Collision Probability through Orbital Uncertainty Propagation

The Hybrid Differential Algebra and Gaussian Mixture Model Approach

Mireia Leon Dasi

Technische Universiteit Delft

Collision Probability through Orbital Uncertainty Propagation

The Hybrid Differential Algebra and Gaussian Mixture Model Approach

by

Mireia Leon Dasi

in partial fulfillment to obtain the degree of Master of Science
at the Delft University of Technology.

Cover image: artist impression of collision risk of space debris with the ISS. Background image credit: NASA. Space debris image corresponds to ATV re-entry. Space debris image credit: David Sliski of Dexter and Southfield Schools.

Student number:	5048397	
Project duration:	May 10, 2021 – December 31, 2021	
Thesis supervisor:	Dr. ir. E. Mooij,	TU Delft

TU Delft

Preface

Caminante, son tus huellas
el camino, y nada más;
caminante, no hay camino,
se hace el camino al andar.
Al andar se hace el camino,
y al volver la vista atrás
se ve la senda que nunca
se ha de volver a pisar.
Caminante, no hay camino,
sino estelas en la mar.

Antonio Machado, “Probervios y Cantares"

This MSc Thesis represents the final step of my studies at TU Delft, to obtain the title of aerospace engineer. As the above poem states, the road is made as you explore, and when turning your head, there lies the path that you have ridden, not again for you to thread. Upon completion of this experience, I realise the challenges and growth that has resulted from the years of work. In this university, I have developed as an engineer, researcher, teacher, and, I hope, a more critical thinker. I trust that I will keep applying this learning in my future career. I chose the topic of this thesis driven by the ideal of participating to the sustainable development of operations in space, without much further regard on the actual work that it entailed. Considering this, I am extremely happy to have enjoyed so much this work, which incorporated mathematics and engineering in a perfect balance. I apologise in advance if the reader does not find this proportion to be in such a perfect equilibrium.

This project has been possible due to the advice, company and help of many people, and I would like to dedicate a few words to them. First, I would like to thank my supervisor, Dr. Ir. Erwin Mooij, who knew how to motivate me to push through the harder phases of this project. His questions, although sometimes turned quite existentialist, always helped me to gain more insight on the task at matter and guide my research. Also a special mention to the graduation committee, Dr. Ir. Wouter van der Wal and Dr. Ir. Erik-Jan van Kampen for agreeing to participate in this defence and taking interest in the topic.

I am truly fortunate that the number of people that have accompanied and encouraged me through this process would not fit on a single acknowledgement page. I hope that these words can express my gratitude towards the friends and family who participated in this experience. These years in Delft have been truly special, and I believe that all my friends here (old and new) have made a great impact in my life. For all of you who stayed the long hours at the library, cheering me up in times of debugging and celebrating in times of successful results. For those who have seen me in "programming mode" for months (or years), and came back the morning after. For those of you who invited me over for dinner when my fridge only contained instant noodles and those who took advantage of every opportunity to come back and spend a few days in Delft. For those who are with me from the beginning, and those who became inseparable in a very short time. For all of you, thank you. I hope to have also risen to the occasion.

Finally, a very special mention to my family, who over the years could have gained a degree in engineering just from hearing me study and worry about exams, projects and code. To my parents, Àngels and Fidel, and my brother Andreu, simply thank you. For being always there and instilling in me the sense of curiosity.

Mireia Leon Dasi
Valencia, January 2022

Abstract

The risk of collision between space objects has largely increased in the recent years, becoming a considerable threat to satellite operations and human spaceflight. This problem is expected to aggravate in the following years due to the deployment of satellite mega-constellations and the result of current collision events. Therefore, accurately computing the risk of collision for any type of encounter is a requirement to ensure a safe future for space activities. Current methods to compute collision probability can be divided into two categories: accurate methods that require a high computational load and faster methods that limit the type of encounter geometry. There exists a research gap for methods that can accurately compute the risk of collision for low relative velocity encounters in a time that allows to perform the avoidance manoeuvre. Moreover, current methods rely on simplified shapes for the satellites (generally a sphere), which introduce a large overestimation in the computation of collision risk, especially for big, elongated geometries such as the case of the International Space Station (ISS). The goal of this research is to develop a method to compute the collision probability for any type of encounter geometry with high accuracy and applicable to any vehicle shape.

The limitations from current methods come from four simplifications that are applied to reduce the computational load. On one hand, in the uncertainty propagation segment, conventional methods linearise the dynamics of motion and assume that the uncertainty distribution of the satellite position and velocity remains Gaussian. On the other hand, in the collision probability calculation process, the problem is simplified to assume a two-dimensional encounter and the vehicles shape consists of simple geometries. To avoid these simplifications, the methodology developed in this work relies on the hybrid Differential Algebra and Gaussian Mixture Model method for uncertainty propagation, and a collision probability integration with a multi-sphere model that can be adapted to any shape. The differential algebra methodology allows to integrate the Taylor series expansion of the final state with respect to a deviation from the initial state by operating in a computer environment based on Taylor series operations. This allows to propagate the uncertainty without linearising the dynamics. Moreover, the Gaussian assumption for the satellite uncertainty is dropped by using a combination of Gaussian probability density functions that can approximate any probability distribution. Regarding the collision risk, it is computed by integrating the flux rate of probability over the surface of the object over the time of the encounter. By following this methodology, no assumptions are made on the relative velocity or uncertainties of the body, which allow to apply the technique to any type of encounter. Finally, the method is extended from a single sphere to any geometry by developing a numerical quadrature technique based on a combination of spheres to model the real volume. The combination of these methodologies to compute the collision probability is labelled as the DA-GMM method.

To test the method, a sensitivity analysis is run and five scenarios are simulated to compute the collision probability: two with fictional data for verification and three real cases with tracking data. The sensitivity analysis provides a reference for the recommended settings of the method depending on the intended application. From this, it is concluded that using third order Taylor series expansion with 51 Gaussian mixture elements provides the best results for general applications. The method is used then to verify two test cases that present low velocity encounters. It is found that the method improves the accuracy in calculating the collision probability by more that 70% with respect to the best result by conventional methods. Following, an analysis of the effect of lead time and initial state uncertainty on the calculated collision probability is performed for a scenario where the nominal trajectories lead to a collision. It is found that an improvement in the tracking accuracy of one order of magnitude can result in a decrease in the computed collision probability of two orders of magnitude. This outcome is very useful for the interpretation of the following real-life encounters.

For the real scenarios, tracking data is used to compute the collision probability and encounter time. The first case studied is the Cosmos-2251/Iridium-33 collision that occurred in 2009. Second, the close encounter between the rocket body Chang Zheng-4C and satellite Cosmos-2004 in 2020, that was predicted to have a probability of collision of 20% but fortunately did not occur. Finally, a screening of close approaches between the ISS and the debris resulting from the anti-satellite missile test that destroyed satellite Cosmos-1408 on November 2021 is performed. The collision probability is calculated for those elements that enter within 25 km of the ISS. It is found that during this screening, one piece of debris presents a high threat to the ISS.

For this piece, the collision probability is calculated with the multi-sphere model to characterise the effect of modelling the real shape of the satellite.

It was found that the DA-GMM method was able to accurately predict the maximum collision probability within 0.2 seconds of the time of the collision/close approach. Moreover, the results obtained match with the outcome of the encounter scenarios. For the ISS close encounter, it was established that by using the detailed multi-sphere model as opposed to a single sphere to describe the collision volume, the collision probability decreased by one to two orders of magnitude (depending on the inclusion off the solar panels). The result is that an encounter that was catalogued as dangerous and required a collision avoidance manoeuvre, becomes inoffensive. As a conclusion, it is verified that the method developed to compute the risk of collision with space debris can accurately predict the collision probability for any encounter geometry fulfilling operational constraints. Moreover, it is proven that modelling the real shape of the ISS has a large impact on the estimated collision risk and the consequent actions.

Nomenclature

Abbreviations

ABM	Adams-Bashforth-Moulton
BS	Bulirsch-Stoer
CAM	Collision Avoidance Manoeuvre
CAN	Close Approach Notification
CARA	Conjunction Assessment Risk Analysis
CDM	Conjunction Data Message
CORAM	Collision Risk Assessment and Avoidance Manoeuvre
DA	Differential Algebra
DA-ASAT	Direct Ascent Anti-Satellite Missile
DA-GMM	Hybrid Differential Algebra and Gaussian Mixture Model method for collision probability calculation
DACE	Differential Algebra Computational Engine
DOPRI	Dormand-Prince
EGM2008	Earth Gravitational Model from 2008
ERA	Earth Rotation Angle
ESA	European Space Agency
FPE	Fokker-Planck equation
GEO	Geostationary Earth Orbit
GME	Gaussian Mixture Element
GMM	Gaussian Mixture Model
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HAC	High Accuracy Catalogue
ISO	International Organisation for Standardization
ISS	International Space Station
LEO	Low Earth Orbit
MECSA	Method of Equivalent Cross-section Area
MEE	Modified Equinoctial Elements
NASA	National Aeronautics and Space Administration
NRLMSISE-00	US Naval Research Laboratory Mass Spectrometer Incoherent Scatter Radar to Exosphere released on year 2000
ODE	Ordinary Differential Equation
PC	Polynomial Chaos
PDF	Probability Density Function
RKF	Runge-Kutta-Fehlberg
RSSS	Russian Space Surveillance System
SGP	Simplified General Perturbations

SRP	Solar Radiation Pressure
SSN	Space Surveillance Network
STT	State Transition Tensor
TCA	Time of Closest Approach
TDB	Barycentric Dynamical Time
TLE	Two-line Element
TOPO	Trajectory Operations Officer
Tudat	TU Delft Astrodynamics Toolbox
USM	Unified State Model
USSPACECOM	U.S. Space Command
UT	Unscented Transformation
UTC	Coordinated Universal Time

Greek Symbols

α	Angle of attack [rad]
β	Sideslip angle [rad]
γ	Flight-path angle [rad]
δ	Infinitessimal change
δ	Latitude [rad]
θ	True anomaly [rad]
μ	Mean of a Gaussian distribution
μ	Gravitational parameter [$\mathrm{m^3/s^2}$]
ν	Mean of univariate Gaussian distribution
ρ	Atmospheric density [$\mathrm{kg/m^3}$]
$\boldsymbol{\rho}$	Miss vector [m]
σ	Standard deviation
σ	Bank angle [rad]
τ	Longitude [rad]
χ	Heading angle [rad]
ω	Argument of perigee [rad]
Ω	Right ascension of the ascending node [rad]
ω_E	Angular velocity of the Earth [rad/s]

Notation

$\boldsymbol{C}$	Transformation Matrix
$\hat{f}$	Approximation to function f
$f^{(1)}$	First derivative of function f
$f^{(n)}$	n^{th} derivative of function f
$\boldsymbol{v}_x$	Projection of vector v into axis x
$\boldsymbol{x}$	Vector
$\dot{x}$	Derivative of variable x
$\hat{\boldsymbol{x}}$	Unit vector $\boldsymbol{x}$
x_0	Variable x at initial time, $t = 0$
x_f	Variable x at final time, $t = t_f$

x^i	i^{th} element of vector x
$\boldsymbol{X}$	Matrix
$\boldsymbol{X}^T$	Transpose of $\boldsymbol{X}$
$\boldsymbol{X}^{-1}$	Inverse of $\boldsymbol{X}$
$\boldsymbol{X}^{ij}$	Element in row i, column j of matrix $\boldsymbol{X}$

Latin Symbols

a	Semi-major axis [m]
$\boldsymbol{a}$	Acceleration vector [m/s^2]
A	Area [m^2]
$\boldsymbol{a}_D$	Aerodynamic acceleration [m/s^2]
$\boldsymbol{a}_G$	Gravitational acceleration [m/s^2]
$\boldsymbol{a}_{SRP}$	Solar radiation pressure acceleration [m/s^2]
c	Speed of light in a vacuum inertial frame [m/s]
C_D	Drag coefficient [-]
C_R	Solar radiation pressure coefficient [-]
e	Eccentricity
E	Objective function
E[ ]	Moment of a statistical distribution
$\boldsymbol{f}$	Force vector [N]
i	Inclination [rad]
k	Expansion order
K	Ballistic coefficient [m^2/kg]
L_2	Fitting metric
m	Mass [kg]
n	Mean motion [rad/s]
p	Semi-latus rectum [m]
$p(\boldsymbol{x})$	PDF of the state variable
$\boldsymbol{P}$	Covariance matrix of a Gaussian distribution
P_c	Collision probability
$p_g(x;\mu,\boldsymbol{P})$	Gaussian distribution
$\boldsymbol{Q}$	Covariance matrix of univariate Gaussian distribution
$\boldsymbol{r}$	Position vector [m]
R	Combined hard-body radius [m]
t	Time [s]
u	Argument of latitude [rad]
$U(a,b)$	Uniform distribution bounded between a,b
$\boldsymbol{v}$	Velocity vector [m/s]
w_i	Weight of the ith element in a distribution
x,y,z	Magnitude of position in X,Y,Z axis in Cartesian space
X,Y,Z	Axes in Cartesian space

Contents

1 Introduction

“A threat to activities in outer space for years to come [...] posing a significant risk to the crew on the International Space Station and other human spaceflight activities, as well as multiple countries’ satellites”. This was the first statement from the U.S. Space Command commander Gen. James Dickinson, on the direct-ascent anti-satellite (DA-ASAT) missile test that destructed Russian satellite Cosmos-1408 on November 15 2021, creating a field of debris in Low Earth Orbit (LEO) (U.S. Space Command, 2021). These declarations transmit the challenges and hazards of space operations in the present and foreseeable future. During the first decades of the space age, scientific advance was the main objective of the audacious space exploration, with few considerations on possible long-term consequences. After this initial euphoria, the scientific community started to rise awareness on safe exploration topics and space debris mitigation. As the number of Earth-orbiting objects rose, with the surge of satellite mega-constellations and affordable space technologies such as CubeSats, the latter became of particular concern. Once the operational life of a man-made object ends, it becomes a piece of space debris. The fate of these objects is clear: either re-enter Earth through an atmospheric decay process, stay in orbit or be transported to a different orbit. The longer the debris stays in orbit, the higher the probability that it will collide with another object, threatening space exploration.

The danger posed by space debris is twofold. First, the consequences of the collision itself, which can be fatal for the satellite. For space debris of ~ 1 mm, a collision in LEO can lead to the destruction of components, while a debris of 10 cm causes a partial or complete break-up of the spacecraft. This leads to the second threat posed by space debris, the cascade effect described by the *Kessler syndrome*. In this scenario, each collision creates multiple pieces of debris, increasing the overall probability of encounter event between space objects and leading to a debris belt encapsulating the Earth (Kessler and Cour-Palais, 1978). Therefore, preventing the creation of space debris is of foremost importance, and multiple mitigation plans have been devised. These include satellite end-of-life strategies to ensure that the operational bands are clear from inactive satellite, design for demise strategies to reduce the risk of explosions and passive de-orbit techniques to remove existing debris. However, until the space environment is completely clean of debris the risk is still present, which demands for collision detection and avoidance strategies.

Estimation of collision risk is the first step to assess the need for collision avoidance manoeuvres that prevent the destruction of active satellite and ensure the safety of space operations. This is the topic of this thesis work. The risk of collisions cannot be directly assessed from the nominal trajectories of the satellites due to the uncertainties in the orbit determination process and propagation environment. The concept of collision probability is therefore introduced, to assess the risk of collision in this scenario. While this type of uncertainty propagation problems are usually computed using Monte Carlo sampling techniques, this process is extremely time consuming for this application and cannot be used for operational collision avoidance. Therefore, alternative techniques have been developed, which have been traditionally divided into two categories. First, methods for high relative-velocity encounters that apply linear approximations and neglect the uncertainty in velocity (Foster and Estes, 1992; Patera, 2001; Alfano, 2005). Second, methods for low relative-velocity encounters that require a larger computational load and present some limitations in terms of encounter geometry (Patera, 2003; Chan, 2004; Alfano, 2006). There exists a trade-off between computational load and accuracy, and some scenarios such as cases with very low relative velocity, cannot be correctly predicted in time for collision avoidance.

The identified trade-off leaves a research gap for methods to increase the lead time in collision detection for all types of encounter geometries with high accuracy. Römgens (2011) approaches this topic by studying the application of verified Taylor series integration to obtain a bound of the satellite state and therefore accept or discard the risk of collision. While this can be very useful since it allows to rule out a collision with high certainty, the method fails to estimate the risk when a potential collision is detected. The method developed in this work is based on a combination of the differential algebra technique for Taylor series integration and the Gaussian Mixture Model to obtain a fast and accurate propagation of uncertainty of position and velocity of space objects. This method evades the linear propagation of uncertainty and the assumption of a Gaussian uncertainty distribution, which limit the accuracy of traditional collision probability calculation methods.

A second aspect that requires further research is the inclusion of multiple three-dimensional shapes in the encounter geometry. Traditional methods assume a spherical geometry and some methods have been extended to account for parallelepipeds. The use of simplified satellite shapes is justified for most satellites and pieces of debris with unknown shape. However, some space objects, such as rocket launchers, have very elongated shapes which would benefit from a more detailed geometry. This is especially important for the International Space Station (ISS). Up to now, the collision risk with the ISS has been overestimated to increase the safety of the crew. However, recent events such as the Russian ASAT missile test have revealed the need for a high-accuracy determination of the collision risk. Following this event, on November 30, 2021, an extra-vehicular activity was delayed due to " the lack of opportunity to properly assess the risk it could pose to the astronauts". Accurately determining the collision risk for the ISS by including the satellite geometry in the collision probability estimation becomes the second topic of this research.

1.1. The space debris environment

The launch of Sputnik 1 on October 4, 1957, marks the beginning of the space age. To the present, more than 6,000 rockets have successfully launched, placing 10,680 satellites in Earth orbit (ESA, 2021b). Approximately two thirds of this population has reached its end of life, and have either re-entered or stayed in uncontrollable orbits, becoming the source of space debris. The term space debris refers to "all man made objects including fragments and elements thereof, in Earth orbit or re-entering the atmosphere, that are non functional"(IADC, 2020). The high and rapidly increasing number of objects placed in Earth orbit elevates the risk of collisions.

Initially, the space debris is formed by objects launched to space (derelict satellites and rocket upper stages). These objects have a large size and are easily tracked by space Surveillance Networks. However, due to separation events such as explosions or collisions, the space debris population increases. As observed in Figure 1.1, the number of objects in orbit has doubled in the past ten years, and most of this population is found in LEO. Some of these separation events, such as surface degradation and release of fuel particles, produce debris of size smaller than 1cm. The effect of such small pieces of debris on other satellites can be mitigated through shielding. Larger pieces with sizes of 1-10 cm can cause the malfunction or breakup of active satellites, contributing to aggravate the Kessler syndrome. In this figure, an abrupt increase in number of objects is observed between the years 2009 and 2011. This is directly related to the collision that occurred between the satellites Cosmos-2251 and Iridium-33, in February 2009. From this tragic event, over 2,000 pieces of debris larger than 10 cm were created, eventually spanning all the altitude shell and obstructing satellite operations.

