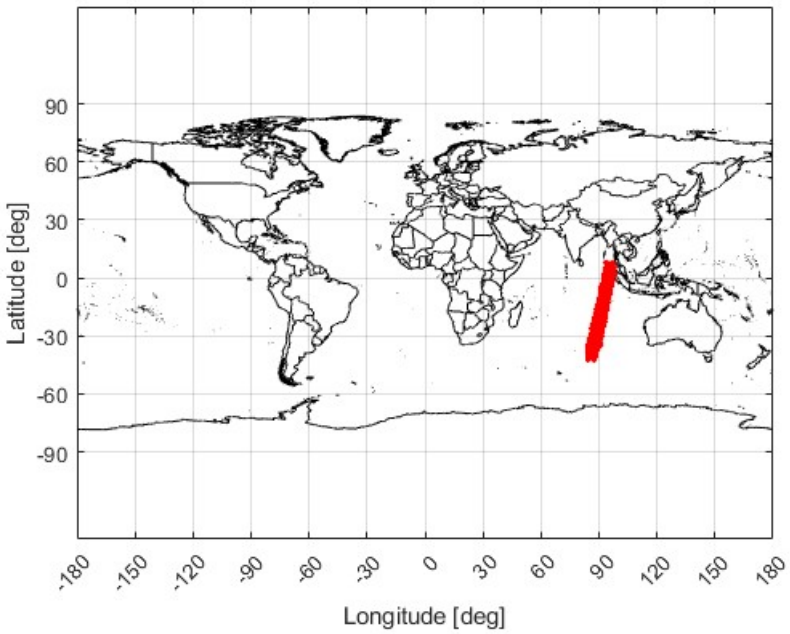

Figure 6.20: Impact points according to ORCA, from reentry module

6.2 Scattered simulations

From the figure 6.20, the impact points of the orbital reentry module can be seen. The outcomes derives from 100 MonteCarlo simulations, it can be noticed that there is a single point detached from the main footprint, it is related to the first fragment and its ballistic coefficient that is very low, with consequences on its reentry conditions. In order to not obtain these results the minimum step can be managed and some consideration on the state vector corresponding to fragment not able to reenter the atmosphere should be done.
Also similar considerations can be done in the launching module, taking into account the major complexity bounded to the problem. The fragments not able to come back on Earth are managed by the control on the perigee altitude, in some future changes this kind of control can be managed by the number of orbits before the reentry. The impact points out of the main groundtrack can be managed by the minimum step, less is this number better will be the simulation, certainly the disadvantage is that the computational cost will increase if the minimum step is reduced, so the user has to find a middle ground between the accuracy of the simulation and its computational cost.

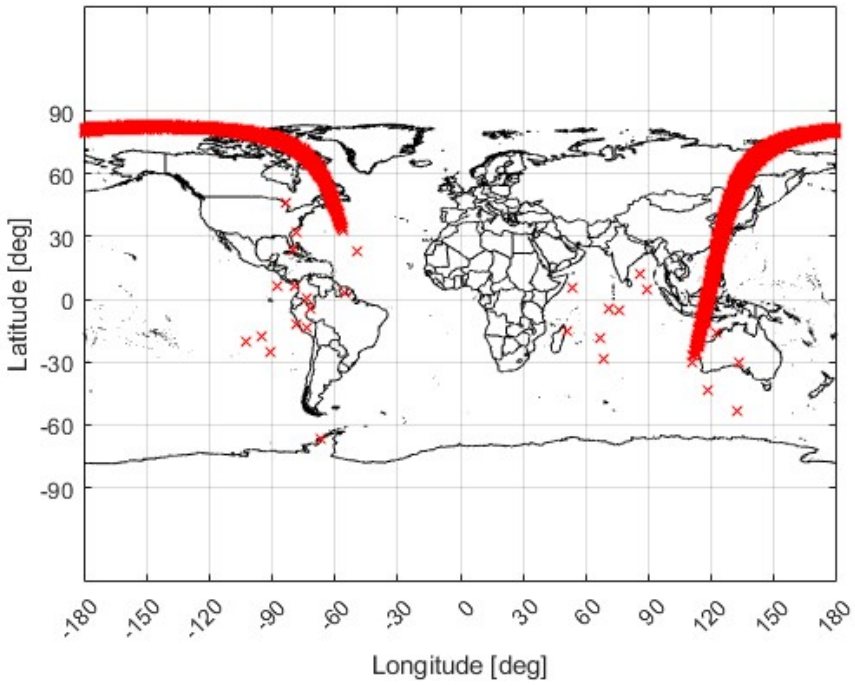

Figure 6.21: Impact points from 100 Monte Carlo simulation on launching module

6.3 Risk computation validation

In this section the risk validation will be shown. In order to compute the risk, first of all, the population grid has been generated and it has been compared with the one from the reference tool. In the table below the number of people in different region of the Earth are indicated. In particular ORCA was run with different initial state vectors in order to change the impact points from the ocean location to a populated region and compute a risk different from zero. Analyzing the numbers in the table 6.4, few things can be noticed:

- Considering the same impact points regions, there are little differences in the number of people, this is due to the fact that the reference tool rearranges the population file of the SEDAC and so the numbers change.
- From the previous point, the risk computation is different between the two tools but, if the ratio between the population count and the risk numbers is computed, they are the same thus it can be concluded that the two tools compute the risk in the same way but with different data.

From figure 6.22, it can be noticed the convergence of the risk number to a

stabilization value with the increase of the Monte Carlo's simulations. It is noticable moreover the fact that in order to have an accurate risk value, it is necessary to have an high number of Monte Carlo's simulations, between 500 and 1000.

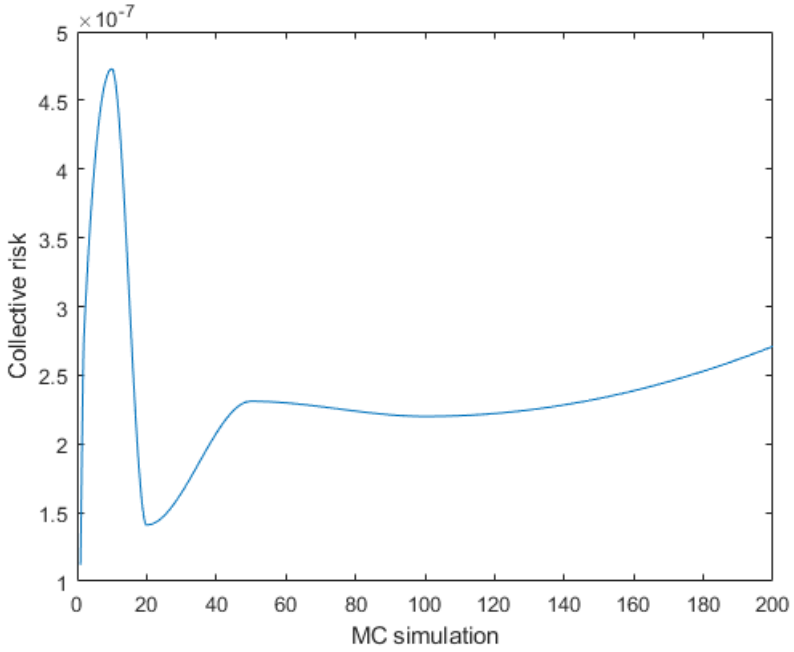

Figure 6.22: Risk convergence

ORCA	Reference tool
1.7E-6	1.4E-6
9.3E-8	5.2E-8
7.6E-6	1.2E-6

Table 6.3: Collective risk comparison

6.3.1 Grid population definition

The grid population that ORCA uses is a customized code that can be used also in a post-processing phase, if the population density or count in a specific region has to be known. The file taken into account are the ones described in the appendix C but they can be changed by the user, considering that they are added in the input paths file. After that the grid is defined through a mesh on MATLAB an the outcomes are displayed in a graph similar to the one shown in the appendix

C. In particular, if a single cell is taken into account, the number of people in that cell corresponds to the number indicated in the point on the lower left. The validation of the grid has been done through the referece tool, considering the same population file. One of the biggest difference between the files is due to the rearrangement described in the previous section, thus it was decided to not change the input file of the population taking into account the one deriving directly from the SEDAC website, without any modification. From the table 6.4 these differences on the number of people in the impacting cell has been underlined: the difference on the number of people in that cell justify the difference on the risk value, without influencing the order of magnitude.

	Colombia	Iran	USA	UAE
Reference tool population count	2405493	212429.6	96078.89	96803.41
ORCA population count	21717550	252862	86819	43924
Reference collective risk	9.92E-04	1.01E-04	4.79E-04	4.53E-04
ORCA collective risk	8.95E-0.4	1.21E-04	4.45E-05	2.04E-05

Table 6.4: Comparison for risk computation

Chapter 7

Analytic impact

In this chapter a simpler way of trajectory propagation will be presented. In this case the hypotesis are stronger and there is not an integration module but the following equation as to be solved:

$$R_{eq} = \frac{a(1-e^2)}{1+e\cos\theta} \tag{7.1}$$

The hypotesis taken into account are:

- the problem is considered as Keplerian, this means that there is no influence from J_2, even more so the other harmonics, and the drag is set to zero;
- the Earth is considered as a sphere, so the impact point has not to be reported in geodetic reference frame.