Based on the predominant regions exploited by satellite missions, the largest pieces of debris (>10 cm) are located in the LEO and GEO regions, especially at 800 km altitude (Horstmann et al., 2018). Fragments located at altitudes below 300 km naturally re-enter due to the drag effect from the still significant atmospheric density. When reaching higher altitudes, the drag effect becomes negligible and the particles do not naturally deorbit. To prevent the spatial density collapse in the regions of high interest, two protected areas with special debris mitigation guidelines are defined. The LEO protected region extends as a spherical shell from the Earth surface to an altitude of 2000 km. The GEO protected region is a spherical shell segment extending from the geostationary altitude with a radius of +/- 200 km and an angle of +/- 15 degrees (IADC, 2020). With the expected increase in objects launched to space,mainly due to the expanding sector of satellite mega-constellations, several legal and operational actions are taken to minimise the footprint in the space environment. To prevent the risk of self-fragmentation due to explosion, satellites undergo end of mission passivisation, which includes depleting the batteries and disconnecting the solar arrays. To protect the satellites from collisions with small pieces of debris, shielding mechanisms are developed. The satellites launched into LEO must be able to clear the protected region in less than 25 years. Satellites in GEO-protected region must be transferred to a graveyard orbit (ISO Central Secretary, 2019). And for objects that cannot deorbit (e.g., due to mission failure), active debris removal systems are under research to clean the space environ-

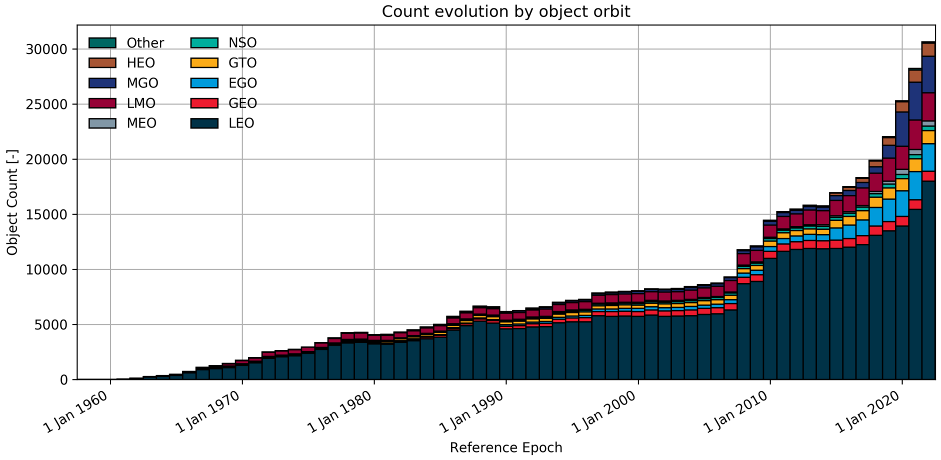


Figure 1.1: Evolution of number of objects > 10 cm identified in orbit (ESA, 2021a).

ment and ensure continued accessibility.

Despite the efforts taken to minimise the effects of space debris on active satellites, the risk of collision is still present. To prevent this, the collision risk is monitored and satellite operators are notified. Improving the prediction capability of space debris collisions is essential to the safety of the future space environment.

1.2. Collision avoidance practices

Space agencies and private satellite operators currently rely on the data provided by the U.S. Space Command (USspaceCOM) to detect potential risk of collision. These entities then apply different collision probability calculation methods to further constrain the collision risk and decide on the mitigation strategy. The process begins with the formation of a space High Accuracy Catalogue (HAC) with the position, velocity and uncertainties of objects larger than 10 cm with the satellite observations from the US. Space Surveillance Network (SSN). These data, together with satellite ephemeris provided by satellite operators is propagated accounting for special perturbations every 24 hours and is screened to detect possible collisions. When a possible collision is found, a notification is sent to the satellite operator via a Conjunction Data Message (CDM) and/or a Close Approach Notification (CAN) email. The lead time and reporting criteria considered are summarised in Table 1.1. Four cases are considered depending on the target orbit (LEO or Deep space) and the data source (ephemeris or high accuracy catalogue). As observed, the criteria are more strict for LEO satellites, for which the collision probability is computed. When a criterion is satisfied, a notification is sent to the satellite provider. In case of no response, the satellite provider is contacted by phone when the risk becomes extreme ($P_c \geq 10^{-2}$ in the case of LEO objects).

Table 1.1: Advanced reporting criteria established by the USspaceCOM (SpaceTrack, 2020, p9)

Notification Method	CDM	CDM& CAN email	CDM, CAN email & phone call
Deep space HAC	TCA ≤ 10 days & all results w/in 5km x 5km x 5km	TCA ≤ 3 days & Overall miss ≤ 5km	TCA ≤ 3 days & Overall miss ≤ 500 m
Deep space ephemeris	TCA ≤ 10 days & all results w/in 20km x 20km x 20km	TCA ≤ 3 days & Overall miss ≤ 5km	N/A
LEO HAC	TCA ≤ 5 days & $P_C \geq 10^{-7}$	TCA ≤ 3 days & Overall miss ≤ 1km& $P_C \geq 10^{-4}$	TCA ≤ 3 days & Overall miss ≤ 1km& $P_C \geq 10^{-2}$
LEO ephemeris	TCA ≤ 7 days & all results w/in 2km x 25km x 25km	TCA ≤ 3 days & Overall miss ≤ 1km& $P_C \geq 10^{-4}$	N/A

Although this methodology is generally effective for generic satellites, a different approach is required for the ISS. In this case, the presence of astronauts and the complexity of the operations involved, require for a more precise collision probability estimation procedure. The collision avoidance strategy of the ISS is decided by the Trajectory Operations Officer (TOPO) flight controller (Dempsey, 2018). The process starts with the screening of all objects in the High Accuracy Catalogue against the ISS. This screening occurs every eight hours. The TOPO is notified if an object is predicted to pass within a ±2km (local vertical) × ± 25 km × ± 25 km (local horizontal) volume in the next 72 hours.

When TOPO is notified, the collision probability is calculated with updated tracking data, and the fight control teams in Houston and Moscow are notified if one of the following conditions are met:

- $TCA \leq 48$ hours
- Local vertical miss ≤ 0.5 km or $P_c \geq 10^{-6}$

The collision probability is then continuously updated and the following levels of concern are defined:

- Green: $P_c < 10^{-5}$. No action required.
- Yellow: $P_c \geq 10^{-5}$. A CAM should be attempted prior to TCA unless there is a major impact to the ISS operations.
- Red: $P_c \geq 10^{-4}$. A CAM should be performed prior to TCA unless the burn itself will place the crew at greater risk.
- Black: $P_c \geq 10^{-2}$. A CAM must be performed prior to TCA unless the burn is in the final minutes of a docking operation.

As it is observed, a CAM is not always the safest option when dealing with a space debris threat. There are a number of operations in the ISS that could be disrupted by a collision avoidance manoeuvre, which can even lead to put the crew at risk. The collision probability with the ISS is calculated considering a spherical shape, which overestimates the collision risk. The approach developed in this work is intended to reduce overestimation, thus providing more information to deal with these safety trade-offs.

1.3. Research question

After careful review of the past and present literature regarding collision probability calculation, the research questions for the thesis work are formulated. These questions will guide the direction of this research and will be answered at the end of the work. The main goal of this thesis work is to develop a new method to compute collision probability that can contribute to the task of space debris mitigation through a faster and more accurate assessment of the collision risk for any type of encounter geometry. Through a literature review of methods to propagate the uncertainty, the hybrid Differential Algebra and Gaussian Mixture Model method (DA-GMM) has been selected as the most promising approach to achieve this goal. Although the summary of alternative methods and the explanation of the DA-GMM technique is detailed in Chapter 2, a brief introduction is provided here, to provide a background for the research questions. The term differential algebra refers to a computational technique that allows to operate using Taylor series expansion coefficients instead of floating point operations. The result is that the Taylor series expansion of the final state with respect to a deviation from the initial state can be readily computed, providing extra information that is used in the propagation of uncertainty. The concept of a Gaussian mixture model consists of approximating the probability density function by a mixture of Gaussian distributions, which allows to consider complex distributions without introducing extra complexity.

The research questions aim to guide and provide a clear statement of the work objectives. First, the operational requirements that the DA-GMM method must meet are stated:

R1. The percentage error in the computed collision probability must be $\Delta P_c < 8\%$ to correctly estimate the order of magnitude of the collision risk.

R2. The method's computational load must allow in any case to compute P_c by the last station passage before time of closest approach (TCA).

With this insight, two research questions are formulated. The first one is related to the full DA-GMM method, while the second one is specific to the computation of collision probability for the ISS using the method of equivalent cross-section area (MECSA) proposed by Chan (2008). This method allows to consider the real shape of the satellite for the cross-section perpendicular to the main debris flow, by converting every element of the body to a sphere of equivalent area. The computation of collision probability for the ISS has been identified as a potential source of improvement. Collision probability must be carefully assessed in this case due to the catastrophic consequences that the event would have, which impact human life and the future of space exploration. However, the current methods to compute collision probability for the ISS induce large overestimation, which leads to an average of one collision avoidance manoeuvre (CAM) a year. These manoeuvres can also pose a risk to life and equipment in the ISS. Reducing overestimation can reduce the risk of astronauts in the ISS and allow to make informed decisions when multiple risks are detected simultaneously. Moreover, this involves a reduction in energy associated to these avoidance events and the posterior return to the nominal orbit. The second research question is devoted to study the field of accurate collision detection for the ISS, using the MECSA.

Q1. How can the hybrid DA-GMM method reduce the time to compute collision probability in LEO satisfying the operational requirements?

 Q1.1. What is the error derived from propagating the trajectory using Taylor expansion integration in the DA framework?

 Q1.2. What environment and acceleration models can be included to meet the operational requirements?

 Q1.3. What is the maximum lead time that can be achieved within the operational requirements?

 Q1.4. How does the accuracy of the DA-GMM method relate to alternative collision detection algorithms?

 Q1.4. Is the method able to predict the Iridium-33/Cosmos-2251 collision and the encounter time?

Q2. How can the MECSA be applied to reduce the over-estimation of collision probability for the ISS?

 Q2.1. What is the error derived from considering a constant attitude and only one cross section of the vehicle?

 Q2.2. Is the MECSA able to discard potential collisions identified by the verified interval orbit propagation that did not occur?

 Q2.3. Can this method be used to determine the best collision avoidance strategy?

The second research question is related to the ISS, since this is the reference vehicle under study. However, the method developed is intended to be applicable for any satellite with a known three-dimensional shape.

1.4. Report outline

The thesis work that allows to answer the research questions is structured as follows. The first part of the thesis provides the required background to perform this research. Chapter 2 introduces the mission heritage that allows to understand the developed theory on uncertainty propagation and collision probability calculation methods. In this chapter, a literature review of the techniques developed for similar applications covering their characteristics and limitations is provided. Moreover, the reference mission and vehicle are introduced and the mission requirements are stated and a road-map to follow the methodology is provided. Following with the framework required to develop the DA-GMM method, Chapter 3 discusses the dynamics and environmental models involved in the propagation of trajectories that will be coded in the differential algebra software. The relevant reference frames and frame transformations are introduced and the equations of motion for the satellite are provided. Moreover, the perturbations in acceleration to be included in the dynamics are motivated and the environmental models are selected. Chapter 4 is dedicated to the definition and selection of the uncertainty models for the state and environmental variables. The types of uncertainty distribution included in this work are mathematically introduced and the values selected for each uncertainty distribution are related to measurement errors.

The second part of the thesis involves the mathematical description of the DA-GMM method for collision probability calculation. In Chapter 5, the uncertainty propagation with the hybrid Differential Algebra and Gaussian Mixture Model approach is introduced. This includes the theory and implementation of the Gaussian Mixture Model, the description of Taylor series integration and how it is related to Differential Algebra computation and the reconstruction of the final state uncertainty. The theoretical explanations are combined with examples to facilitate the comprehension of the method. In Chapter 6, the two methods developed for collision probability calculation are explained, again with simplified examples. The multi-sphere method of collision probability developed in this work is explained and illustrated in detail.

The third part of the thesis consists of the simulation, verification, sensitivity analysis and final results of the DA-GMM method for collision probability calculation with space objects. Chapter 7 covers the simulation, verification and validation plan for the developed method. First, the software involved in the simulation and the capabilities developed in this work are explained. The architectural design is described with the help of two flow diagrams and the code optimisation required to reduce the CPU load of the DA integration is explained. Moreover, the numerical methods especially used for the applications developed in this work are introduced, while the remaining numerical methods are explained in Appendix A. The code verification performed for the propagation of uncertainty developed in the Differential Algebra software is detailed and the verification and validation plan for the full model is introduced. Chapter 8 treats the sensitivity analysis on the uncertainty propagation to test the behaviour of the method for different model and method parameters. Finally, Chapter 9 covers the results of the collision probability calculation for a series of test cases that include fictitious and real-life scenarios. The final test performs a screening of the space debris resulting from the Cosmos-1408 destruction and computes the collision probability of the threatening pieces with the single-sphere and multi-sphere model. Chapter 10 concludes the research and provides recommendations for future extensions of this work.

2 Mission heritage

In this chapter, the mission heritage regarding the uncertainty propagation and collision probability methods is introduced. In Section 2.1, the uncertainty propagation methods developed to date for several applications are described. After a trade-off, the hybrid Differential Algebra and Gaussian Mixture Model method is selected as the best option to develop in this thesis work. Section 2.2 introduces the state of the art of collision probability calculation methods and details the assumptions and limitations of each technique. The reference missions and reference vehicles selected for this work are introduced in Section 2.3 and Section 2.4 respectively. Section 2.5 details the mission and system requirements that will guide the development of the collision probability calculation method. Finally, Section 2.6 introduces a road-map describing the methodology followed through the document. The objective of this diagram is to provide the reader with an extra guidance and a place to re-visit during the document.

2.1. Uncertainty propagation methods

In the event of a perfect system, where the satellite state and environment could be accurately modelled, the propagated trajectory of the target and chaser would allow to predict or discard a collision with no need of further calculations. However, this scenario does not occur in real operations. Uncertainty is introduced in the problem, coming from multiple sources. The most apparent is the uncertainty in the satellites state (position and velocity), which is limited by the orbit determination process. Current techniques allow to determine the position of active satellites to centimetre accuracy. This does not apply to space debris, which are more difficult to track and its position is determined with a precision of tens to hundreds of meters. In any case, the uncertainty in the state can never be zero and therefore a collision can never be completely discarded by simply propagating the nominal trajectory. Moreover, there are other sources of uncertainty, such as the characteristics of the environment causing the satellite acceleration or the physical properties of each orbiting body. As a result, a different approach must be followed to estimate the risk of collision with space debris or other active satellites. Once the concept of uncertainty is introduced, the satellite is modelled by a probability density function. From this point, the risk assessment process can be divided into two steps. First, propagate the probability density function (PDF) to the TCA. Second, calculate the collision probability from the propagated probability density function. In this section, a summary of the methods proposed to conduct the first step is provided.

The analytical approach to tackle this problem is through the time evolution of the probability density function $p(\boldsymbol{x},t)$ of the state vector $\boldsymbol{x}$, described by the Fokker-Planck equation (FPE):

$$\frac{\partial p(\boldsymbol{x},t)}{\partial t} = -\sum_{i=1}^{n} \frac{\partial}{\partial x^i}\left[p(\boldsymbol{x},t)\boldsymbol{f}^i(\boldsymbol{x},t)\right] + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\partial^2}{\partial x^i \partial x^j}\left\{p(\boldsymbol{x},t)\left[\boldsymbol{G}(t)\boldsymbol{P}(t)\boldsymbol{G}^T(t)\right]^{ij}\right\} \tag{2.1}$$

where n is the number of dimensions in the problem, $\boldsymbol{f}(\boldsymbol{x},t)$ defines the dynamical evolution of the state, $\boldsymbol{P}(t)$ is the covariance matrix and $\boldsymbol{G}(t)$ is the diffusion matrix. The analytical solution of the FPE can be obtained for simple dynamics (e.g., the two body problem). When other perturbing accelerations are added to the dynamics, this equation becomes unsolvable, and alternative solutions need to be found to propagate the PDF. Subsection 2.1.1 summarises the existing approaches to propagate uncertainty and solve or approximate Equation (2.1).