For this kind of computation are taken into consideration only the components of the initial state vector. First of all the module computes the orbital parameters:

- the major semiaxis is estimated from the mechanical energy conservation :

$$-\frac{\mu}{2a} = \frac{V_0^2}{2} - \frac{\mu}{r_0} \tag{7.2}$$

  where V_0 and r_0 are the velocity and position norms of the initial state vector.

- the orbit's eccentricity taken out from the angular momentum:

$$\vec{h} = \vec{r_0} \wedge \vec{V_0} \tag{7.3}$$

  from 7.3 it is simple to derive the eccentricity as:

$$e = \sqrt{1 - \frac{|h|^2}{a * \mu}} \tag{7.4}$$

- the orbit's inclination is defined as:

$$\vec{i} = \vec{k} \cdot \vec{w} \tag{7.5}$$

  where k and w are defined as shown in figure. From this definition is simple to point out the inclination as the ratio between the third component of the angular momentum and its norm.

- the longitude of the ascending node is defined in the following manner:

$$\Omega = \arccos \frac{N_x}{|\vec{N}|} \tag{7.6}$$

  where the $\vec{N}$ is the direciton of the line of the nodes, namely the intersection between the orbital plane and the reference plane.

- the perigee's arguments is determined as:

$$\omega = \arccos \frac{\vec{N} \cdot \vec{e}}{|\vec{N}||\vec{e}|} \tag{7.7}$$

- eventually the true anomaly is set as:

$$\theta = \arccos \frac{\vec{e} \cdot \vec{r}}{|\vec{e}||\vec{r}|} \tag{7.8}$$

The module defines all the previous illustrated parameters and then computes the true anomaly of the impact point solving the initial equation 7.1. Thanks to the new true anomaly the new state vector can be formed:

$$\vec{r_1} = \frac{h^2}{\mu} \frac{1}{1 + e\cos\theta_1} \begin{Bmatrix} \cos\theta_1 \\ \sin\theta_1 \\ 0 \end{Bmatrix} \tag{7.9}$$

$$\vec{v_1} = \frac{\mu}{h} \begin{Bmatrix} -\sin\theta_1 \\ e + \cos\theta_1 \\ 0 \end{Bmatrix} \tag{7.10}$$

It should be noted that the state vector defined in this way is in the perifocal frame, so it has to be rotated in the ECI reference frame.

7.1 Results

In order to control if the results were proper, they were compared with the ones taken out from the integrator, so using the Runge-Kutta method, and from the ode function of Matlab. First of all the hypotesis of ORCA have been modified according to the ones of the analytic module, then the simulations were run.

The results obtained are the following:

- for the integrator the longitude of the impact point is -38.7263 ° and the latitude is 16.7251 °;
- for the analytic module are -38.7243 ° and 16.7267 °;

	Longitude [deg]	Latitude [deg]	Distance [m]
Integrator	-38.7263	16.7251	277.82
Analytic mode	-38.7243	16.7267	

Table 7.1: Impact points obtained in the two different way

As can be noticed the results are fairly accurate within a computation time of 3 seconds, lower than the one of the integrator, so it is justified the use of the module in the launching module to compute the first impact points. In this way, if the impact point is in the ocean, or in general not on land, the integrator will skip that point and will save up the time. The launch module uses this way to compute the first impact points and control if they are on the ocean or on habitated land. Thanks to the accuracy of the analytic mode the use of the method is justifyed and it gives a realistic impact point on Earth, in this way about 700 initial state vectors can not be propagated.

Chapter 8

Conclusion

In this chapter the outcomes achieved in this work will be summed up together with the developed knowledges. ORCA is a 3 dof tool capable of simulating the impact points after a failure, during the launch ascent or during a deorbiting manouver. The validation of the tool demonstrated that it is able to compute the risk with a great accuracy, comparing it to the other tools. It is also compliant with the regulation's requirements, especially the FAA's ones, as showed in the introduction chapter. Below the aims achived in the work will be shown together with the new knowledges developped in the company. Firstly, the prefixed targets has been accomplished:

- ORCA is capable to compute the on-ground risk due to an orbital reentry and to a launch event;
- The impact points identified by the tool had been compared to the ones coming from off-shelf tools and it comes out that they are comparable, with a difference around 1.5 km.
- The risk computation is correct, if compared with the ones coming from the reference tool, as demonstrated in previous chapters. The main difference between the results is due to the different population density.

In addition to the main objectives, ORCA has capability for the data's post-processing. It can display:

- the impact points on a map;
- the graphs that the user chooses (Mach vs time, altitude vs time...);
- the population density grid;
- the risk per latitude and longitude band, in both modules.