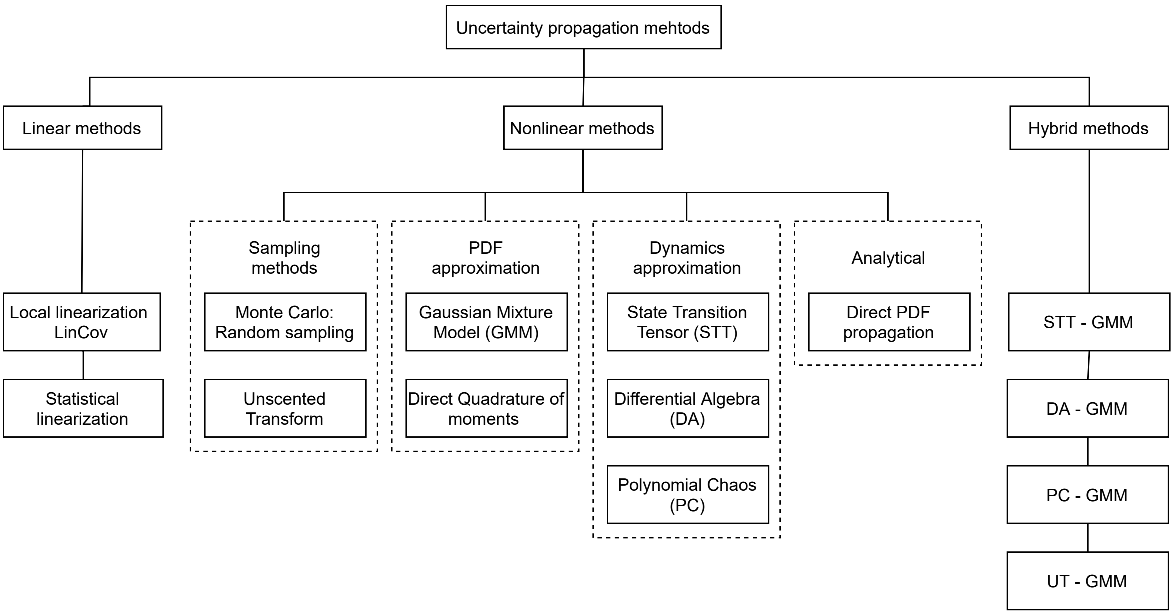

Uncertainty propagation mehtods
Linear methods
Nonlinear methods
Hybrid methods
Local linearization LinCov
Statistical linearization
Sampling methods
Monte Carlo: Random sampling
Unscented Transform
PDF approximation
Gaussian Mixture Model (GMM)
Direct Quadrature of moments
Dynamics approximation
State Transition Tensor (STT)
Differential Algebra (DA)
Polynomial Chaos (PC)
Analytical
Direct PDF propagation
STT - GMM
DA - GMM
PC - GMM
UT - GMM

Figure 2.1: Overview of uncertainty propagation methods

2.1.1. Methods summary

As illustrated in Figure 2.1, multiple methods have been derived to propagate uncertainty through nonlinear dynamics, which can be classified into three categories. First, linear methods involving linearisation of the dynamics that have been traditionally used to obtain a first approximation of the propagated PDF. Second, nonlinear methods that are able to propagate the uncertainty considering the full expression of the dynamics. These are divided into sampling methods, methods that approximate the PDF, methods that approximate the dynamics and methods that solve Equation (2.1) analytically. Finally, there are hybrid methods that combine several techniques to escape the drawbacks of the independent techniques. From these options, linearisation techniques are excluded since they offer a very low accuracy and are not suitable for the purpose of high-accuracy impact prediction. The set of nonlinear methods present several drawbacks that make them suitable for different applications. Sampling methods can offer a very high accuracy at the expense of a high computational cost. Methods that approximate the PDF allow to increase the fidelity of the propagated distribution, but their accuracy is linked to the dynamics propagation method combined with it. Finally, dynamics approximation methods are efficient in accurately integrating the nonlinear equations of motion, but when used alone, these methods do not provide large advantages in terms of computational time compared to regular sampling techniques. To complement the benefits of different techniques, hybrid methods are developed, which allow to obtain a high accuracy with a reduced computational load. In this section, a brief description of each category and the benefits and drawbacks of each method is provided to introduce the trade-off that includes hybrid method and allows to select the best option.

Nonlinear - Sampling methods

Sampling methods rely on the selection of individuals from the initial probability distribution to be propagated through the nonlinear dynamics and reconstruct the final probability. Two methods fall into this category: Monte Carlo sampling and the unscented transformation (UT).

The Monte Carlo approach is the most widely used method to obtain a high accuracy estimate of the propagated distribution. It consists of drawing a large number of random samples from the initial PDF, and using a given method (e.g., numerical integration) to propagate each sample to TCA. In this method, the accuracy is directly related to the population size. With a sufficient number of samples, the results show a very high agreement with the analytical solution (Sabol et al., 2012). To obtain a collision probability with a standard deviation of 10^{-X}, at least $10^{-(X+3)}$ samples are required. For the accuracy required to accept/discard a collision and the need of CAM, this imposes a minimum of 10^7 simulations for each satellite. This number can suppose a prohibitive computational load, especially considering the importance of lead time and alert time in this scenario. As a result, although this method produces results with very high accuracy, it cannot be used for operational collision avoidance.

To reduce the number of samples, deterministic sampling methods have been developed. From these, the unscented transformation approach is the most widely used. The main concept consists in reducing the population size by selecting samples (denoted as σ-points) that capture the mean and covariance of the original distribution. With this method, the number of samples required is twice the number of dimensions in the problem. This reduces the sample size to 12 points in the case of the two-body problem. Naturally, the drawback of this method is a decline in accuracy. This method can approximate the mean and standard deviation of the propagated distribution, but fails to capture the nonlinear effects. Solutions to increase the number of σ-points have been proposed by Adurthi and Singla (2015). However, a large number of points are required to obtain a high accuracy, which would further increase if non-Gaussian distributions are considered. Therefore this method is discarded for the application under study.

Nonlinear - PDF approximation methods

As an alternative solution to describing the population by its samples, the PDF approximation techniques are introduced. These need to be coupled with an integration technique and allow to simplify the propagation process.

A common drawback in uncertainty propagation algorithms is the limitation to propagate Gaussian distributions. However, when an initially Gaussian distribution is propagated trough nonlinear dynamics, it becomes non-Gaussian. To overcome this issue, the Gaussian Mixture Model (GMM) provides an alternative method to reconstruct a non-Gaussian PDF. In this approach, the non-Gaussian distribution is approximated by the weighted sum of a set of individual Gaussian density functions. This concept is further explained in Subsection 2.1.3 and illustrated in Figure 2.2. Coupled with traditional numerical integration, this technique does not provide a large advantage, since the number of samples to propagate is still large, leading to a large computational cost with no visible increase in accuracy. However, this technique can be coupled to other uncertainty propagation algorithms, reducing the number of runs to the number of Gaussian mixture elements (GMEs). This approach is referred to as hybrid method, as illustrated in Figure 2.1.

Similarly to the GMM, the direct quadrature of moments method makes use of a weighted sum of Dirac functions to approximate the initial distribution. These are then propagated to reconstruct the final PDF. However, the propagation of Dirac functions is more complex than that of Gaussian components. Due to the reduced applications and high complexity of this method compared to the GMM, the direct quadrature of moments is discarded as an option for uncertainty propagation.

Nonlinear - Dynamics approximation methods

The most efficient option to reduce the computational load of propagating the uncertainty is though simplifying the dynamics. There are four methods developed to this end, which can be classified into two categories: Taylor integration and response surface mapping. The methods relying on Taylor integration are the state transition tensor, interval analysis and differential algebra. The response surface mapping can be achieved through so-called polynomial-chaos expansions. The purpose and description of these techniques is introduced in this section.

Starting with methods based on Taylor series integration, Park and Scheeres (2007) proposed a method to propagate the initial uncertainties through the Taylor expansion of the dynamics. This was achieved by solving the state-transition tensors, which are the higher-order partial derivatives of the dynamics. These state transition tensors offer information from the final state of the solution related to a deviation from the initial state. In this way, instead of obtaining information from only the central point being propagated, Taylor series integration offers extra information about the final uncertainty distribution, and these integration only needs to be conducted once, as opposed to the multiple propagation of samples required in Monte Carlo techniques. However, to apply this method with high accuracy, high-order Taylor series expansion is required, which increases the computational load. For a fourth order Taylor expansion, the number of partial derivatives to be evaluated is $\sim 10^4$, making the method invalid for operational use when combined with sampling techniques.

Following the concept of a state-transition tensor (STT), interval analysis was introduced by Moore (1968). In this case, upper and lower bounds are set to a truncated Taylor series expansion of the initial state. Through Taylor series integration, the solution of each integration step becomes the initial value for the following step. This results in a bound for the values that the final solution can take. The benefit of this approach is that, instead of combining Taylor series expansion with a Monte Carlo sampling technique, which requires to propagate multiple points, only the extreme values of each variable are propagated. This approach was followed by Römgens (2011) to predict collisions with space debris. The main drawback of this method is the overestimation of the solution set. This is caused by the fact that taking the interval values propagate the uncertainty as a box with constant probability inside and zero probability outside. This overestimates the actual uncertainty and since the bounds of the box (which in reality have a low probability) are propagated, this effect increases with time. As a result, the bounds of the final solution are overestimated and this effect can grow exponentially. This method allows to rule out a collision, but it is impractical to evaluate the risk in cases when a potential impact is detected.

Differential algebra offers an alternative way to deal with the state-transition tensors. It allows to compute the Taylor expansion of an initial value problem in a computerised environment, creating a relation between an initial state perturbation and the final state. In this case, the computational performance of the method to obtain the Taylor series integration is much faster than the numerical integration of the partial derivatives of the state transition tensor. However, if this method is combined with a Monte Carlo sampling algorithm to obtain the density function of the satellite state at the final time, the computational load is still too high. Morselli et al. (2012) found that the method required 19,000 seconds to compute the collision probability with 10^7 samples. Therefore, the method is still not valid for operational collision probability computation on its own.

Finally, the polynomial chaos expansion method (PC) consists in approximating the solution of the differential equation as a weighted sum of multivariate polynomials. To do so, a reduced number of samples is drawn from the initial distribution and propagated through numerical integration (a Monte Carlo technique with reduced sample size). The propagated solutions are used to fit a response surface of the final state as a function of an initial state deviation, using polynomials. The accuracy of the method can be adjusted with the sample size used for the fit and the polynomial order. The method has the advantage that it can be used to propagate any initial distribution and that it is independent of the dynamical model considered. However, the method can be complex to set up, depending on the technique used to fit the polynomial coefficients. The techniques used include least squares regression (Jones and Doostan, 2013), stochastic collocation (Bhusal and Subbarao, 2018), Monte Carlo sampling (Le Maître and Knio, 2010) and compressive sampling (Doostan and Owhadi, 2010). The computational load can be too high if the method is coupled with a Monte Carlo sampling to propagate the uncertainty of the initial state. Moreover, the polynomial chaos expansion provides the relation between a single final time and the initial state. If the relation between more integration steps is required (e.g., to integrate the collision probability with time in a long term encounter), the polynomial fit should be done in multiple time steps, leading to a prohibitive computational load.

Nonlinear - Analytical integration methods

Analytical integration methods rely on solving the Fokker-Planck equation, which describes the time evolution of the flow. The FPE reduces to Liouville's equation when no process noise is considered. The advantage of solving this equation is the high accuracy and low computational load that it requires. Moreover, it allows to propagate the uncertainty from any initial probability density function. The method was used by Kumar (2012) to propagate the uncertainty in the two-body problem, with excellent results. However, the analytical solution can only be obtained for problems with simple dynamics. Adding the J_2 perturbation already makes the equation unsolvable analytically. Weisman et al. (2016) tried to adapt the technique to include the J_2 perturbation by approximating its effect using mean element theory. However, this was already too complex, and adding other perturbations is thus unfeasible.

Hybrid methods

None of the methods introduced so far appears to be useful to achieve a high accuracy propagation of the state uncertainty with a medium-low computational load. Hybrid methods provide an alternative to improve the uncertainty propagation process in terms of accuracy and computational load. Two functions can be identified in the uncertainty propagation process: the treatment of the initial state and the propagation to final time. The initial state can either be sampled (Monte Carlo approach or Unscented Transformation) or approximated. The state propagation to final time can be done through numerical integration, dynamics approximation methods (STT, DA, PC, Interval analysis) or analytical evaluation of the FPE. By properly combining the treatment of the initial state with the propagation method the uncertainty propagation process becomes optimal. The hybrid methods presented up to date consist in combining a Gaussian mixture model with a dynamical approximation method.

When combining the GMM with STT or DA, instead of propagating each sample, the mean and covariance of each GME can be obtained from the Taylor series expansion of the final state, and the final PDF can be reconstructed as the weighted sum. This reduces the number of samples to be propagated from 10^7 to the required number of GMEs. Sun et al. (2019) applied this method to satellite state uncertainty propagation and confirmed that the propagated PDF had a high accuracy, which increased with increasing number of GMEs and Taylor expansion order.

Combining the GMM with PC follows a different procedure. In this case, dividing the initial PDF into multiple GMEs with smaller standard deviation allows to reduce the order of the polynomial. This reduces the number of coefficients to fit and therefore the computational load. The method was applied by Vittaldev et al. (2016) to propagate the uncertainty in the state of satellites.

2.1.2. Trade-off

Based on the characteristics of the available uncertainty propagation methods, the following conclusions have been drawn:

- The Monte Carlo approach requires a high computational cost to provide an accurate solution. As a result, the method is very useful for verification but not valid to compute the collision probability during satellite operations.
- Contrary to the Monte Carlo, the unscented transform method is very fast but cannot capture the non-linear effects and provides a low accuracy solution. Therefore, this method is not valid for accurate collision probability calculation.
- The state transition tensor method integrates the partial derivatives of the equations of motion to obtain the Taylor expansion of the flow. This process of calculating the Taylor expansion results in a high computational cost, for which the method is disregarded.
- Interval analysis is a very useful approach to rule out a collision. However, the method does not provide an insight on the collision probability when a collision is detected. This method was already applied to the topic of collision detection by Römgens (2011). For this work, the objective is to calculate the collision probability and therefore interval analysis is not the selected method.
- Differential algebra offers an efficient approach to propagate the uncertainty through Taylor integration. However, this method alone does not provide much improvement compared to a traditional Monte Carlo with numerical integration. The computational cost of this process is still prohibitive.
- The polynomial chaos expansion is also a good candidate, although complex method due to the polynomial fitting. Moreover, the method by itself requires a high polynomial order to obtain an accurate solution, which results in a high computational cost.
- Analytically solving the FPE is not possible if perturbations are added to the dynamical model. Having a complete dynamical model is crucial to obtain an accurate solution and therefore this approach is not selected.
- Hybrid methods are the best option to accurately propagate the uncertainty in the satellite state while keeping a low computational cost. Two approaches are interesting, the hybrid DA-GMM and the hybrid PC-GMM. The hybrid PC-GMM is more complex and presents the drawback of providing a response surface for only one time step.

Based on the information gathered on each method and the characteristics in terms of accuracy, computational load and complexity, the hybrid DA-GMM method is selected. This method has already been tested in the field of satellite state uncertainty propagation. In Subsection 2.1.3 and Subsection 2.1.4 a review of the history of the Gaussian Mixture Model and Differential Algebra applied to the field of orbital mechanics is provided.

2.1.3. Overview of the Gaussian Mixture Model

The basic concept of the Gaussian mixture model is to divide the initial PDF into a weighted sum of Gaussian density functions as illustrated in Figure 2.2. There are several reasons to use this approach. First, by dividing the initial distribution into a large number of elements, the uncertainty (standard deviation) of each element is smaller. This allows to reduces the error in the propagation and allows to use lower-order techniques. Moreover, reducing the uncertainty of each GME allows to assume that the distribution remains constant

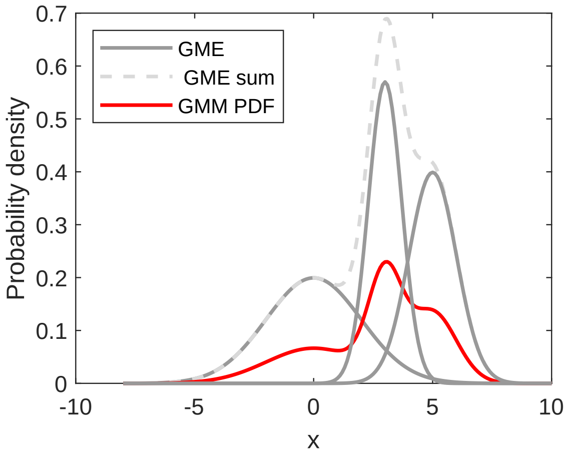


Figure 2.2: Example of one-dimensional GMM using three GMEs with $\mu = [0,3,5]$,$\sigma = [2,0.7,1]$ and equal weights

during propagation through nonlinear dynamics. Second, using a GMM allows to employ techniques that only propagate the mean and standard deviation of a distribution instead of using a Monte Carlo approach that requires a large number of samples. This allows to reduce the number of simulation runs and therefore the computational load.

As explained, the GMM needs to be combined with a propagation technique. Multiple approaches have been tried in the field of orbital uncertainty propagation, which have proved that the method's reliability. Terejanu et al. (2008) used this approach combined with linear theory to propagate the uncertainty in non-linear dynamic systems. Although the use of a GMM improves the accuracy of the propagation, the linear propagation using state transition matrix is a very simplified method and is still not valid for high accuracy applications. Continuing in this direction, Vishwajeet et al. (2014) combined the Gaussian mixture with the use of unscented transform for uncertainty propagation in perturbed two-body orbits. Although the method allowed to use a complex dynamical model including the J_2 and drag perturbation, the accuracy of the unscented transform is still too low for propagation. With this approach, the mean and standard deviation of the position PDF reached errors of 2 km after 5 hours of propagation. To improve the accuracy, the GMM needs to be combined with high accuracy propagation methods. Vittaldev and Russell (2016) used a polynomial chaos expansion to propagate the state uncertainty of objects in LEO and GEO. In this way, the accuracy of a Monte Carlo simulation with 10^8 samples could be reached with only 10^4 runs. However, this results in the instantaneous collision probability at TCA, but does not allow to integrate for long-term encounters. This issue does not appear when combining GMM with differential algebra. Sun et al. (2019) tested this approach to propagate the uncertainty of satellites in LEO, taking into account the J_2 perturbation. In this case, the Monte Carlo accuracy could be reached with 5,000 simulations, which correspond to the 5,000 GMEs that form the Gaussian mixture.

One of the main challenges that the Gaussian mixture model presents is the potential need to update the weights during the integration. In linear system the weights are constant, but for highly nonlinear systems it can be required to modify the weights to keep following the Fokker-Planck equation. This process can be very complex since it requires to perform an optimisation problem at each integration step for each GME. Terejanu et al. (2008) developed two approaches to apply this update to simple nonlinear systems. For a large number of elements, the computational load required to solve the optimisation problem would make the approach invalid for operational use. Horwood et al. (2011) found that in the case of orbital state uncertainty propagation, the weight update is not required if the correct set of coordinates is used (Kepler elements or modified equinoctial elements). The research proved that in this case the effect of weight update can even be counterproductive, reducing the accuracy.

2.1.4. Overview of Differential Algebra

Differential algebra is a computational technique to perform Taylor series integration in an efficient way. On its own, this technique does not show many advantages compared to traditional numerical integration methods. However, the mean and standard deviation of a variable distribution can be reconstructed from the Taylor expansion coefficients of the variable. This makes differential algebra a powerful tool to avoid multiple sample propagation when combined with the concept of Gaussian mixture models.

Differential algebra has been well tested in the field of orbital dynamics, showing excellent results. Di Lizia et al. (2008) applied this technique to integrate the trajectory in the case of Halo orbits in the circular restricted tree body problem and to a Mars aerocapture problem. In both cases, the trajectory was propagated using an eighth-order Taylor series expansion, which results in very high accuracy results. A benefit of this approach is that it allows to easily perform a sensitivity analysis on the dynamical systems under study. A similar research was conduced by Lunghi et al. (2018) for optimisation in atmospheric entry studies. Using the Taylor series expansion it is possible to obtain information about the sensitivity of the solution to variations in the optimisation variables.

Directly applied to the field of uncertainty propagation for collision risk estimation, Morselli et al. (2012) researched on advanced Monte Carlo methods coupled with differential algebra. Although the work only considered the unperturbed two body problem, it demonstrated that differential algebra can be used to propagate trajectory uncertainty in LEO and GEO. Valli et al. (2013) applies the same method to the perturbed two-body problem (including the spherical harmonic up to degree and order four and the drag acceleration). In this work, only uncertainty in the satellite position and velocity is considered. With this setup, the PDF is propagated achieving an error in the position mean and standard deviation after five hours of propagation < 1 cm for the sixth-order Taylor expansion. In this study the eight-order Taylor expansion is used as benchmark, proving that the method can be used for high-accuracy applications.

2.2. Collision probability calculation methods

The second step in the process of collision risk assessment is the calculation of the collision probability from the propagated state uncertainty. This step allows to quantify the risk and set a threshold for mitigation strategies to be applied. When no information on the state uncertainty was available, the only metric to assess the risk was the miss distance at time of closest approach. However, this magnitude did not provide full information on the risk. With the development of uncertainty propagation techniques, the collision probability can be calculated. This metric can still cause to underestimate the risk due to a phenomenon called *probability dilution*. When the initial covariance of one or both objects is too large, the final collision probability can become small due to the lack of knowledge of the satellite real position. However, collision probability is still the best metric to assess the risk of collision, and multiple techniques have been developed to calculate it.

The calculation process is divided into two categories: long-term and short-term encounters. The main differences are the time span and relative velocities of the encounter, which allow to apply special assumptions to avoid time integration. Essentially, short-term encounters deal with the computation of a two-dimensional collision probability, while long term encounters require time integration of the joint probability density function.

2.2.1. Short-term encounter

Short-term or high-velocity encounters occur in LEO, where the relative velocity between the target and chaser is very high (up to 15 km/s). This allows to apply certain assumptions to simplify the calculation of the collision probability.

A1. The target and chaser trajectories can be represented by straight lines with constant velocity during the encounter.

A2. The uncertainty in velocity can be ignored.

A3. The position uncertainty during the encounter is constant and equal to the value estimated at the conjunction.

A4. The uncertainties in the position of the target and chaser follow a Gaussian distribution and are uncorrelated.

A5. Both bodies are modelled as spherical.

Foster's method

The original method to compute the collision probability (P_c) using these assumptions was developed by Foster and Estes (1992). This method is still used and all modifications developed are intended to increase the computational efficiency, not to modify the results. For this approach, it is required to know the covariance matrix ($\boldsymbol{P}$) and the state (position and velocity) of the target and chaser at the TCA. From these data, the probability density function at TCA is obtained for each body and is merged into a combined PDF. To simplify the collision, it is assumed that both objects follow a straight trajectory (which defines a linear encounter geometry). This allows to define a conjunction plane, which contains the relative velocity vector and the target and chaser velocity vector. This plane contains all the uncertainty in position, and allows to reduce the problem to two dimensions. The remaining task is to project the uncertainty into the encounter plane and evaluate the two-dimensional integral of the PDF to calculate P_c. Following this approach, the probability is computed as:

$$P_c = \frac{1}{2\pi\,|\boldsymbol{P}^*|^{\frac{1}{2}}} \int_{-R}^{R} \int_{-\sqrt{R^2-y^2}}^{\sqrt{R^2-y^2}} \exp\left(-S^*\right) \mathrm{d}z\, \mathrm{d}y \tag{2.2}$$

where

$$\begin{aligned} S^* &= \left(\tilde{\boldsymbol{\rho}}^* - \bar{\boldsymbol{\rho}}_o^*\right)^T \boldsymbol{P}^{*-1} \left(\tilde{\boldsymbol{\rho}}^* - \bar{\boldsymbol{\rho}}_o^*\right)/2 \\ \tilde{\boldsymbol{\rho}}^* &= T^* C \tilde{\boldsymbol{\rho}}, \quad \bar{\boldsymbol{\rho}}_o^* = T^* C \bar{\boldsymbol{\rho}}_o \end{aligned} \tag{2.3}$$

$\boldsymbol{P}^*$ is the covariance matrix projected in the conjunction plane, R is the combined radius, C and T are coordinate transformation matrices, $\bar{\boldsymbol{\rho}}$, $\tilde{\boldsymbol{\rho}}$ are the nominal and perturbed miss vector between both bodies respectively.

The purpose of this method is to screen the collision probability of hundreds of thousands of space objects against each other, and therefore a very low computational load is required. This was not achieved with Foster's formulation, and therefore several mathematical re-definitions of problem have been conducted to increase the computational efficiency.

Patera's method

To increase the computational load of the method, Patera (2001) proposed an alternative way to evaluate Equation (2.2) by reducing the two-dimensional integral along the hard-body area to a one-dimensional integral along the hard-body perimeter. The hard-body is defined as the volume formed by the two colliding bodies. While the assumptions concerning the geometry of the encounter and the state uncertainty still apply to this method, it allows for an easier numerical implementation with a reduced computational effort. A small adjustment of the method was published by Patera (2005), which further reduced the computational time, becoming 20 times faster than Foster's method.

Besides the increased efficiency, this method presents a clear advantage with respect to equivalent formulations regarding the allowed object shapes. Since the one-dimensional integral is conducted along the perimeter of the body, it is not limited to a spherical shape anymore. In this case, the hard-body can be defined by a set of polygons that form a completely closed surface. This increases the fidelity of the model. The grid points defining the polygons are projected to the encounter plane to compute the collision probability. This improvement makes this method the baseline for the development of derived methodologies for short-term encounters. Moreover, this is the method of choice to compute the collision probability for short term encounters in ESA's software for Collision Risk Assessment and Avoidance Manoeuvre (CORAM), (Merz et al., 2017).

Alfano's method

Alfano (2005) developed an equivalent method to Patera's in terms of accuracy and computational load, again finding alternative ways to integrate Equation (2.2). However, this method assumes that he shape of the satellites can be defined as a sphere. In this case, the double integral is reduced to a single integral by making use of error functions (which in the end are defined as an integral, but are computationally more efficient to calculate). The resulting equation is numerically integrated to obtain the probability of collision.

Chan's method

Finally, Chan (1997) proposes a method to analytically compute the collision probability, with the assumptions defined so far. This is achieved by converting the two-dimensional Gaussian PDF into a one-dimensional Rician PDF. The main advantage of this formulation is that it increases the computational efficiency as it is 1,000 times faster than Patera's and Alfriend's method. However, the method presents several limitations in terms of the combined object size ($[1 \leq R \leq 100]$ m), and the standard deviation of the position error at TCA ($[1 \leq \sigma \leq 10]$ km). The method is included in the NASA Conjunction Risk Analysis (CARA) and STK software for the computation of two-dimensional collision risk. While this approach requires the object to be spherical, Chan develops an approximation that transforms each surface into an equivalent circle and performs the two-dimensional integration on that circle. This method is labelled the Method of Equivalent Cross-Section Area (MECSA) and was initially considered to be selected for this work, but finally an alternative technique was developed and implemented as described in Section 6.3.

2.2.2. Long-term encounter

The assumptions described so far cannot be applied to cases of a long-term or low-velocity encounter. This scenario occurs in higher orbits such as GEO and in cases where the relative velocity is low (e.g., if both satellites are in very similar orbits). Therefore, an alternative approach needs to be derived for these cases.

There are two main approaches that have been followed to tackle this problem. The first approach is based on the two-dimensional method described so far, usually Patera's method due to the flexibility provided by using a line integral. For these cases, the encounter geometry is defined by a set of volumes (cylinders, parallelepipeds, etc.) that can be divided into planar sections. In each time step, the two-dimensional probability is computed for each section, assuming a linear motion and following Equation (2.2). Moreover, a one-dimensional probability is computed along the relative velocity vector on the volume. For each volume, the collision probability is obtained by multiplying the values obtained for these two probabilities. Finally, the total collision probability is obtained by adding up the probabilities for the individual volumes. A common limitation of these methods is that both objects must have a non-negligible relative velocity. Considering that these methods are developed for long-term, slow encounters, there are cases where this condition will not be satisfied. The second approach consists of computing the three-dimensional integral of the collision probability over time. This is done by defining the variable of collision probability rate, and calculating the flow of this variable over the surface of hard-body volume during the encounter time interval. Finally, the collision probability rate is integrated over time.

Patera's method

Patera (2003) proposed an approach using contour integration, which did not require assumptions on the trajectory of the target and chaser. The method follows the first approach and consists of transforming the problem to a different frame in which the covariance matrix is symmetric in three dimensions. With this methodology, the collision probability was calculated with an error of 6% for satellites in geo-synchronous orbits. The method was been included in ESA's software CORAM, for the calculation of collision probability in long-term encounters. Since the two-dimensional collision probability in the encounter plane is calculated using Patera's method with the line integral along the perimeter, in principle any shape composed by parallelepipeds can be modelled. However, for simplicity a spherical body is assumed.

Chan's method

Based on the work for two-dimensional collision probability developed by Chan (1997), the method was extended to long-term encounters (Chan, 2004). Following the first approach for this type of collisions, the two dimensional probability is integrated for a section of the collision volume and then integrated along the volume. It is chosen to evolve the size of the volume with the change in the uncertainty ellipsoid. As a result, the integration volume is set to be bounded by an envelope of a family of ellipsoids. Also in this case, Chan uses the method of equivalent cross-section area, applied to the two-dimensional integration in each "slice" within the integration volume.

Alfano's methods

Salvatore Alfano developed three different methods to compute the probability of collision for long term encounters by performing two-dimensional integration over the sections forming the volume and then one-dimensional integration over the span of the volume (Alfano, 2006; Alfano, 2007). For the encounter volume, three different options are proposed. The first method consists of adjoining tubes (cylinders) in Cartesian space. The second method considers adjoining parallelepipeds to model the encounter volume. Finally, a method that discretises the encounter geometry through volume elements (volumes) is presented. The firsts two methods introduce an error due to the gaps that are created between the successive tubes/parallelepipeds. This error can be reduced by incriminating the number of elements, which affects the computational load. The third method requires the definition of grid of voxels (or unit cubes), which becomes more computationally intense than the previous two. According to Alfano (2006), even with a reduced number of voxels the computation runs slower than real time, and therefore the method is only recommended for reference case determination, similarly to the Monte Carlo methodology. The methods are included in the commercial software STK to compute the probability of collision in long-term encounters.

McKinley's method

McKinley (2006) proposed a variation of the cylinder approach that bypasses the problem of gaps between the different volumes by truncating the cylinders at their intersection. The three-dimensional integral of the Gaussian probability over the cylinder is calculated taking into account the previously computed integration limits for the truncated body. The collision probability is calculated for each volume and added up to obtain the total value. This method is currently included in the NASA-CARA software and is the method selected for long term encounters. The results provided by this method have an error within an order of magnitude of the Monte Carlo results. However, this still has large room for improvement.

Coppola's method

Finally, the only method developed up to date that follows the second approach is the time integration of collision probability rate presented by Coppola and McAdams (2012). This method is the one selected by NASA-CARA to calculate the collision probability for long-term encounters. The method includes both the position and velocity uncertainties and allows for any encounter geometry. To solve the probability of collision (a 12-dimensional integral), nine dimensions can be integrated analytically. The remaining three are solved numerically through a two-dimensional integral over a sphere and a one-dimensional integral over time. To do so, it is assumed that the probability distribution of the target and chaser are independent. Moreover, the method assumes that the probability is Gaussian and that the encounter geometry is a sphere.

2.3. Reference mission

In this section, the reference scenarios considered as a baseline for this work are discussed and motivated. From the review of collision probability calculation methods, it is observed that the problem can be divided into short-term and long-term encounters that require different assumptions. While the short-term encounter has been deeply studied and computationally optimised, the long-term encounter remains a challenge. The methods developed up to date are either computationally expensive, or not applicable to all cases. Therefore, this work aims to be valid for both long-term and short-term encounters.

Besides the encounter geometry, the allowed mission scenarios are also constrained by the orbital altitude, which affects the dynamical model required for the propagation. Based on the density of space objects at different altitudes, it is observed that LEO is the most crowded region and therefore presents the highest risk in the present and in the foreseen future. Accurate collision detection is fundamental in this region and will become increasingly crucial with the deployment of satellite constellations. Therefore, the reference mission will be placed in LEO, although the collision detection method will be extended and tested for higher altitudes. Not all altitude bands in LEO pose the same risk to satellites. As observed in Figure 2.3, the peak of concentration lies around 800 km altitude and it has increased significantly between the two dates depicted. This altitude band corresponds to the region where the Cosmos-2251/Iridium-33 collision took place in February 2009, creating more than 10^6 pieces of debris larger than 1 cm. Due to this event, this altitude region has become of high interest to perform collision risk detection and it is selected as a baseline for this study.

A second area of interest is found at an altitude of 400-450 km, due to the presence of the International Space Station(ISS). Until now, collision risk for the ISS has been overestimated, to ensure the safety of the crew and the vehicle. However, nowadays there are multiple events that could lead to overestimation posing a larger threat. For example, in the case that multiple debris are detected to potentially collide with the ISS, it is required to know which of these objects pose a higher risk to perform the best collision avoidance manoeuvre. The ISS orbit is therefore selected as the second reference mission.

Both of these reference missions have very low eccentricities and low altitudes. A third reference scenario is created to include highest orbital altitudes and verify that the method can be used for orbits with high eccentricity. The orbital parameters defining these orbits are presented in Table 2.1. The orbital inclination of these orbits are also selected following the same reasoning. For the LEO case, an inclination of 60 degrees is chosen, which represents the most crowded orbital plane at this altitude. For the ISS orbit, the inclination coincides with the real inclination of the ISS (51.64 deg) and for the eccentric orbit, a retrograde trajectory with 98deg inclination is selected to test all possible scenarios.

It must be noted that these reference missions are used to develop and test the uncertainty propagation algorithm. Moreover, real verification and validation scenarios are simulated and analysed, and these will be presented separately in Chapter 9. For this purpose, three real scenarios are considered. First, the Cosmos-2251/Iridium-33 collision event is simulated to compute the risk of collision with the available tracking data.

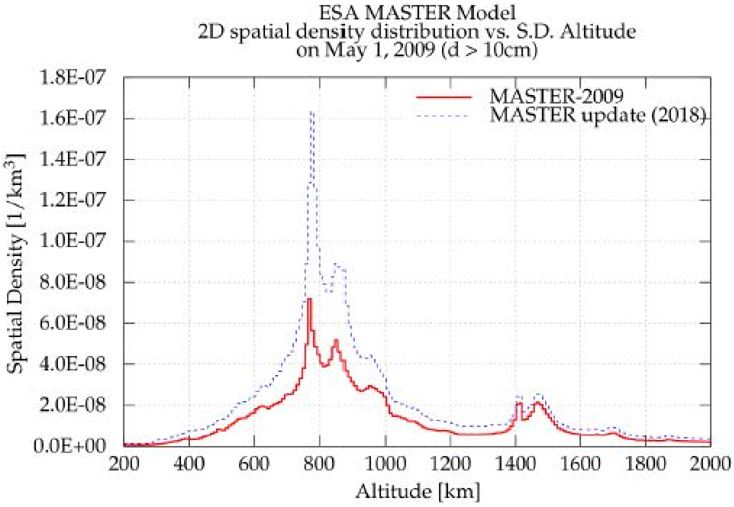


Figure 2.3: Spatial density of space objects >10 cm in LEO on 2009 and 2018 (Horstmann et al., 2018)

Table 2.1: Orbital parameters and initial velocity of reference missions.

Case	a [km]	h_p [km]	h_a [km]	e [-]	i [deg]	v [km/s]
LEO	7178.137	800	800	0	60	7.45
ISS	6800.137	419.416	424.58	0.00038	51.64	7.65
Eccentric	9378.137	655.46	5344.53	0.25	98	8.41

Second, the close encounter between the Chinese rocket body Chang Zheng-4C and the Cosmos-2004 satellite that occurred in 2020 is simulated and compared to the predictions performed by LeoLabs [1]. Finally, after the anti-satellite missile test performed in November 2021 that destroyed the satellite Cosmos-1408, a screening of potential approaches and collision probability estimation with the ISS is performed.

2.4. Reference vehicles

The reference vehicles are selected according to the mission scenarios that are studied as part of the validation. For the general application of the DA-GMM method of collision probability calculation with the single-sphere assumption, the only parameters of the vehicle that are required are: ballistic coefficient (defined by area, mass, drag coefficient) and the radius of the sphere fitting the body. For the case of the ISS, the multi-sphere model is developed in Section 6.3, which takes into account the full three-dimensional shape of the vehicle. For this, a three-dimensional model of the satellite is required. The following objects are involved in the collision scenarios studied. From the satellites introduced, the full geometrical shape is only considered for the ISS, due to its more elongated shape, and the need for an accurate collision probability estimation regarding this inhabited vehicle.

Cosmos-2251

Cosmos-2251 was a Russian military communications satellite launched in 1993 and functioning until 1995 although the nominal mission duration was of five years. By the time of the collision with Iridium-33, the satellite was defunct and there was no option to perform any collision avoidance manoeuvres. Although at the time of satellite end-of-life there were no international regulations forcing to remove the satellite from the congested orbit, this is currently a protected region and satellites need to be de-orbited or put in a graveyard orbit in less than 25 years from nominal end-of-life. The satellite had a mass of 900 kg and a frontal area of 4 m^2 is assumed.

Iridium-33

Iridium-33 was part of the Iridium constellation consisting of 66 satellites, 11 in each orbital plane, to provide global L-band mobile communication. The satellite was still controllable at the time of the collision with

[1] `https://www.leolabs.space/`

Cosmos-2251 in 2009. The Iridium satellite operator had the power to perform a Collision Avoidance Manoeuvre, but it was concluded that the collision risk was low and a manoeuvre was not required. The satellite had a mass of 689 kg, and a frontal area of 2.2 m^2 is assumed.

Cosmos-2004
Cosmos-2004 is a Russian military navigation satellite currently inactive. The satellite was involved in a close encounter event with the Chinese rocket module Chang Zheng-4C in October 2020, and will be used as a test case for the method presented in this work. The spacecraft has a mass of 825 kg and a frontal area of 13 m^2 is assumed. The satellite has a large boom of 18 m which must be taken into account in the calculation of collision probability.

Chang Zheng-4C
The Chang Zheng 4C (also known as Long March 4C) is a Chinese launch vehicle consisting of three stages. The upper stage is currently orbiting Earth with a perigee and apogee altitudes of 977 km and 1210 km respectively. With a mass of 50,000 kg and a height of 14.79 m, the consequences of a head-on collision with another space object would have been disastrous. Fortunately, this event did not occur but it provides an interesting case for study. Moreover, in this case the object is very elongated, and the effect of changing the encounter geometry based on the orientation of the object can be evaluated.