Thanks to the development of this tool the company is able to manage the risk computation internally, and the following additional competences have been developed in order to apply them also to other works:

- the development of a 3 dof model as requested by the FAA in the 417.207[17][18], including all the forces such as the lift, the drag and the gravity.
- the covariance matrix propagation and its potential applications to other tools and situations, especially if they are linked to the orbital propagation that, as said before, is the better way to use the Merkley's approximation.

Although the ORCA module have been optimized from all the points of view there are still some improvements that could be done:

- First of all, in order to respect the compliance with the regulations, the air and maritime risk should be added, considering air traffic and maritime one. Usually these considerations are not taken into account because the authorities stops both the traffics when there is the risk of a failure during the operations.
- from the computational point of view the codes can be improved considering the input files. It is necessary to sample the trajectory file to 1 second step and not to 0.1 as it is now, in this way the tool has not to read less initial state vector and it can go faster;
- the other improvement that could be performed is translating ORCA in another programming language. In particular it can be written in Java or FORTRAN: thus the code is near to machine language and it surely speeds up.

Appendix A

Input file

The two modules have different input files in which the user can load the quantities for the simulation.

One of the input file for the launch module is the trajectory file, that has to contains in each row the state vector from column 2 to 7, in the first one there are the instants of time, and is important to have the perigee altitude in the 34th column. It is important to comply with this column subdivision because of the way in which the code reads the files, if these instructions are on a wrong disposition, an error could be displayed. Another kind of file is the so called Input file.txt. It is a text file which contains the path where all the necessary file can be found. In this file is necessary to put the full path, including the file name. The file listed are the following, as indicated also in figure A.3:

- the fragment file;
- the population file;
- Atmosphere dispersion file, they are two, one for the density and one for the temperature;
- the file which contains the covariance matrix;
- the file containing the information on the primary body, called input before fragmentation.

The input file before fragmentation, as shown in figure A.1 is required by both the modules and it contains information about the primary body, specifically data about the L/D ratio, drag coefficient, mass and reference surface. The user has to provide another file called fragment file, it includes data about each fragment that survives. These information are arranged in an Excel file as a table. The table has to been set always in the way indicated in the ReadMe file of the code

```
#Mass [kg]
750

#Supersonic Cd
1.2

#Reference surface [m^2]
2.901

#Cl
0.2

#Bank angle [deg]
0
```

Figure A.1: File layout before the break-up altitude

in order to not have any error in the simultion. This file is the same in both the modules, as specified in the chapter 5. The last file is not properly an input file, this means that the usr has not to provide it, it is the file from which the tool reads the quantities about the simulation. These values can be changed by the user, opening this file. The major difference between the two modules is the presence of the initial condition in the orbital reentry module, it is worth of notice that this value has to be separated by a space not a tab or any other punctuation, as it can be seen in the figure A.2. In this file there are two different flag that can be set: the one for the Earth, if it is necessary to consider it as a geoid or as a sphere, and the one for the ejection velocity. This last values has to be set if the user wants to consider an explosion at the fragmentation altitude, adding in this way a ΔV to the velocity in the state vector. The value of the velocity is in the fragment file while the direction is generated randomly.

```
#Simulation Date dd/mm/yyyy h:min:sec
24 01 2021
21 55 28

#flag_Earth, 0 sphere 1 geoid
1

#Number of Montecarlo's simulations
1000

#Simulation time [s] t0 dt_max tend
0.745408E+04
30
1000000

#dt_min
0.15

#Fragmentation altitude [km]
78

#Covariance stop altitude [km]
80

#Minimum altitude [km]
20

#Initial condition x y z vx vy vz ECI_Avio
-0.240407E+07  0.151620E+07  0.626884E+07 -0.402209E+04  0.547214E+04 -0.315699E+04

#Probability sigma
3.1

#Ejection velocity, 0 no 1 yes
0
```

Figure A.2: Input file with simulation quantities

```
#Fragment file
Input\Ballistic_Coefficients_AVUM_1959.xlsx

#Population file
Input\gpw_v4_population_density_rev11_2000_1_deg.asc

#Atmosphere dispersion file
Input\Dispersion_density.txt
Input\Dispersion_Temperature.txt

#Initial covariance matrix
Input\CM0.txt

#Input before fragmentation
Input\Input_bf.txt
```

Figure A.3: Input file for the definition of the paths

Appendix B

Atmosphere file

The atmosphere files are a group of documents .mat generated from a MATLAB code that considers the NRLMSISE-00 model for every day of each month. The atmosnrlmsise00 is the MATLAB's function used for the production of the file. This function needs some quantities such as:

- the Ap factor sets to 15;
- the f107 factor and average f107 set to 140;
- the longitude, the latitude band and the altitude set, respectively, from -180° to 180° with 10° of step, from -90° to 90° and from 0 to 300 km;
- the UTC time, in this case it is 0 seconds;
- the day of the year, this is the quantity that changes from the first of January to the first of December;
- the year taken into consideration.