International Space Station
The International space Station, launched in 1998, has been the centre of scientific research in micro-gravity and demonstration for human spaceflight activities. From its deployment, the ISS has been home of 66 expeditions with of a total of 251 crew members from 19 countries, including the visit of the first tourist, Yusaku Maezawa who arrived at the station on December 8, 2021. Being an inhabited vehicle, collision risk estimation is crucial and special guidelines and offices have been created to perform this task. However, these often rely on overestimation, which can lead to a lack of knowledge of the real risk of collision. This can be critical if the encounter occurs during a docking or extra-vehicular operation, or if multiple debris simultaneously threaten the vehicle. The ISS has a span of 109 m and weights 444,615 kg. In this work, the complete shape of the ISS is modelled and included in the collision probability calculation software by using a multi-sphere mesh to approximate the surface integration of the probability flow.

2.5. Mission and system requirements

The following mission and system requirements are specified as a guideline that is followed through the development of the DA-GMM method. First, the mission requirements setting the types of inputs and the expected outputs of the method are defined.

MISS-1 The error in P_c calculation of the DA-GMM method with respect to the Monte Carlo simulation must be $< 5\%$.

MISS-2 The method shall detect a potential collision with a lead time of at least two days with the specified accuracy.

MISS-3 The method shall be applicable to all objects > 5 cm at an orbital altitude range of 400-3000 km.

MISS-4 The method shall be applicable to objects in any orbital plane (i.e., inclination range $[0° \;\; 180°]$).

MISS-5 Simplifications in the force and environment models shall have an acceleration error $< 10^{-7}$ m/s^2.

MISS-6 The method shall correctly predict/discard past events of potential collision such as the Iridium-33/Cosmos-2251 collision or the CZ4C/Cosmos-2004 close approach.

MISS-7 The method shall predict the time span of the encounter.

MISS-8 The method shall be valid for all types of encounter geometries, including long-term and short-term encounters.

The simulators is developed in this work to apply the DA-GMM method shall obey the following system requirements:

SYS-1 The method's computational load must allow to compute P_c within the alert time before the last station passage of the satellite under analysis.

SYS-2 The uncertainty model shall at least contain the uncertainty of the target and chaser state.

SYS-3 The possibility to add uncertainty in the atmospheric density and ballistic coefficient shall be included in the simulator.

SYS-4 The rotational dynamics shall be neglected.

SYS-5 For each body, the hard body radius shall be an input to the model. For the ISS, a simplified geometrical model shall be provided.

SYS-6 The integration error for numerical and Taylor series integration shall be at least three orders of magnitude lower than the model acceleration threshold (10^{-10} m/s^2).

SYS-7 The method shall provide as a result the P_c and the time history of the target and chaser state PDF.

Finally, this work relies highly on statistics, and therefore must satisfy the following statistical requirements:

STAT-1 The L_2 error for the GMM splitting shall be $< 10^{-7}$.

STAT-2 For the P_c calculation with time integration, the integration time shall contain the period when the miss distance is < 30 km.

STAT-3 The number of GMEs shall allow the propagated GME to pass the specified Gaussian hypothesis tests.

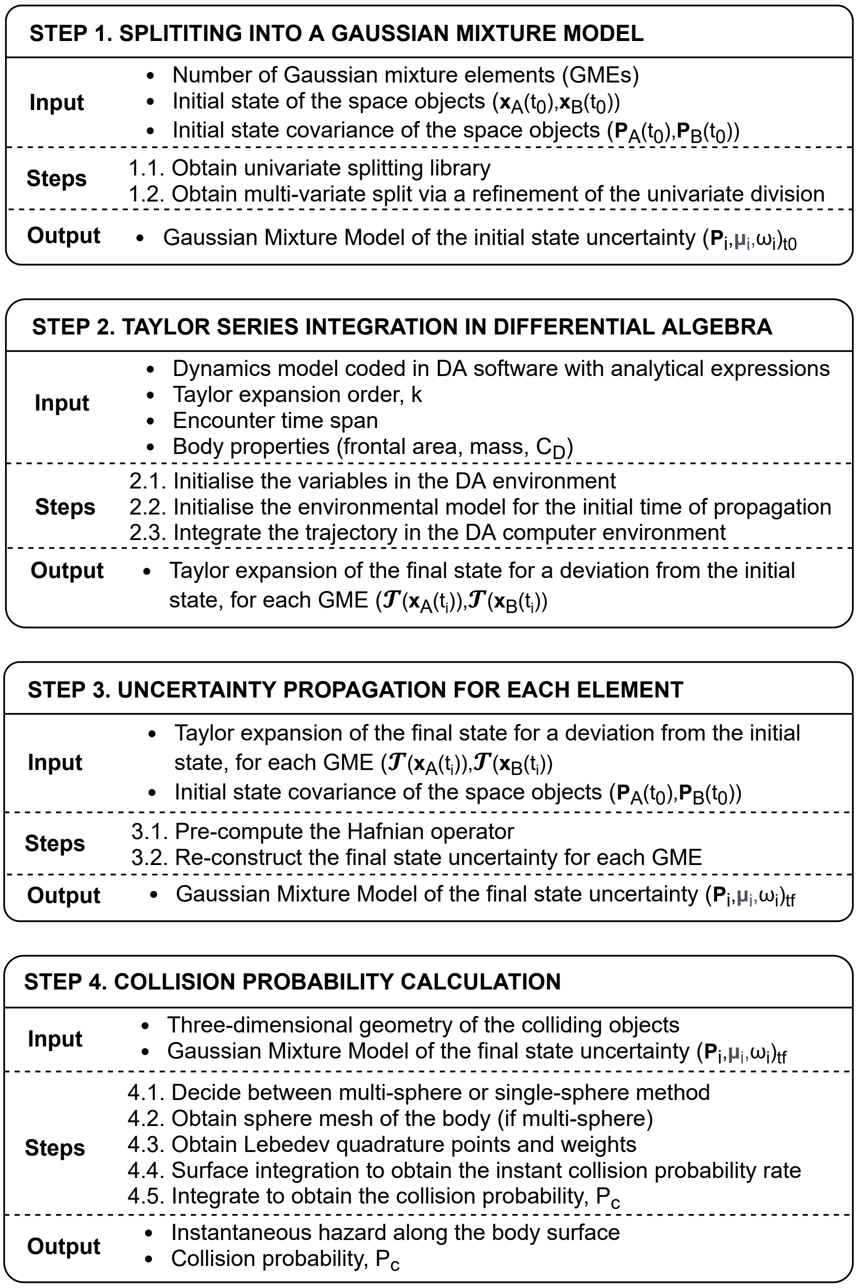
2.6. Methodology road-map

The methodology followed to develop the DA-GMM method for collision probability can become quite cumbersome and mathematically extensive. Despite the effort made to complement the mathematical concepts with visual examples, the procedure can still be complex to understand. To assist the reader through the immersion in the following chapters, a road-map is added. Along the explanation of the methodology, the reader can refer back to the road-map to understand the bigger-picture.

The road-map is divided into four steps, which can be categorised into two different phases of the collision probability calculation process. The first three steps belong to the uncertainty propagation process, to obtain the probability distribution of the satellites position and velocity at the time of the encounter. The detailed mathematical explanation of this phase is provided in Chapter 5. To develop these method, a set of environmental and force models are required to be coded in a differential algebra software, using analytical expressions only. As observed, this is an input to Step 2. The theory and choices related to the environmental models are found in Chapter 3, and are introduced before diving into the methodology of the DA-GMM method. Moreover, to develop the uncertainty propagation method it is required to have some insight on the types of uncertainty distributions and the specific values that are expected for the satellites and missions under study. This is also provided as complementary information to the methodology in Chapter 4.

The last step corresponds to the collision probability calculation phase, which is detailed in Chapter 6. In this work, two different models are developed to compute this probability. First, a baseline model that considers the shape of the satellite to be a sphere, and second, an extended model that allows to consider any geometry. The choice of model does not have a large effect on the methodology. The full multi-sphere DA-GMM methodology, summarised in these four steps, is therefore introduced in Chapter 5 and Chapter 6.

In Chapter 7, the methodology presented in this road-map is transformed into a software architecture diagram, that presents all the processes and relationships involved in the computation of the collision probability. Although both diagrams share a same background, the software architecture includes details that are defined later in the document, and is more related to the underlying mathematics.



STEP 1. SPLITITING INTO A GAUSSIAN MIXTURE MODEL

Input	• Number of Gaussian mixture elements (GMEs) • Initial state of the space objects ($\mathbf{x}_A(t_0)$,$\mathbf{x}_B(t_0)$) • Initial state covariance of the space objects ($\mathbf{P}_A(t_0)$,$\mathbf{P}_B(t_0)$)
Steps	1.1. Obtain univariate splitting library 1.2. Obtain multi-variate split via a refinement of the univariate division
Output	• Gaussian Mixture Model of the initial state uncertainty $(\mathbf{P}_i,\mu_i,\omega_i)_{t0}$

STEP 2. TAYLOR SERIES INTEGRATION IN DIFFERENTIAL ALGEBRA

Input	• Dynamics model coded in DA software with analytical expressions • Taylor expansion order, k • Encounter time span • Body properties (frontal area, mass, C_D)
Steps	2.1. Initialise the variables in the DA environment 2.2. Initialise the environmental model for the initial time of propagation 2.3. Integrate the trajectory in the DA computer environment
Output	• Taylor expansion of the final state for a deviation from the initial state, for each GME ($\mathcal{T}(\mathbf{x}_A(t_i))$,$\mathcal{T}(\mathbf{x}_B(t_i))$)

STEP 3. UNCERTAINTY PROPAGATION FOR EACH ELEMENT

Input	• Taylor expansion of the final state for a deviation from the initial state, for each GME ($\mathcal{T}(\mathbf{x}_A(t_i))$,$\mathcal{T}(\mathbf{x}_B(t_i))$) • Initial state covariance of the space objects ($\mathbf{P}_A(t_0)$,$\mathbf{P}_B(t_0)$)
Steps	3.1. Pre-compute the Hafnian operator 3.2. Re-construct the final state uncertainty for each GME
Output	• Gaussian Mixture Model of the final state uncertainty $(\mathbf{P}_i,\mu_i,\omega_i)_{tf}$

STEP 4. COLLISION PROBABILITY CALCULATION

Input	• Three-dimensional geometry of the colliding objects • Gaussian Mixture Model of the final state uncertainty $(\mathbf{P}_i,\mu_i,\omega_i)_{tf}$
Steps	4.1. Decide between multi-sphere or single-sphere method 4.2. Obtain sphere mesh of the body (if multi-sphere) 4.3. Obtain Lebedev quadrature points and weights 4.4. Surface integration to obtain the instant collision probability rate 4.5. Integrate to obtain the collision probability, P_c
Output	• Instantaneous hazard along the body surface • Collision probability, P_c

Figure 2.4: Road-map to follow the methodology of the multi-sphere DA-GMM method developed in this work.

3

Flight dynamics and environmental model

This chapter introduces the resources required to model the physical environment of the satellite and propagate its trajectory. To correctly identify the state of the satellite in the trajectory and during the conjunction, a set of reference frames are presented in Section 3.1. To operate between different reference systems, frame transformations are developed in Section 3.2. Section 3.3 presents the different sets of elements that can be used to model the state of a body, and their advantages and drawbacks. These include Cartesian coordinates, Kepler elements, modified equinoctial elements (MEE) and unified state models (USM). In Section 3.4, the equations of motion governing the dynamics in the different state models are introduced. Finally, Section 3.6 summarises the perturbation forces that act on the satellite for the reference mission presented in Section 2.3 and discusses their inclusion in the physical model.

3.1. Reference frames

This section summarises the reference frames required for the description of vector states in the context of space debris collision probability. For each reference frame, the origin and the directions of each axis are given and their application is described. Unless specified otherwise, the definition of the reference frames is based on the work by Mooij (2016).

Earth Centred Inertial Reference Frame (*I*-Frame)

This reference frame is in reality pseudo-inertial due to the movement of the central body (Earth in this case) about the centre of mass of the Solar System. However, this effect is negligible considering the proximity of the objects to the central body and the short periods of time. For Earth, the J2000 frame is used, with the X-axis pointing towards the Vernal Equinox at 12:00 UTC on 1 January 2000. The orientation of the prime meridian of Greenwich with respect to the J2000 frame is denoted as Θ at time t.This reference frame is used to describe and propagate in time the position and velocity of the satellites and space debris in LEO.

Origin	Centre of mass of the Earth
X_I axis	Vernal equinox: intersection of ecliptic and equatorial plane
Y_I axis	Completes the right-handed system
Z_I axis	Aligned with the rotation axis of the Earth

Earth Fixed Rotating Reference Frame (*R*-Frame)

This frame rotates along with the Earth, and is therefore fixed to its surface. Once a day the frame coincides with the Earth Centred Inertial reference frame. The position of ground stations to track satellites and space debris is specified in this frame.

Origin	Centre of mass of the Earth
X_R axis	Intersects the equator at zero longitude
Y_R axis	Completes the right-handed system
Z_R axis	Aligned with the rotation axis of the Earth

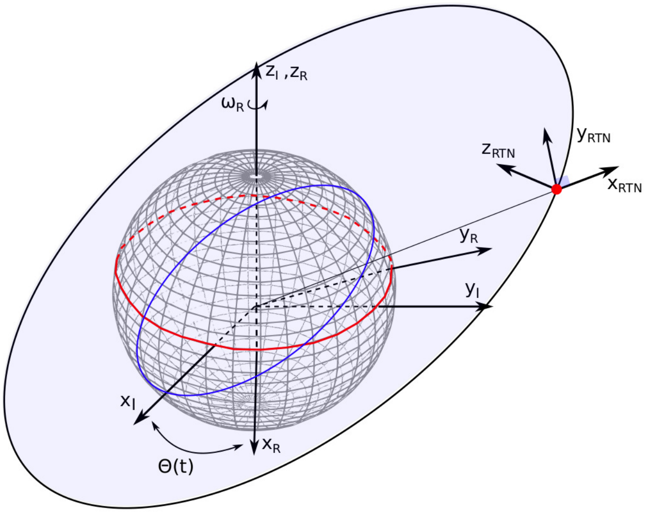


Figure 3.1: Illustration of the Earth Centred Inertial, Earth Fixed Rotating and RTN Reference frames. ω_R is the Earth's angular velocity

Radial-Transverse-Normal Reference Frame (*RTN*-Frame)

This local frame centred in the spacecraft is used to represent the state uncertainties of the spacecraft and debris and the encounter geometry as described by SpaceTrack (2020). This reference frame is also referred to as Radial-Intrack-Crosstrack (RIC) and UVW in the literature. Figure 3.1 illustrates the relation between the Earth Centred Inertial, Earth Fixed Rotating and RTN reference frames. It should be noted that the $X-Y_{RTN}$ plane coincides with the orbital plane.

Origin	Centre of mass of the spacecraft
X_{RTN} axis	Unit vector of the satellite position from the centre of Earth
Y_{RTN} axis	Completes the right-handed system
Z_{RTN} axis	Unit vector of the satellite's angular momentum (normal to orbital plane)

Body Reference Frame (*B*-Frame)

This reference frame is used to represent the geometry of the spacecraft and allows to define the attitude angles.

Origin	Centre of mass of the spacecraft
X_B axis	Lies in the spacecraft plane of (longitudinal) symmetry, in the positive forward direction
Y_B axis	Completes the right-handed system
Z_B axis	Lies in the spacecraft plane of (longitudinal) symmetry, in the positive downward direction

Trajectory Reference Frame (*T*-Frame)

The trajectory reference frame is included to orient the spacecraft's velocity with respect to the rotating Earth, which is essential to calculate the collision probability of the spacecraft with the incoming debris.

Origin	Centre of mass of the spacecraft
X_T axis	Points in the direction of the spacecraft velocity relative to the Earth
Y_T axis	Completes the right-handed system
Z_T axis	Lies in the vertical plane, pointing downwards to Earth

Conjunction Reference Frame (*C*-Frame)

This reference frame was introduced by Foster and Estes (1992) to simplify the calculation of collision probability for fast encounters by confining all the uncertainty in a plane, called the conjunction plane. This frame was continued to be used in subsequent formulations of the collision probability calculation such as the methods of Akella and Alfriend (2000) and Patera (2001). Considering the spacecraft and debris velocity vectors respectively, $\boldsymbol{v_s}$ and $\boldsymbol{v_d}$, and the relative velocity vector $\boldsymbol{v_r} = \boldsymbol{v_d} - \boldsymbol{v_s}$, the new reference frame is defined as:

$$\hat{\boldsymbol{i}} = \frac{\boldsymbol{v}_r}{|\boldsymbol{v}_r|}, \quad \hat{\boldsymbol{j}} = \frac{\boldsymbol{v}_d \times \boldsymbol{v}_s}{|\boldsymbol{v}_d \times \boldsymbol{v}_s|}, \quad \hat{\boldsymbol{k}} = \hat{\boldsymbol{i}} \times \hat{\boldsymbol{j}} \tag{3.1}$$

In this formulation, all the uncertainty of the miss distance vector is restricted to the $Y - Z_c$ plane.

Origin	Centre of mass of the spacecraft
X_C axis	Points the relative velocity between the spacecraft and the space debris
Y_C axis	Points to the normal vector of the plane containing the velocity of the spacecraft and the debris
Z_C axis	Completes the right-handed system

3.2. Frame transformations

Working with different reference frames requires to transform the state variables from one frame to another. This transformation is conducted through the rotation matrix. The transformation matrix from frame A to frame B is denoted by: $\boldsymbol{C_{B,A}}$. There are multiple ways to define the rotation between axis. The most common which are treated here are unit axis rotations about Euler angles and directional cosine matrices. The basic transformation about a single axis for an arbitrary angle α is:

$$\mathbf{C}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \quad \mathbf{C}_y(\alpha) = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \quad \mathbf{C}_z(\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

for rotations about the X, Y and Z axis respectively. Any rotation from frame A to B can be decomposed into a series of unit axis rotations. The resulting transformation matrix is obtained by the multiplication of the individual rotations in reversed order. Considering the case of a negative rotation of angles α_1, α_2 and a positive rotation of α_3 about the Y-axis, X-axis and Z-axis respectively, the resulting transformation matrix is:

$$\mathbf{C_{B,A}} = \mathbf{C}_z(\alpha_3)\,\mathbf{C}_x(-\alpha_2)\,\mathbf{C}_y(-\alpha_1)$$

And the inverse transformation is given by:

$$\mathbf{C_{A,B}} = \mathbf{C}_{\mathbf{B,A}}^{-1} = \mathbf{C}_{\mathbf{B,A}}^{T} = \mathbf{C}_y(\alpha_1)\,\mathbf{C}_x(\alpha_2)\,\mathbf{C}_z(-\alpha_3) \tag{3.3}$$

When the unit vectors of a reference frame can be expressed as a function of the unit vectors of the final frame, the rotation matrix can be directly computed by means of a directional cosine matrix. This is the case of the transformation to the conjunction reference frame. Consider a reference frame defined by the unit vectors $[\hat{\boldsymbol{i}}_1, \hat{\boldsymbol{j}}_1, \hat{\boldsymbol{k}}_1]$ and a second frame defined by the set of unit vectors $[\hat{\boldsymbol{i}}_2, \hat{\boldsymbol{j}}_2, \hat{\boldsymbol{k}}_2]$. Both reference frames can be related by:

$$\hat{\boldsymbol{i}}_1 = A_{11}\hat{\boldsymbol{i}}_2 + A_{12}\hat{\boldsymbol{j}}_2 + A_{13}\hat{\boldsymbol{k}}_2 \tag{3.4}$$

$$\hat{\boldsymbol{j}}_1 = A_{21}\hat{\boldsymbol{i}}_2 + A_{22}\hat{\boldsymbol{j}}_2 + A_{23}\hat{\boldsymbol{k}}_2 \tag{3.5}$$

$$\hat{\boldsymbol{k}}_1 = A_{31}\hat{\boldsymbol{i}}_2 + A_{32}\hat{\boldsymbol{j}}_2 + A_{33}\hat{\boldsymbol{k}}_2 \tag{3.6}$$

where A_{ij} is the cosine of the angle α_{ij} between axis i in frame 1 and axis j in frame 2. These components are called directional cosines. The transformation matrix between these frames obtained by reconstructing the directional cosines in a single matrix.