Running this code, the necessary file for ORCA can be obtained. There are two options that can be considered the first, is the one adopted by default in the tool, takes the .mat file, the second has a simplification on the longitude in order to write a clear text file. This one is the averaging of the quantities on the longitude band and allows to write the file that can be given to ORCA to simulate the reentry. Here there is an example of the two kind of file. They are made up of 37 rows, 19 columns and a third dimension equals to 601: the rows correspond to the longitude, the columns consider the latitude and the third dimension is the altitude.

```
val(:,:,1) =

  Columns 1 through 13

   1.3428    1.3322    1.3081    1.2773    1.2454    1.2141    1.1846    1.1608    1.1490    1.1533    1.1721    1.2000    1.2317
   1.3428    1.3294    1.3047    1.2755    1.2458    1.2155    1.1854    1.1608    1.1492    1.1546    1.1741    1.2011    1.2306
   1.3428    1.3267    1.3012    1.2734    1.2456    1.2161    1.1855    1.1601    1.1487    1.1552    1.1755    1.2019    1.2292
   1.3428    1.3241    1.2977    1.2708    1.2446    1.2159    1.1850    1.1590    1.1478    1.1551    1.1763    1.2021    1.2274
   1.3428    1.3216    1.2941    1.2678    1.2428    1.2151    1.1842    1.1579    1.1466    1.1545    1.1763    1.2018    1.2255
   1.3428    1.3195    1.2907    1.2646    1.2407    1.2140    1.1837    1.1574    1.1458    1.1539    1.1760    1.2013    1.2237
   1.3428    1.3177    1.2876    1.2614    1.2384    1.2132    1.1840    1.1578    1.1459    1.1536    1.1758    1.2010    1.2226
   1.3428    1.3163    1.2851    1.2586    1.2366    1.2131    1.1852    1.1594    1.1469    1.1540    1.1760    1.2012    1.2222
   1.3428    1.3155    1.2834    1.2567    1.2357    1.2138    1.1874    1.1619    1.1488    1.1551    1.1767    1.2018    1.2226
   1.3428    1.3152    1.2826    1.2558    1.2356    1.2153    1.1901    1.1649    1.1512    1.1566    1.1776    1.2028    1.2234
   1.3428    1.3156    1.2829    1.2561    1.2364    1.2171    1.1928    1.1677    1.1534    1.1580    1.1785    1.2036    1.2241
   1.3428    1.3166    1.2842    1.2574    1.2378    1.2188    1.1949    1.1699    1.1552    1.1590    1.1787    1.2037    1.2243
   1.3428    1.3181    1.2865    1.2596    1.2394    1.2199    1.1960    1.1712    1.1563    1.1592    1.1781    1.2027    1.2238
   1.3428    1.3203    1.2898    1.2625    1.2409    1.2202    1.1961    1.1717    1.1569    1.1589    1.1766    1.2008    1.2225
   1.3428    1.3229    1.2937    1.2658    1.2422    1.2198    1.1953    1.1717    1.1572    1.1583    1.1745    1.1981    1.2210
   1.3428    1.3259    1.2983    1.2694    1.2434    1.2188    1.1940    1.1715    1.1576    1.1579    1.1724    1.1954    1.2196
   1.3428    1.3292    1.3032    1.2733    1.2444    1.2176    1.1924    1.1711    1.1582    1.1579    1.1708    1.1932    1.2189
   1.3428    1.3326    1.3083    1.2773    1.2456    1.2163    1.1907    1.1706    1.1588    1.1582    1.1698    1.1917    1.2190
   1.3428    1.3360    1.3135    1.2814    1.2468    1.2150    1.1887    1.1695    1.1590    1.1585    1.1692    1.1908    1.2196
```

Figure B.1: First rows and columns of january density files

```
val(:,:,1) =

  Columns 1 through 13

  261.1215  263.2323  268.2905  275.3123  283.0032  290.0980  295.6957  299.4077  301.2566  301.4002  299.8143  296.1201  289.7900
  261.1215  263.5266  268.7299  275.6702  283.1175  289.9921  295.5443  299.3436  301.2214  301.2375  299.5191  295.9293  289.9533
  261.1215  263.8001  269.1365  275.9960  283.2110  289.8777  295.3877  299.2694  301.1734  301.0697  299.2345  295.7659  290.1640
  261.1215  264.0442  269.4976  276.2796  283.2808  289.7582  295.2308  299.1873  301.1139  300.9018  298.9693  295.6348  290.4160
  261.1215  264.2515  269.8019  276.5121  283.3246  289.6373  295.0781  299.0997  301.0447  300.7390  298.7312  295.5398  290.7017
  261.1215  264.4153  270.0397  276.6862  283.3412  289.5187  294.9344  299.0094  300.9681  300.5861  298.5274  295.4839  291.0126
  261.1215  264.5306  270.2034  276.7963  283.3300  289.4060  294.8040  298.9191  300.8864  300.4477  298.3639  295.4686  291.3394
  261.1215  264.5938  270.2877  276.8389  283.2915  289.3025  294.6908  298.8316  300.8020  300.3280  298.2456  295.4944  291.6722
  261.1215  264.6028  270.2900  276.8127  283.2266  289.2114  294.5982  298.7494  300.7174  300.2307  298.1760  295.5605  292.0010
  261.1215  264.5573  270.2101  276.7185  283.1376  289.1354  294.5290  298.6751  300.6354  300.1586  298.1570  295.6650  292.3157
  261.1215  264.4588  270.0506  276.5593  283.0271  289.0769  294.4852  298.6110  300.5583  300.1138  298.1893  295.8048  292.6066
  261.1215  264.3104  269.8167  276.3402  282.8985  289.0375  294.4682  298.5589  300.4885  300.0977  298.2719  295.9757  292.8646
  261.1215  264.1168  269.5159  276.0681  282.7558  289.0185  294.4785  298.5205  300.4280  300.1108  298.4023  296.1727  293.0819
  261.1215  263.8840  269.1576  275.7515  282.6034  289.0205  294.5157  298.4968  300.3789  300.1528  298.5768  296.3898  293.2515
  261.1215  263.6193  268.7531  275.4003  282.4458  289.0432  294.5788  298.4886  300.3424  300.2222  298.7902  296.6205  293.3681
```

Figure B.2: First rows and columns of january temperature file

Appendix C

Population file

The population file is a text file in which there are the information about the density of the world population. It has to be downloaded from the SEDAC website[19] and the user can choose it according to its needs. In this case, for the valdiation of the results, the file choosen is the density one, the year can be decided as the user wants. It has the first 6 rows for the explanation of the file, as indicated in the figure C.3:

- the rows number;
- the columns number;
- the information of the data in the corners;
- the accuracy of the table;
- the no data value, which identifies the cell with no data about the population available, it means that it can be a sea region or a zone without reliable data;

In figure C.4 there is an example of a part of the file, it can be noticed that the value -9999 corresponds to the absence of data or to the zero value instead, the other numbers identify the number of people on square kilometer, in that specific cell. It is also reported an example of one of the post-processing of ORCA tool: the population density map. It can be completely managed by the user who can decides which resolution and which file taking into consideration. The tool then uses these information for displaying of a world map, divided in cells with the dimension of the resolution chosen.