3.2.1. Reference angles

To conduct these transformations, the following set of reference angles is required. Table 3.1 introduces the reference angles required for the derivation of the frame transformation matrices. Four categories are identified. First, the longitude and latitude angles are used to identify the location of the satellite with respect to the rotating Earth. Second, the orbital angles u, i and Ω determine the location of the satellite in the orbital plane. Third, the aerodynamic angles β, σ and α orient the satellite with respect to the velocity vector. Finally, the attitude angles γ and χ define the orientation of the satellite with respect to the local horizontal plane. The definitions are based on the work by Mooij (2016).

Table 3.1: Definition of reference angles required for frame transformations

Symbol	Name	Range [deg]	Definition
τ	Longitude	[0 360)	Angle between the projection of the secondary body (e.g., spacecraft) position vector on the *XY* plane and the X-axis of the R-frame
δ	Latitude	[-90 +90]	Angle between the spacecraft position vector and the XY plane in the R-frame
u	Argument of latitude	[0 360)	Angle on the orbital plane between the ascending node and the secondary body
i	Inclination	[0 180]	Angle between the *XY* plane in the I-frame and the orbital plane
Ω	RAAN	[0 360)	Angle between the Vernal Equinox and the ascending node (intersection of the orbit with the Earth's equatorial plane)
γ	Flight path angle	[-90 +90]	Angle formed between the velocity vector and the local horizontal plane
χ	Heading angle	[-180 +180)	Angle of the projection of the velocity vector in the local horizontal plane from the local north
α	Angle of attack	[-180 +180)	Angle between the *XY* plane in the T-frame and the longitudinal axis of the spacecraft
β	Sideslip angle	[-90 +90]	Angle between the velocity vector and the projection of the spacecraft longitudinal axis in the *XY* plane of the T-frame
σ	Bank angle	[-180 +180)	Angle between the spacecraft longitudinal symmetry plane and the Z-axis of the T-frame

3.2.2. Transformation matrices

To work with different frames, the rotation matrices shall be defined in a way that the state can be transformed from any initial frame to all the frames considered. In this section, five rotation matrices are introduced. Further frame transformations can be achieved by combining and transposing the matrices given.

Earth Fixed Rotating to Earth Centred Inertial Reference Frame ($\mathbf{C}_{I,R}$)

The Earth fixed and the Earth centred inertial reference frames are chosen to coincide at time zero ($t_0 = 0$). For each Earth revolution, the R-frame completes a full rotation about the Z-axis. At a given time from epoch (t), the rotated angle equals $\omega_E \cdot t$ where ω_E is the angular velocity of the Earth. The transformation matrix between these frames is:

$$\mathbf{C_{I,R}} = \mathbf{C}_3\left(-\omega_E \cdot t\right) \tag{3.7}$$

Earth Centred Inertial to Radial-Transverse-Normal Reference Frame ($\mathbf{C}_{RTN,I}$)

The *RTN*-frame defines the state with respect to the position of the satellite in the orbital plane. The transformation from the inertial to the *RTN*-frame uses the angles which define the orbital plane and the position within the orbit. It consists of three unit axis rotations. First, a rotation about the $Z-$axis to align the $X-$axis with the ascending node. Then, a rotation about the X-axis to align with the orbital plane. Finally, a rotation about the $Z-$axis to align the $X-$axis with the position vector. With the angles defined in Table 3.1 and illustrated in Figure 3.3, the transformation matrix is given by:

$$\mathbf{C_{I,R}} = \mathbf{C}_3\left(u\right)\mathbf{C}_1\left(i\right)\mathbf{C}_3\left(\Omega\right) \tag{3.8}$$

Body to Earth Fixed Rotating Reference Frame ($\mathbf{C}_{R,B}$)

The transformation from the $B-$frame to the R-frame uses the attitude angles defined, χ and γ and the aerodynamic angles α, β and σ. The derivation of this rotation matrix can be consulted in Mooij (2016).

$$\mathbf{C_{R,B}} = \mathbf{C}_3(-\tau)\mathbf{C}_2\left(\frac{\pi}{2}+\delta\right)\mathbf{C}_3\left(-\chi\right)\mathbf{C}_2\left(-\gamma\right)\mathbf{C}_1\left(\sigma\right)\mathbf{C}_3\left(\beta\right)\mathbf{C}_2\left(-\alpha\right) \tag{3.9}$$

Body to Trajectory Reference Frame ($\mathbf{C}_{T,B}$)
For this transformation, the aerodynamics angles allow to orient the vehicle with respect to the body velocity. The transformation matrix is:

$$\mathbf{C_{T,B}} = \mathbf{C}_3(\beta)\mathbf{C}_2(-\alpha)\,\mathbf{C}_1(\sigma) \tag{3.10}$$

Conjunction to Inertial Reference Frame ($\mathbf{C}_{I,C}$)
The definition of the conjunction reference frame is given in terms of the satellite and debris inertial velocities. Let us consider the following three auxiliary vector definitions.

$$\boldsymbol{v_1} = \frac{\boldsymbol{v}_r}{|\boldsymbol{v}_r|} \qquad \boldsymbol{v_2} = \frac{\boldsymbol{v}_d \times \boldsymbol{v}_s}{|\boldsymbol{v}_d \times \boldsymbol{v}_s|} \qquad \boldsymbol{v_3} = \boldsymbol{v_1} \times \boldsymbol{v_2}. \tag{3.11}$$

where $\boldsymbol{v}_s$, $\boldsymbol{v}_d$ are the satellite and debris velocity respectively in the inertial frame and $\boldsymbol{v}_r = \boldsymbol{v}_d - \boldsymbol{v}_s$ is the relative velocity.

The encounter frame, defined by the unit vectors $[\hat{\boldsymbol{i}}_{\boldsymbol{C}}, \hat{\boldsymbol{j}}_{\boldsymbol{C}}, \hat{\boldsymbol{k}}_{\boldsymbol{C}}]$ is formed by:

$$\hat{\boldsymbol{i}}_{\boldsymbol{C}} = \boldsymbol{v_1} = v_{1_x}\hat{\boldsymbol{i}}_{\boldsymbol{I}} + v_{1_y}\hat{\boldsymbol{j}}_{\boldsymbol{I}} + v_{1_z}\hat{\boldsymbol{k}}_{\boldsymbol{I}} \tag{3.12}$$

$$\hat{\boldsymbol{j}}_{\boldsymbol{C}} = \boldsymbol{v_2} = v_{2_x}\hat{\boldsymbol{i}}_{\boldsymbol{I}} + v_{2_y}\hat{\boldsymbol{j}}_{\boldsymbol{I}} + v_{2_z}\hat{\boldsymbol{k}}_{\boldsymbol{I}} \tag{3.13}$$

$$\hat{\boldsymbol{k}}_{\boldsymbol{C}} = \boldsymbol{v_3} = v_{3_x}\hat{\boldsymbol{i}}_{\boldsymbol{I}} + v_{3_y}\hat{\boldsymbol{j}}_{\boldsymbol{I}} + v_{3_z}\hat{\boldsymbol{k}}_{\boldsymbol{I}} \tag{3.14}$$

Which, following the procedure of the directional cosine allow to reconstruct the transformation matrix.

3.3. State models

The reference frame and transformation matrices defined provide the framework to describe the position and velocity of the satellite. However, different state models can be considered to describe these variables. The choice of state model has an effect on the accuracy of the propagation as well as the computational time, since the equations of motion take a different shape for different models. The most commonly used state models are the Cartesian coordinate system and Kepler elements due to their simplicity, physical meaning and abundance of environmental models expressed in these frames. However, these models are not always the most optimum representation of the trajectory, since they can introduce numerical errors due to large variations in the state and singularities. Alternative state models have been developed that solve this issues, at the expense of losing physical meaning and introducing more complexity. Some examples of alternative state models are the modified equinoctial elements (Hintz, 2008), and the unified state models (Altman, 1972; Vittaldev et al., 2012). Although these methods can provide multiple benefits in terms of computational accuracy, they introduce an extra level of complexity. In this work, it is already required to create the dynamics of the spacecraft including all the environmental models in the differential algebra software, and therefore it is chosen to rely on Cartesian and Kepler elements for this first application of the DA-GMM method. In this section, Cartesian coordinates and Kepler elements are introduced and related.

3.3.1. Cartesian coordinates

In Cartesian coordinates, the state vector is defined by the position and velocity components in the three-dimensional space. This model is widely used due to its simplicity and the direct relation to the spacecraft tracking data. The state of the satellites provided in ephemeris by satellite operators and space agencies are described in these coordinates (SpaceTrack, 2020). Therefore, this set of coordinates is likely to be used to determine the initial state of the satellite. The state vector ($\boldsymbol{x}$) in Cartesian components is given by:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{r} & \boldsymbol{v} \end{bmatrix}^T = \begin{bmatrix} x & y & z & v_x & v_y & v_z \end{bmatrix}^T \tag{3.15}$$

where $\boldsymbol{r}$ and $\boldsymbol{v}$ are the position and velocity vectors, respectively, in Cartesian components for a given reference frame. Figure 3.2 illustrates the defined state model for a body about a general orbit in the inertial reference frame.

The main advantage of this representation is that the equations of motion are simple and allow to easily include the perturbation models. Moreover, this representation does not require any assumptions on the orbit. In turn, since the Cartesian elements are not related to the orbital plane, the state suffers large variations through the orbit, which result in large state derivatives. This can introduce high numerical errors, hindering the selection of the integration time-step. To prevent this problem, different representations have been developed.

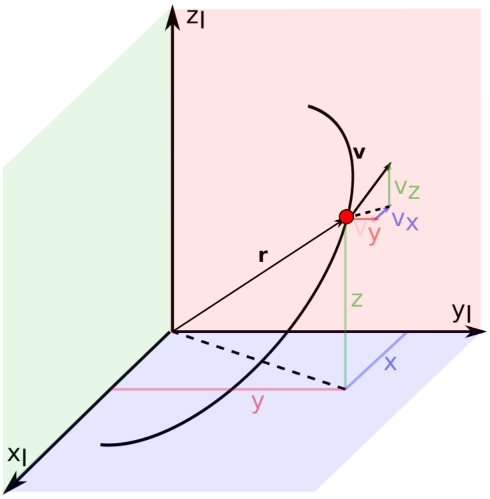


Figure 3.2: Illustration of the Cartesian state model

3.3.2. Kepler elements

When considering the unperturbed two-body problem, the orbital solution can be represented by a set of five constant components defining the shape, size and orientation of the orbit and a sixth component that defines the position of the satellite within the orbit. When orbital perturbations are considered, the elements defining the orbit are not constant anymore. However, the variations are small and this state representation is still very useful and widely used to describe the position of satellites (e.g., in two-line element sets). The position at each time is defined by the instantaneous Kepler elements that form the instantaneous orbital plane.

The orbit is defined by a conical section, which is limited to an ellipse for Earth orbiting satellites. The shape of this ellipse is determined by its semi-major axis (a) and its eccentricity (e). The orbit is contained within a plane. The orientation of the orbital plane with respect to the reference plane of the central body is defined by two angles: the right ascension of the ascending node (Ω) and the inclination (i). The orientation of the orbit within the orbital plane is defined by the argument of periapsis (ω). Finally, the position of the satellite in the orbit can be described by several elements, the most common being the mean anomaly (M) or the true anomaly (θ). The argument of periapsis ω and the true anomaly θ both define an angle in the orbital plane and can be combined in a single variable, the argument of latitude $u = \omega + \theta$.

Figure 3.3 illustrates the Kepler elements defining the orientation of the orbit with respect to the reference plane and the position of the satellite in the orbit. The state vector in Kepler elements is defined by:

$$\boldsymbol{x} = [\begin{array}{cccccc} a & e & i & \Omega & \omega & \theta \end{array}]^T \tag{3.16}$$

The main advantage of Kepler elements is that the variations in the state are smaller which reduces the numerical error. Moreover, this representation is very related to the dynamics of the problem and it is useful to visualise the motion of the spacecraft. The drawback of this method is that, when including the perturbation it yields a more complex differential equation.

This representation encounter two singularities in the cases when $e \rightarrow 0$ and/or $\sin i \rightarrow 0$. The first singularity derives from the fact that in a circular orbit ($e = 0$), the periapsis is not defined as there is no point of closest approach. As a result, when an orbit is close to circular, the argument of periapsis (ω) continues to shift in position. The second singularity occurs when the orbital plane is close or at the equatorial plane, it is with inclinations close to 0 or 180 degrees (equatorial prograde and retrograde orbits). In this case, the concept of ascending node does not apply and therefore the right ascension of the ascending node (Ω) cannot be defined. For the reference missions considered in this work, the orbits are nearly circular, and therefore the first singularity applies.

The state vector in Kepler elements can be obtained from the Cartesian representation as follows (Wertz et al., 2001). First, the norm of the position and velocity vectors are defined as:

$$r = \|\boldsymbol{r}\| \qquad v = \|\boldsymbol{v}\| \tag{3.17}$$

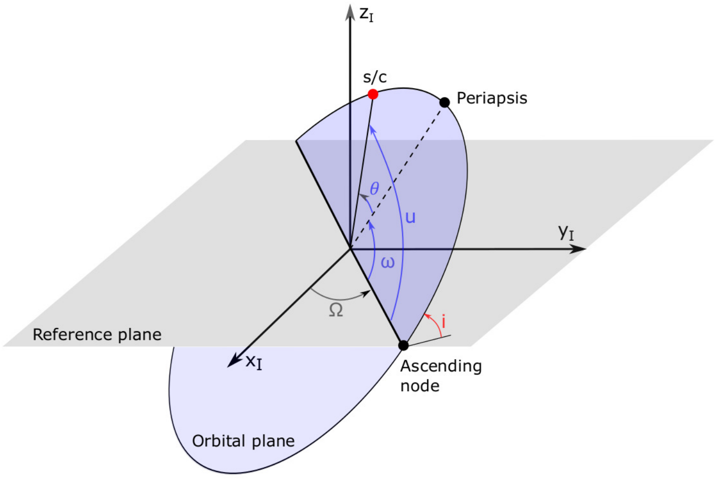


Figure 3.3: Illustration of the Kepler elements of a spacecraft (s/c) in the two-body problem

As auxiliary variables, the orbit angular momentum, $\boldsymbol{h}$ and the vector to the ascending node, $\boldsymbol{N}$, are defined.

$$\boldsymbol{h} = \boldsymbol{r} \times \boldsymbol{v} \tag{3.18}$$

$$\boldsymbol{N} = \hat{\boldsymbol{z}} \times \hat{\boldsymbol{h}} \tag{3.19}$$

where $\hat{\boldsymbol{z}}$ is the unit vector in the direction normal to the equatorial plane. The semi-major axis and eccentricity can be computed from these magnitudes as:

$$a = \left(\frac{2}{r} - \frac{V^2}{\mu}\right)^{-1} \tag{3.20}$$

$$\boldsymbol{e} = \frac{\mathbf{V} \times \mathbf{h}}{\mu} - \frac{\mathbf{r}}{r} \tag{3.21}$$

$$e = \|\boldsymbol{e}\| \tag{3.22}$$

The remaining elements (inclination, i, right ascension of the ascending node, Ω, argument of perigee, ω, and true anomaly, θ are obtained as:

$$\cos i = h_z / |\mathbf{h}| \tag{3.23}$$

$$\tan\Omega = N_y / N_x \tag{3.24}$$

$$\cos\omega = \hat{\mathbf{e}} \cdot \hat{\mathbf{N}} \tag{3.25}$$

$$\cos\theta = \hat{\mathbf{r}} \cdot \hat{\mathbf{e}} \tag{3.26}$$

where the inclination is defined in the range $0° \leq i \leq 180°$, Ω is defined in the range $0° \leq \Omega < 360°$ and can be calculated using the $atan2$ function, ω is in range $0° \leq \omega \leq 180°$ when $(\hat{\mathbf{N}} \times \mathbf{e})\mathbf{h} \geq 0$ and in range $180° < \omega \leq 360°$ otherwise and the true anomaly, θ, satisfies $0° \leq \theta \leq 180°$ when $(\mathbf{e} \times \mathbf{r})\mathbf{h} \geq 0$ and $180° < \theta < 360°$ otherwise.

3.4. Equations of motion

This section introduces the equations of motion that govern the dynamics of the vehicle in space under a series of perturbation that will be described in Section 3.6. For each of the state models introduced, the equations of motion are the variational equations of the state model elements with time. To simulate the trajectory, these equations are integrated over time.

3.4.1. Cartesian coordinates

For Cartesian coordinates, the equations of motion follow a simple derivation form Newton's laws of motion. The dynamics of the vehicle are defined by the addition of the central gravitational force of the central body and the sum of perturbation forces. The acceleration caused by a central body $\boldsymbol{a_{cb}}$ is defined in Cartesian coordinates as:

$$\boldsymbol{a_{cb}} = -\frac{\mu}{r^3}\boldsymbol{r} \tag{3.27}$$

where μ is the gravitational parameter of the central body and r is the norm of the position vector.