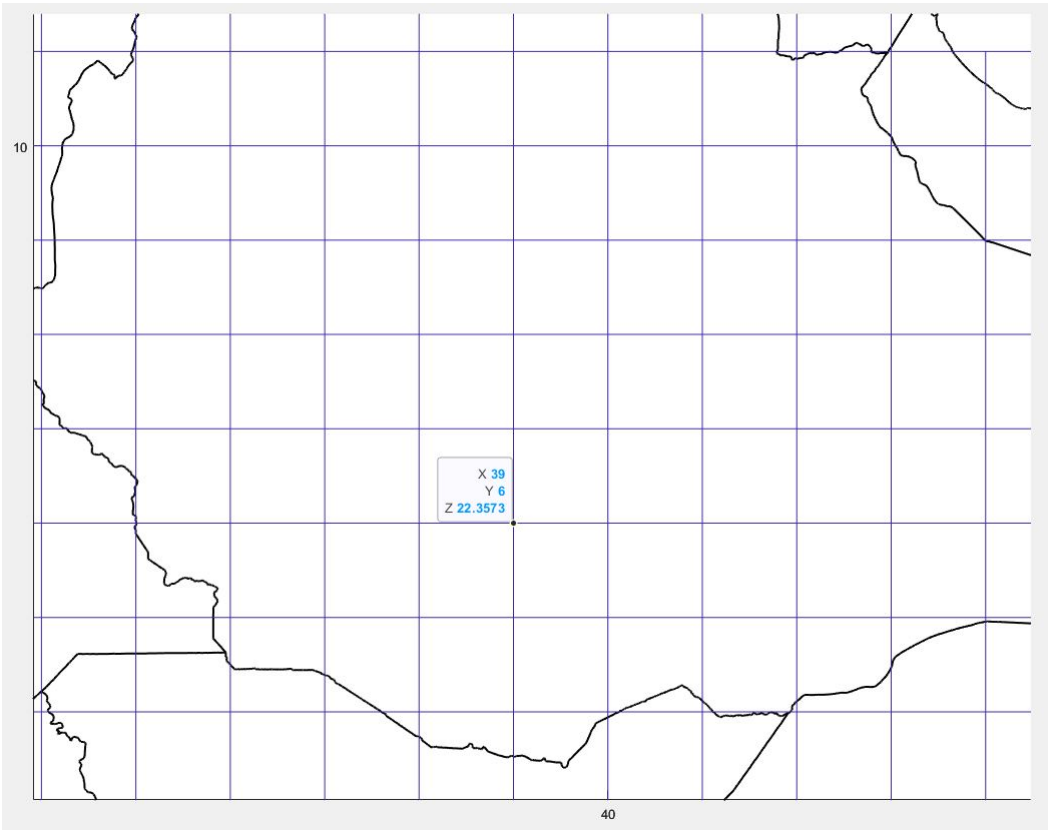

Figure C.1: Subdivison of the world map in cells

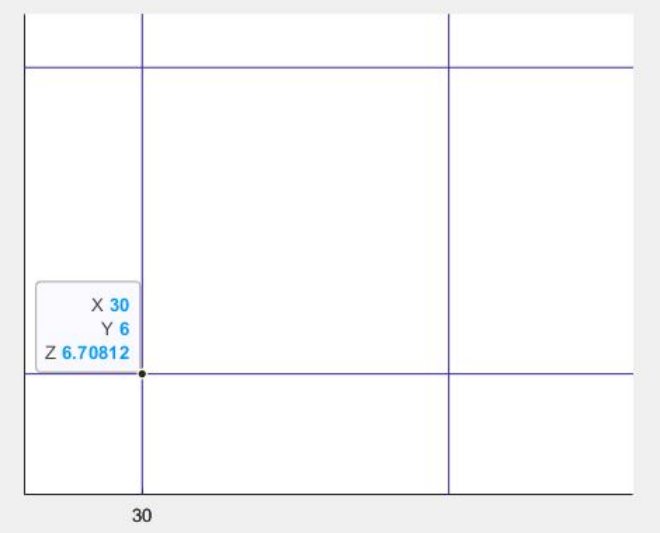

Figure C.2: Cell example

```
ncols         360
nrows         180
xllcorner     -180
yllcorner     -90
cellsize      1
NODATA_value  -9999
```

Figure C.3: First rows of the file

```
-9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999
-9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999
-9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 0.0652151 -9
-9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999
079528 -9999 -9999 -9999 0.079528 -9999 -9999 0.07952788 0.079528 -9999 -9999 -9999 -9999 -9999
0.03138881 -9999 -9999 0.03138881 -9999 -9999 -9999 0.0652151 -9999 0.0652151 -9999 -9999 -9999
-9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999
9 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -999
```

Figure C.4: Extrapolation of the density population file

Appendix D

Output file

The output file of ORCA depend on the module that has been chosen. The launching mode gives as output a text file containing all the fragments impact points and impact velocity in every single instant of the trajectory file. The main numbers, expected number of victim and the probability of almost one victime, are on the top of the file for an easy reading. The deorbiting module's file, instead, has the information on the impact points of the fragments, impact velocity and the risk numbers, only for the instant of time of the reentry. Considering the non

```
12-Jan-2024 12:32:09
Expected number of victim
1.247953e-07
Probability of almost one victim
1.247953e-07
Fragment number  Longitude[deg]    Latitude[deg]    Impact velocity[m/s]

              1     -57.799689      41.319807      13.961374
              2     -58.126046      42.521569      44.378508
              3     -58.187181      42.728083     104.759406
              4     -58.150762      42.609511      51.405434
              5     -58.178466      42.705819      65.899267
              6     -58.172623      42.685931      61.462298
              7     -58.178839      42.707066      66.247603
              8     -58.183992      42.723858      72.770896
              9     -58.177213      42.701598      64.807535
             10     -58.028672      42.168929      29.148555
             11     -58.110902      42.467255      41.085870
             12     -57.887183      41.646114      18.510127
```

Figure D.1: Output file launching module

nominal simulation the outputs file are slightly different because of the presence of the simulation number, as shown in figure D.2.

```
01-Feb-2024 15:04:10
Initial condition ECI Avio
-2404070.000000 1516200.000000 6268840.000000 -4022.090000 5472.140000 -3156.990000
Initial time 7454.080000 s
Expected number of victim
3.283526e-08
Probability of almost one victim
3.283526e-08
```