The total acceleration of the body is obtained by:

$$\frac{d^2\mathbf{r}}{dt^2} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{f}_{pert} \tag{3.28}$$

where

$$\mathbf{f}_{pert} = \frac{\sum_i \boldsymbol{F}_{pert_i}}{m} \tag{3.29}$$

and $\boldsymbol{F_{pert_i}}$ defines each individual perturbing force vector and m is the vehicle's mass. According to these equations, and decomposing the total perturbing acceleration $\boldsymbol{f_{pert}}$ into the Cartesian components, the equations of motion of the state $\boldsymbol{x}$ in Cartesian coordinates are:

$$\frac{d\mathbf{x}}{dt} = \begin{pmatrix} v_x \\ v_y \\ v_z \\ -\frac{\mu}{r^3}x + f_{pert_x} \\ -\frac{\mu}{r^3}y + f_{pert_y} \\ -\frac{\mu}{r^3}z + f_{pert_z} \end{pmatrix} \tag{3.30}$$

3.4.2. Kepler elements

Kepler elements are constant in the two-body problem. However, this does not apply when orbital perturbations are considered. In this case, the motion of the vehicle can be defined by a set of osculating Kepler elements which determine the evolution of the instantaneous orbit. To evaluate the dynamics of an object in this representation, the variational equations of the Kepler elements under perturbing accelerations are derived. This differential equations for Kepler elements and its derivation can be found in Wakker (2015). To simplify this derivation, the total perturbing acceleration is decomposed in the components of the *RTN*-frame described in Section 3.1. An equivalent form of this representation that uses the true anomaly θ opposed to the mean anomaly M can also be derived. The equations of motion in Kepler elements are:

$$\begin{aligned}
\frac{da}{dt} &= 2\frac{a^2}{\sqrt{\mu p}}\left[f_R e\sin\theta + f_T\frac{p}{r}\right] \\
\frac{de}{dt} &= \sqrt{\frac{p}{\mu}}\left[f_R\sin\theta + f_T(\cos E + \cos\theta)\right] \\
\frac{di}{dt} &= f_N\frac{r}{\sqrt{\mu p}}\cos u \\
\frac{d\omega}{dt} &= -\sqrt{\frac{p}{\mu}}\left[f_N\frac{r}{p}\cot i\sin u + \frac{1}{e}\left\{f_R\cos\theta - f_T\left(1+\frac{r}{p}\right)\sin\theta\right\}\right] \\
\frac{d\Omega}{dt} &= f_N\frac{r}{\sqrt{\mu p}\sin i}\sin u \\
\frac{dM}{dt} &= n - f_R\left[\frac{2r}{\sqrt{\mu a}} - \frac{1-e^2}{e}\sqrt{\frac{a}{\mu}}\cos\theta\right] - f_T\frac{1-e^2}{e}\sqrt{\frac{a}{\mu}}\left(1+\frac{r}{p}\right)\sin\theta
\end{aligned} \tag{3.31}$$

where

$$p = a(1-e^2) \tag{3.32}$$

$$\tan\left(\frac{E}{2}\right) = \frac{1}{\sqrt{\frac{1+e}{1-e}}}\tan\left(\frac{\theta}{2}\right) \tag{3.33}$$

$$M = E - e\sin E = n(t - t_{perigee}) \qquad n = \sqrt{\frac{\mu}{a^3}} \tag{3.34}$$

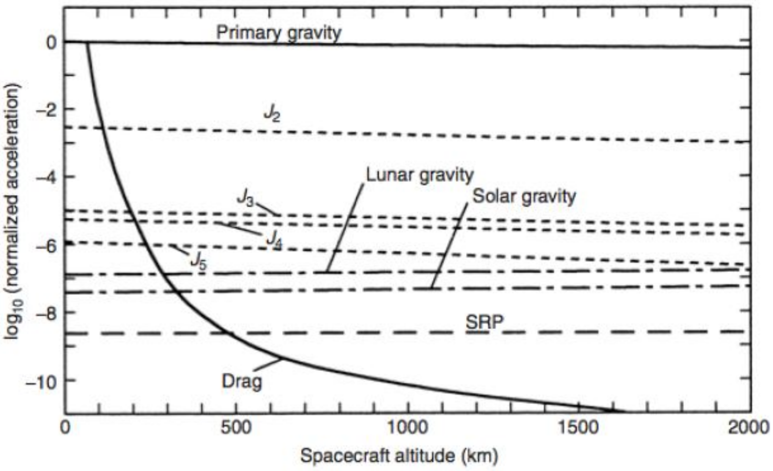


Figure 3.4: Magnitude of the perturbing accelerations normalised with the central gravitational acceleration at Earth's surface versus orbital altitude. Credit: (Fortescue et al., 2011)

It should be noted that for this derivation no assumptions were made on the size of the perturbing accelerations. Therefore this method is valid both for small and large perturbations. To solve the eccentric anomaly E, a root finder algorithm is required.

3.5. Perturbing accelerations

The unperturbed two body problem only includes the satellite and the point-mass central body, and therefore the satellite is only affected by the central body gravitational acceleration. The term perturbing accelerations refers to the sources that contribute to the total acceleration of the satellite besides the central body, and that produce a deviation from the Kepler orbit solution. As observed in Figure 3.4, the magnitude of these forces depends on the spacecraft altitude. While a full environmental and acceleration model that adds all identified perturbations could be included, this would increase the computational cost to prohibitive margins. Therefore, it is required to define a threshold for the required precision in the acceleration model and select the perturbing accelerations to include in the model accordingly. It must be noted that relativistic perturbations are not included in this analysis due to its complexity and the fact that its effect is two orders of magnitude smaller than the solar radiation pressure (SRP) and Drag accelerations (Eshagh and Alamdari, 2007).

In this section, the main sources of perturbation acting on Earth-orbiting satellites are presented, and the equations describing these accelerations are introduced. The accelerations considered are the gravitational acceleration of Earth (modelled using spherical harmonics), the third-body gravitational acceleration, the atmospheric acceleration and the solar radiation pressure.Finally, the acceleration model to be used in the DA-GMM method is selected and motivated.

3.5.1. Spherical harmonics gravity

The gravity field of real bodies does not perfectly correspond to a that of a point mass, but instead presents deviations due to the irregularity in shape and density distribution. These deviations are introduced in the gravitational potential through a set of spherical harmonics as defined in Equation (3.40). From a mathematical perspective, adding the spherical harmonics does not change the procedure to compute the gravitational acceleration, as:

$$\boldsymbol{a}_G = \nabla U_B(\boldsymbol{r}_{BA}) \tag{3.35}$$

where body A is the satellite, body B is the central body and U_B is the gravitational potential exerted by body B. However, from an analysis point of view the effect of each spherical harmonic can be studied individually to select to which degree and order the gravity field should be modelled. This is illustrated in Figure 3.4 by individual lines corresponding to the effect of each spherical harmonic.

3.5.2. Third body gravity

Third-body gravitational accelerations are obtained assuming a known position of the third-body (usually retrieved from an ephemeris). In this case, the gravitational acceleration can be calculated in the same way, through the gradient of the gravitational potential. However, in this case it must be taken into account that the acceleration does not have an inertial frame origin anymore. Let's consider the case of the third-body acceleration of the Sun (body B) to a spacecraft (body A) orbiting Earth (body C). In this example, the gravitational acceleration of the Sun affects the spacecraft but also the Earth. To account for this influence, the third body acceleration on the central body is subtracted from the third body acceleration on the spacecraft as:

$$\boldsymbol{a}_G = \nabla U_B(\boldsymbol{r}_{BA}) - \nabla U_B(\boldsymbol{r}_{BC}) \tag{3.36}$$

For satellites orbiting Earth, the Sun and Moon are usually included in the model as third body perturbations while other bodies such as Jupiter have a smaller effect and are usually discarded.

3.5.3. Aerodynamic acceleration

As observed in Figure 3.4, the drag perturbation has a high effect for low altitudes but its effect quickly decreases as the atmospheric density drops with altitude. However, this acceleration can still be significant, specially for objects in low orbits such as the ISS. Considering the effect of drag only, the aerodynamic acceleration is:

$$\boldsymbol{a}_D = -\left(\frac{\rho v_{\text{air}}^2 C_D S_{ref}}{2m}\right)\hat{\mathbf{v}}_{\text{air}} = -\left(\frac{\rho v_{\text{air}}^2 K}{2}\right)\hat{\mathbf{v}}_{\text{air}} \tag{3.37}$$

where ρ is the atmospheric density at the satellite location, $\mathbf{v}_{\text{air}}$ is the vehicle velocity with respect to the atmosphere, S_{ref} is the reference area, C_D the drag coefficient and m the vehicle mass. If no wind model is included, the basic model assumes a co-rotating atmosphere and the airspeed is equal to the ground-based velocity. The last three variables can be grouped into the ballistic coefficient, K. The atmospheric density at a given location is provided by the atmosphere model and the ballistic coefficient is an input that depends on the satellite properties. This perturbation will be included in the model and the air density will be obtained from the NRLMSISE-00 model, introduced in Subsection 3.6.3.

3.5.4. Solar radiation pressure acceleration

The last acceleration present in the diagram is Solar Radiation Pressure (SRP) which is caused by the transfer of momentum by photons when encountering the satellite. This process depends on the type and roughness of the satellite surface, which impact the photon reflection mechanism. To model this effect, the radiation pressure coefficient (C_r) is introduced, which is defined as one plus the mean reflection coefficient. The mean reflection parameter ranges from 0 for a perfect black body to 1 for an ideal refractor (Wakker, 2015). The first approximation to model the SRP acceleration is through the cannonball model, which assumes a spherical satellite with constant properties through the surface. The acceleration due to SRP in this model is given by:

$$\mathbf{a}_{SRP} = \left(\frac{P}{4\pi c}\right)\left(\frac{C_r S_{ref}}{m}\right)\frac{\mathbf{r}_{BA}}{\|\mathbf{r}_{BA}\|^3} \tag{3.38}$$

where P is the total photon power output of the Sun, which can be obtained from environmental models or considered constant for this purpose. S_{ref} is the reference area, m is the mass of the satellite and $c = 299,792,458$ m/s is the speed of light in the vacuum.

3.5.5. Model selection

Including the full acceleration model is extremely time consuming, and it is not required to achieve an accurate solution. To avoid this issue, a threshold is set for the minimum perturbing acceleration to be included. The selection of perturbing accelerations to include in the model must be consistent, meaning that if a given perturbation source is added, all perturbations with a larger effect must be included as well. In Figure 3.4 the effect of each perturbing acceleration relative to the primary gravity is illustrated up to an altitude of 2,000 km. For higher altitudes, the effect of the spherical harmonics decreases and the effect of the lunar and solar gravity increases

As a reference to set this threshold, the work of Flores et al. (2021) to find the acceleration model for an accurate propagation of orbits applied to the Molniya case is used as a baseline. The Molnyia orbit covers a wide range of orbital altitudes and velocities, and therefore it is a good upper limit for the required accuracy.

In this work, it was found that the threshold required to have an accurate propagation of the Molnyia orbit over 30 days for tracking purposes was of $\epsilon_a = 10^{-8}$ m/s^2. This corresponds to an acceleration (normalised with the local gravity of the Earth, g) of approximately $\hat{\epsilon}_a = 10^{-9}$. However, this is an upper limit for the threshold, which can suppose an excessive increase in the computational load of the method. Instead, the threshold for the normalised acceleration is set to $\hat{\epsilon}_a = 10^{-8}$. This allows to discard the acceleration due to the solar radiation pressure.

Adding the solar radiation pressure is not a trivial task, due to the fact that this acceleration is discontinuous. Differential algebra computation cannot deal with discontinuous functions, since it is based on the Taylor series expansion of the function. Therefore, including the solar radiation pressure would require to apply variable transformations that allow to model this acceleration as a continuous function. This largely increases the complexity of the model and is considered outside the scope of this work. Only perturbing accelerations that have a larger effect will be included. Although in Figure 3.4 it appears that the drag acceleration is smaller than the solar radiation pressure effect for altitude higher than 500 km, it must be considered that the graph shows the nominal drag acceleration. Due to variations in the atmospheric density, drag acceleration can fluctuate in a range of three orders of magnitude and as a result, this acceleration must be included.

To summarise, the accelerations included in the DA-GMM method (hereafter also referred to as *full dynamical model*) include the gravitational acceleration of Earth modelled as a spherical harmonics field up to order and degree six, the central gravitational acceleration from the Sun and the Moon, and the atmospheric acceleration.

3.6. Forces and environmental models

This section defines the models that are created to describe the environment of the satellite and the forces to which it is subjected for the selected acceleration model. This includes a description of the gravity field model and the spherical harmonics concept, the Earth shape and rotation model and the atmosphere model.

3.6.1. Gravity field model

The gravitational acceleration is the main driver of the dynamics of a spacecraft, and should therefore be modelled with sufficient detail to accurately propagate the orbital trajectory of any vehicle. In the case of a spacecraft in LEO, the gravitational pull from the central body (Earth) is the main acceleration acting on the body, while the effect of further orbiting bodies (e.g., Sun and Moon) are considered third-body accelerations.

A simple gravitational model considers the body to be a point mass, which causes a central acceleration (Mooij, 2016). The gravitational acceleration can be expressed as the negative gradient of the potential: $\boldsymbol{a_G} = -\nabla U$. For the point mass acceleration, the gravitational potential is defined by:

$$U = -\frac{\mu}{R} \tag{3.39}$$

where $\mu = GM$ is the gravitational constant of the central body and R is the distance to the central body. However, when the central body is not a perfect sphere, the point mass acceleration does not accurately describe the gravitational interaction between the vehicle and the central body. In this case, more complex models have been derived. When the shape of the body is close to spherical (spheroid) the most common approach is to make use of spherical harmonics to define the gravitational potential U. This model expresses the potential as a summation of the central term and a series of correction terms for the mass asymmetry of the Earth. In this approximation, the gravitational potential is expressed as:

$$U(R,\tau,\delta) = \frac{\mu}{R}\left\{1 + \sum_{n=2}^{n_{max}}\left[\left(\frac{R_e}{R}\right)^n \sum_{m=0}^{n}\left(C_{nm}\cos m\tau + S_{nm}\sin m\tau\right)P_n^m(\sin\delta)\right]\right\} \tag{3.40}$$

where R_e is the radius of the central body (Earth), τ is the longitude, δ is the latitude, P_n^m are Legendre polynomials and C_{nm}, S_{nm} are the expansion coefficients of the spherical harmonics. For the Earth, the full expansion has been obtained, which arrives to degree and order 280.

Although the complete gravitational potential has been modelled, the inclusion of the all the coefficients is not required. As seen in Figure 3.4, the effect of the J_2 acceleration is two orders of magnitude smaller than the central acceleration, and this further reduces for the following terms. It shall be analysed up to which order the gravity field should be modelled to meet the accuracy requirements.

Table 3.2: Body shape parameters of Earth (Mooij, 2016)

Equatorial radius R_e [km]	**Polar radius R_p [km]**	**Ellipticity e**
6378.137	6356.751	$3.3528 \cdot 10^{-3}$

3.6.2. Shape and rotation model

Regular bodies such as the Earth are not perfectly spherical. Their shape is influenced by multiple parameters such as the interaction with other bodies (e.g., gravitational tides) and Earth phenomena, such as motion of tectonic plates. However, for a practical application in the field of propagation in LEO there is no need for an extremely accurate shape model (Mooij, 2016). The approximation to a spherical Earth is sufficient in this context.

Once the Earth shape is defined, the rotation model provides the relation between the Earth-centred inertial frame and the Earth-fixed rotating frame. The rotation of the central body affects the accelerations that are fixed to the body such as the gravity field and the planetary albedo. The basic model assumes that the rotation is constant in rate and orientation. In this simplified model is considered, the rotation rate of the Earth is:

$$\omega_e = 7.29220 \cdot 10^{-5} \text{ rad/s} \tag{3.41}$$

Based on the short propagation times (in the order of days) required for this problem and the higher order assumptions made in other parts of the problem (e.g., initial state uncertainty), the simplified shape and rotation model of the Earth are selected.

3.6.3. Atmosphere model

The properties of the atmosphere have a direct effect on some of the accelerations acting on the spacecraft, but most notably on the drag perturbation. To correctly model the trajectory of an object subject to this acceleration, an atmosphere model has to be derived. The evaluation of an atmosphere model can be computationally expensive. Several models have been derived with different capabilities to adapt to the needs of each mission.

A first simplification is the exponential atmosphere model, which assumes ideal gas behaviour and constant temperature. This approximation introduces large errors and therefore the model is constrained to initial computations. Second, the United States Standard Atmosphere models with the last update in 1976 introduces linear variations in temperature for given altitude ranges (Mooij, 2016). This model, however, does not provide any information on the time or position variations of the atmospheric properties. These variations have shown to create significant differences in the position of satellites in LEO after days of propagation, especially in the along-track direction. To overcome this issue, more complex models have been developed which take into account temporal and position variations.

From the complex Earth atmosphere models that have been derived, the NRLMSISE-00 model is selected. This model is an update from the MSIS-90 model that have been derived from temperature and composition measurements, both from ground and from orbit. NRLMSISE-00 achieves to improve both the MSIS-90 and the Jacchia-70 model by merging advantages of both (Picone et al., 2002). Besides from these advantages, the NRLMSISE-00 is the only advanced atmosphere model currently included in Tudat, which is an important factor for model selection. The NRLMSISE-00 atmosphere model requires to be build with a space weather file [1] that contains information on solar flux, and magnetic flux as a function of time. These values are either measurements from the past or predictions for the future. Moreover, the position (altitude, longitude and latitude) and time are provided as inputs to the model. With this information, the NRLMSISE-00 model provides multiple outputs. from which only the atmospheric density is required and will be used for this application.

[1] Available at: `http://celestrak.com/SpaceData/`. Accessed on: 10-12-2021.

4

Uncertainty Model

This chapter details the models to be used in the propagation of uncertainty for the satellite state and environmental model. First, Section 4.1 introduces the basics of uncertainty modelling by formally defining the concept of a probability density function and providing the mathematical description of the distributions that are required in this work. These are the Gaussian, uniform and log-normal probability density distributions. Following, Section 4.2 describes the uncertainty model for the initial state (position and velocity) of the satellites, accounting or the differences in orbit determination accuracy for the different vehicles and objects considered. In Section 4.3, the uncertainty in the environment is studied and a model is proposed for the atmospheric density and ballistic coefficient.

4.1. Probability distributions

This section introduces the preliminary concept of probability distributions that is required to understand the methodology of uncertainty propagation for the state and environmental variables. To this end, the mathematical descriptions of the probability distributions of interest are provided. These are the Gaussian, uniform and log-normal distributions. In Chapter 5 it is explained how these distributions are converted into a Gaussian Mixture Model. Finally, the covariance error ellipsoid is introduced, as it is a key piece in traditional uncertainty propagation methods and it is used to illustrate the requirement for a Gaussian Mixture Model.