Simulation number	Fragment number	Longitude[deg]	Latitude[deg]	Impact velocity[m/s]
1	1	92.723514	-10.166399	13.966035
1	2	92.647819	-12.243239	44.393720
1	3	91.668183	-16.204249	104.795320
1	4	92.376120	-12.071677	51.423092
1	5	91.983058	-13.224064	65.921922
1	6	92.254104	-13.478222	61.483415
1	7	92.279133	-12.800253	66.270392
1	8	92.309982	-14.968014	72.795882
1	9	92.462666	-12.388939	64.829838
1	10	93.157674	-14.541659	29.158442
1	11	91.480037	-18.964114	41.099825
1	12	93.122168	-9.774754	18.516359
2	1	95.627859	2.100686	13.965860
2	2	94.962994	0.170784	44.393876
2	3	94.507197	-2.163243	104.795909
2	4	95.051581	0.290515	51.423223
2	5	94.318745	-2.257557	65.922213

Figure D.2: Output file deorbiting with scattering

Bibliography

[1] Tommaso Sgobba, Gary Eugene Musgrave, Gary Johnson, and Michael T. Kezirian. *Safety design for space systems*. Butterworth-Heinemann, 2009 (cit. on pp. 1, 43).

[2] *14 CFR Part 450-123*. 2023. URL: https://www.ecfr.gov/current/title-14/chapter-III/subchapter-C/part-450#450.123 (cit. on p. 2).

[3] *14 CFR Part 450-101*. 2023. URL: https://www.ecfr.gov/current/title-14/chapter-III/subchapter-C/part-450#450.101 (cit. on p. 2).

[4] *French Space Operations Act*. 2008. URL: http://legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000018931380&fastPos%20=9&fastReqId=1846263462&categorieLien=cid&oldAction=rechTexte (cit. on p. 2).

[5] *Vega's Family*. 2017. URL: https://www.eoportal.org/other-space-activities/vega-evolution#vega-launcher-evolution (cit. on p. 6).

[6] *Vega User's Manual issue 4 review 0*. Arianespace. 2014 (cit. on pp. 8, 9).

[7] E.Nardi, G.Cecchetti, and M.Marcone. «Evolution and Development of the Vega Launcher Family and Lessons Learned». In: (2022) (cit. on p. 8).

[8] E.Nardi, G.Cecchetti, and M.Marcone. «Vega Family Enhanced Flexibility for Multipayload Mission». In: (2022) (cit. on pp. 9, 11).

[9] Defense Mapping Agency. «Department of Defense World Geodetic System 1984: its definition and relationships with local geodetic systems». In: *Defense Technical Information Center* (1991) (cit. on p. 15).

[10] JM Picone, AE Hedin, D Pj Drob, and AC Aikin. «NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues». In: *Journal of Geophysical Research: Space Physics* 107.A12 (2002), SIA–15 (cit. on p. 19).

[11] *Space weather data*. 2023. URL: https://celestrak.org/SpaceData/SW-Last5Years.txt (cit. on p. 20).

[12] Sushil Chandra, Eric L Fleming, Mark R Schoeberl, and John J Barnett. «Monthly mean global climatology of temperature, wind, geopotential height and pressure for 0–120 km». In: *Advances in Space Research* 10.6 (1990), pp. 3–12 (cit. on p. 20).

[13] *Layers of the atmosphere*. 2023. URL: https://www.noaa.gov/jetstream/atmosphere/layers-of-atmosphere (cit. on p. 21).

[14] F Landis Markley. «Approximate Cartesian state transition matrix». In: *The journal of the Astronautical Sciences* 34.2 (1986), pp. 161–169 (cit. on p. 29).

[15] Ana Paula Marins Chiaradia, Helio Koiti Kuga, Antonio Fernando Bertachini de Almeida Prado, et al. «Comparison between two methods to calculate the transition matrix of orbit motion». In: *Mathematical Problems in Engineering* (2012) (cit. on p. 31).

[16] Conrad Schiff. «Adapting covariance propagation to account for the presence of modeled and unmodeled maneuvers». In: *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. 2006, p. 6294 (cit. on p. 31).

[17] FAA Associate Administrator for Commercial Space Transportation. *Flight Safety Analysis Handbook*. FAA, 2011 (cit. on p. 66).

[18] *14 CFR Part 417-207*. 2024. URL: https://www.ecfr.gov/current/title-14/chapter-III/subchapter-C/part-417/subpart-C/section-417.207#p-417.207(a) (cit. on p. 66).

[19] *Population files*. URL: https://sedac.ciesin.columbia.edu/data/collection/gpw-v4/sets/browse (cit. on p. 72).