The concept of probability density function (PDF) is used to model the uncertainty in the position and velocity of the satellite, as well as in other parameters of the model. It is a mathematical method to describe the probability distribution of a random variable which. The value of a PDF evaluated at one point can be physically interpreted as the likelihood that the random variable being equal to that point. The probability that a random variable will be between an interval of points in a sample space $S \in [a \quad b]$ is obtained by the integration of the PDF $p(x)$ between the boundary points:

$$P[a \leq X \leq b] = \int_a^b p(x)dx \tag{4.1}$$

Multiple probability density functions have been studied and associated to physical phenomena. For the topic of this work, it is interesting to introduce the Gaussian, uniform and log-normal distribution, which will allow to model the uncertainty in the satellite state and in environmental variables involved in the dynamical model.

4.1.1. Gaussian distribution

The Gaussian or normal distribution is a common way of modelling natural phenomena and a pillar for the development of alternative probability models. In the field of uncertainty propagation of space objects it is particularly interesting, since the errors obtained by orbit determination methods can be closely defined as Gaussian. As a result, all the collision detection methods presented so far have assumed this distribution to define the uncertainty in the initial state of both the satellite and the chaser. A Gaussian or normal distribution is a type of PDF, which can be completely determined from the first two moments, the expected value, μ

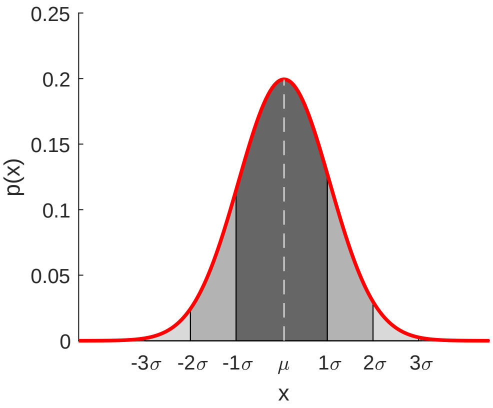


Figure 4.1: Gaussian distribution with $\mu = 3$ and $\sigma = 2$.

and the variance, σ^2. The general form of this distribution is given by:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{4.2}$$

Figure 4.1 presents an example of a generic Gaussian distribution. The three coloured bands delimit the span of the values that lie within the distance of once, twice and three times the value of the standard deviation, respectively. As it can be observed, at a distance of 3σ from the centre of the distribution ($x = \mu$), the value of the PDF drops considerably. The values within the 1σ distance account for 68.27% of the total probability. The 2σ limit covers 95.45% and the 3σ the 99.73%. Since the definition of the function ranges from $[-\infty \;\; \infty]$ it is required to select a threshold after which the probability is considered to be zero. Depending on the required accuracy, this threshold is selected at a given $n\sigma$ value. The application of the Gaussian distribution in this work is purely mathematical and there is no need to define this threshold.

4.1.2. Uniform distribution

A uniform distribution models properties that take an arbitrary value within a given interval. Therefore, the probability is constant within this range of values, which can be closed or open. The notation for a uniform distribution between the bounds a,b is $U(a,b)$ and is PDF is given by Equation(4.3). This distribution can be used to approximate the behaviour of the ballistic coefficient, which is affected by the change in drag coefficient due to the change in attitude of the orbital element.

$$p(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b, \\ 0 & \text{for } x < a \text{ or } x > b \end{cases} \tag{4.3}$$

The mean and standard deviation of the distribution are given by:

$$\mu = \frac{a+b}{2} \tag{4.4}$$

$$\sigma = \frac{b-a}{2\sqrt{3}} \tag{4.5}$$

To convert the state distribution into a Gaussian Mixture Model, the full state vector needs to be modelled as a multi-variate Gaussian distribution. To do so, the elements that are uniformly distributed must be converted to a Gaussian distribution by a variable transformation. This procedure is summarised in Subsection 5.1.1.

4.1.3. Log-normal distribution

The log-normal PDF is useful to model phenomena that follow exponential relations. In this case, it is found that this distribution is the most adequate to describe the uncertainty in the atmospheric density provided by the NRLMSISE-00 model. A log-normal distribution is the probability distribution of a variable whose natural logarithm follows a Gaussian distribution. Therefore, if the probability of a variable X follows a log-normal

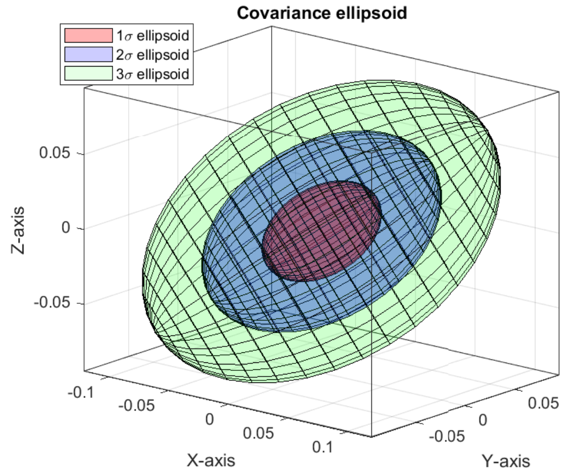


Figure 4.2: Covariance error ellipsoids up to the $1\sigma, 2\sigma$ and 3σ boundaries

distribution, the variable $Y = \ln(X)$ is normally distributed. The probability density function in a log-normal distribution is defined as:

$$p(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{(\ln(x-\mu))^2}{\sigma^2}\right)} \tag{4.6}$$

Unlike the Gaussian distribution, higher moments such as the skewness and kurtosis are non-zero. The result is that the distribution can obtain an asymmetrical, peaky shape compared to a normal distribution. This type of behaviour is common in biological processes and elements involved in exponential relations.

The DA-GMM method requires the initial uncertainty distribution to be Gaussian to apply the splitting library to obtain the Gaussian mixture model. Although there are methods to approximate a log-normal distribution to a Gaussian distribution, in this case it is more efficient to use of the fact that for the distribution lognormal(X), the $\ln(X)$ is normally distributed. The variable $\ln(X)$ becomes the state variable to include in the model.

4.1.4. Covariance ellipsoid

The uncertainty of the satellite position is distributed in the three directions in space. Initially, the position uncertainty in each direction is Gaussian, and the combination of uncertainty in three-dimensional space is visualised in the Gaussian error ellipsoid centred at the satellite's more likely position. When the satellite trajectory is propagated through the nonlinear dynamics, the error ellipsoid deforms. As explained for the Gaussian distribution, the error ellipsoid expands to infinity in the three-dimensional space. The deformation of the uncertainty ellipsoid occurs mostly along the direction of the orbital trajectory. When the propagation is performed for short time periods, the trajectory of the satellite between two points can be considered linear, and therefore the Gaussian ellipsoid can accurately describe the uncertainty distribution. However, as the uncertainty increases and it spans larger portions of the orbits, it begins to curve, forming a banana shape that can no longer be represented by a Gaussian distribution (Luo and Yang, 2017). Figure 4.2 illustrates the Gaussian error ellipsoid centred at (0,0,0) and with a covariance matrix:

$$\boldsymbol{P} = \begin{bmatrix} 1 & 0.5 & 0.2 \\ 0.5 & 1 & 0.4 \\ 0.2 & 0.4 & 1 \end{bmatrix} \cdot 10^{-3} \tag{4.7}$$

In the three-dimensional case, the 1σ ellipsoid covers only 19.87% of the total probability. Expanding to the 2σ-case increases to 72.85%, and the 3σ-rule covers 97.07%.

4.2. Initial state uncertainty model

The main source of uncertainty in the final position of the satellite comes from the initial state uncertainty. In recent years, several techniques have been developed to accurately track the position and velocity of orbital

Table 4.1: Correlation matrix for initial state in *RTN*- reference frame

	r_R	r_T	r_N	v_R	v_T	v_N
r_R	1					
r_T	0.15	1				
r_N	0	0	1			
v_R	-0.15	-0.9	0	1		
v_T	-0.9	0.15	0	-0.15	1	
v_N	0	0	-0.15	0	0	1

objects. The most common techniques are satellite laser ranging, doppler tracking and determination with global positioning systems. From these measurements, and taking into account the different sources of error in the orbit determination (ionosphere and troposphere uncertainties, receiver and satellite clock, multi-path etc.).

The initial state of the satellite can usually be modelled as a Gaussian distribution (Hilton et al., 2019). The mean of the distribution corresponds to the expected position of the satellite, and the standard deviation provides a measure of the tracking uncertainty. However, the uncertainty in the position and velocity components is correlated, and it is also important to evaluate the effect of this correlation. Geul et al. (2017) studied the uncertainty from two-line elements (TLEs) in the state of the GOCE mission and obtained the correlation matrix for this scenario. Although this information is particular to every mission, the main conclusion from this study is that the radial position and along-track velocity and the along-track position and radial velocity have a high negative correlation $\rho \approx -0.9$. For the remaining components, small correlations in the order of $\rho \approx 0.1$ were observed. Similarly, Bai and Chen (2018) studied the correlation in the initial state based on the relative motion between the true state and the predicted space object. Using this approach, the strong negative correlation between the radial and along-track components was physically explained. Another conclusion of this study was that small to no correlation was expected between the in-plane components (*R*,*T*) and the out-of-plane component (*N*). Combining this information it is assumed that the correlation between $r_R - v_S$ and $r_S - v_R$ are -0.9. The correlation between in-plane components and out-of-plane components is assumed to be negligible and the remaining correlations are set to a magnitude of 0.15 with the sign as obtained by Geul et al. (2017). This is summarised in the correlation matrix presented in Table 4.1 and is used for every simulation unless stated otherwise. The diagonal elements are equal to one, by definition, and the remaining components of the symmetric matrix represent the correlation between several position and velocity variables. The values of zero indicate no correlation between in plane and out of plane components. The uncertainty in the state is hereafter described by the standard deviation of each position in the $RTN-$frame.

General satellite state

Depending on the satellite application, a technique of orbit determination with a given accuracy is used. For active LEO satellites measured by GPS, the error is in the order of [1 10] m for position and [1 100] mm/s for velocity (Montenbruck et al., 2002; Rongzhi and Kaizhong, 2020b). However, this accurate tracking data is not publicly available and these advanced tracking technique are not applied to all satellites. In these cases, the tracking information is obtained from the US Space Surveillance Data catalogue as TLEs. For LEO satellites, the averaged standard deviation of the position is $\sigma_R = 102$ m, $\sigma_T = 471$ m and $\sigma_N = 126$ m in the $RTN-$frame (Flohrer et al., 2008). When expressing the uncertainty in the *RTN*-frame it is observed that the absolute error is dominated by the uncertainty in the along-track position (r_T) and in the radial velocity (v_R). The error in the other components is usually smaller by a factor of four to ten. To stay on the safe side it will be assumed that the error in other components is reduced by a factor of four.

The tracking data provided in TLEs represents the mean orbital elements of the satellite. To accurately model the initial state it is required to convert this data from mean orbital elements to perturbed orbital elements. This is done by using the SGP4 perturbations model (Vallado and Crawford, 2008).

ISS state

The ISS is a critical spacecraft since it is populated by a crew and requires a very precise orbit determination. This allows to obtain a more accurate estimate of the collision risk and therefore lower the thresholds for avoidance operations and increase the safety of life at the vehicle. With the available single-frequency GPS

Table 4.2: Standard deviation in initial state given in *RTN*-components of position and velocity

Mission	$\sigma_{\mathbf{R.R}}, \sigma_{\mathbf{R.N}}$ **[m]**	$\sigma_{\mathbf{R.T}}$ **[m]**	$\sigma_{\mathbf{V.R}}$ **[m/s]**	$\sigma_{\mathbf{V.T}}, \sigma_{\mathbf{V.N}}$ **[m/s]**
ISS	0.2	0.8	0.004	0.001
General LEO satellite	1	4	0.01	0.0025
TLE	100	400	1	0.25
Space debris (Case 1)	250	1000	1	0.25
Space debris (Case 2)	10	40	0.1	0.025

receivers on board the ISS, an orbit determination accuracy of 1 m is reached, which further increases for dual-frequency (Wermuth et al., 2012). The ACES mission on board the ISS has equipped the experiment with a JAVAD GNSS receiver. With this instrument, the errors on the position have decreased to 0.03 m (Shum et al., 2008). However, this data is not publicly available and this work is bounded to the precision of TLEs.

Space debris state

The position of the space debris is also tracked by the US Space Surveillance catalogue for objects up to 10 km in diameter. However, in these cases the position uncertainty can go as high as 1 km. Velocity uncertainty also increases for inactive satellites, being in the order of 0.1 – 1 m/s. The tracking uncertainty shows to be related to the orbit inclination and eccentricity. This relation can be taken into account when modelling the initial state of the space debris. Alternatively, a conservative approach can be adopted, selecting the maximum uncertainty independently of the debris orbit.

Considering that uncertainty is generally larger in the transverse direction for position and radial direction for velocity, and taking into account the typical values on tracking error on several missions, a set of reference values for the standard deviation in tracking several objects is defined. The values summarised in Table 4.2 and will be used herein. Without further knowledge, any tracking data coming from a TLE is assumed to have the accuracy corresponding to this method.

4.3. Uncertainty model in environment forces

The initial state of the satellite is not the only source of uncertainty. In Section 3.6 the different acceleration models that describe the environment of the satellite along its trajectory are introduced. Although these models have been designed to provide an accurate representation of the real environment, they till introduce uncertainty due to imperfect knowledge or to the effect of simplification. Errors in the model can either be parametrised as errors in the model variables or have to be studied as acceleration errors. If possible, the first approach is preferred. To perfectly introduce all the uncertainty in the model, all the parameters included in the simulation should be considered uncertain variables. This includes all the gravity field spherical harmonics of the bodies considered, the parameters defining the Earth shape and mass and the parameters defining the perturbing accelerations (drag and solar radiation pressure). Preliminary simulations run by Römgens (2011) for a similar problem using verified interval propagation showed that including seven parameters as uncertain variables resulted in a computational cost of 24 hours for an orbit integration of 24 hours. Although this is just an estimation, which could be improved for the problem at hand with current software, it largely exceeds the computational time that is aimed for operational collision detection. Therefore, the number of variables modelled as uncertain parameters has to be limited to the variables that have a larger effect on the problem.

The variables describing the gravitational spherical harmonics and the mass and size properties of the Earth are known with high accuracy and therefore will be considered constant with no uncertainty. The drag acceleration is subject to a higher uncertainty from the point of view of the model parameters. To account for this, the atmospheric density and the ballistic coefficient are modelled as uncertain parameters. Finally, the effects of simplification have to be included in the model. All perturbing accelerations that are not modelled are merged in a single variable.

4.3.1. Atmospheric density

One of the important sources of uncertainty that can be added to the model is the atmosphere model, especially the atmospheric density. As discussed in Subsection 3.6.3, the NRLMSISE-00 atmosphere model will be used in the simulations. The uncertainty in the density provided by this model has been treated in multiple ways. The first approximation is to use common probability distributions. Ronse and Mooij (2013) used a uniform distribution to model the error in density and Yang et al. (2016) used a normal distribution. However, these choices were not properly motivated in relation with the atmospheric models of choice. The approach adopted by Hoogendoorn et al. (2017) tailors the distribution to the atmospheric model. In this work, a similar approach is followed and customised to the mission under study.

The study developed by Picone et al. (2002) analyses the NRLMSISE-00 atmospheric model and provides extensive information on the atmospheric density and its comparison with orbital data from the NRLMSIS database and other models. This study references the work by Hedin (1988), who proved that the residuals of the atmospheric density follow a log-normal distribution in the case of the MSIS-86 models. Picone et al. (2002) use the same distribution for the NRLMSISE-00 and provide the parameters of the log-normal distribution of the ratio of observed density ρ_{data} to model density ρ_{model}:

$$X_\rho = \frac{\rho_{data}}{\rho_{model}} \tag{4.8}$$

$$\ln X_\rho \sim \mathcal{N}(\beta, \omega^2) \rightarrow X_\rho \sim \text{lognormal}(\beta, \omega^2) \tag{4.9}$$

The parameters defining this distribution are the mean β and the standard deviation ω. A negative mean indicates that the model overestimates the measured values on average. These parameters are not constant. The standard deviation has been shown to increase with altitude and with active geomagnetic conditions. Picone et al. (2002) provides the log-normal distribution parameters for the cases of quiet, active and average geomagnetic conditions and for four altitude bands. Both of the reference missions under study fall within the 400-800 km altitude band, where the statistical properties of the non-dimensional density ratio X_ρ are:

$$\beta = -0.08 \qquad \omega = 0.25 \tag{4.10}$$

The relation between the parameters of the normal distribution of $\ln X_\rho$ (β and ω) and the mean μ and standard deviation σ of the random variable X_ρ is given by:

$$\mu = \exp\left(\beta + \omega^2/2\right) \tag{4.11}$$

$$\sigma^2 = \exp(2\beta + \omega^2)(e^{\omega^2 - 1}) \tag{4.12}$$

For the values of β and ω associated to $\ln X_\rho$, the mean and standard deviation of the non-dimensional density ratio is are:

$$\mu = 0.9524 \qquad \sigma = 0.2418 \tag{4.13}$$

In the studies by Ronse and Mooij (2013); Hoogendoorn et al. (2017), the distribution was characterised by $\mu = 1, \sigma = 0.12$. In this case, the uncertainty in the density is larger since the problem develops in a higher altitude region.

It is trivial that the observed density is $\rho_{data} = X_\rho \cdot \rho_{model}$. The PDF of the density is therefore the PDF of the product $X_\rho \cdot \rho_{model}$, which is given by:

$$\rho \sim \text{lognormal}(\beta + \ln\left(\rho_{model}\right), \omega^2) \tag{4.14}$$

4.3.2. Ballistic coefficient

To completely constrain the drag acceleration model, the ballistic coefficient is introduced and treated as a parameter with uncertainty. The ballistic coefficient K is defined as:

$$K = \frac{C_D A}{m} \tag{4.15}$$

where C_D is the drag coefficient, A is the reference area and m is the mass. In this way, the uncertainty in the drag coefficient is studied together with the uncertainty of the satellite mass (which changes during orbit and is an unknown for space debris) and the uncertainty in the reference area.

The drag coefficient is known estimated to launch for a set of atmospheric conditions. Although a reference value is known for active satellites, this still includes uncertainties due to the variability of the coefficient with the environment and the spacecraft attitude. An approach to calculate the drag coefficient is through fitting of orbital measurements considering C_D the only unknown variable. However, this technique is considered out of the scope of this work, and it is preferred to model the drag coefficient as a constant parameter.

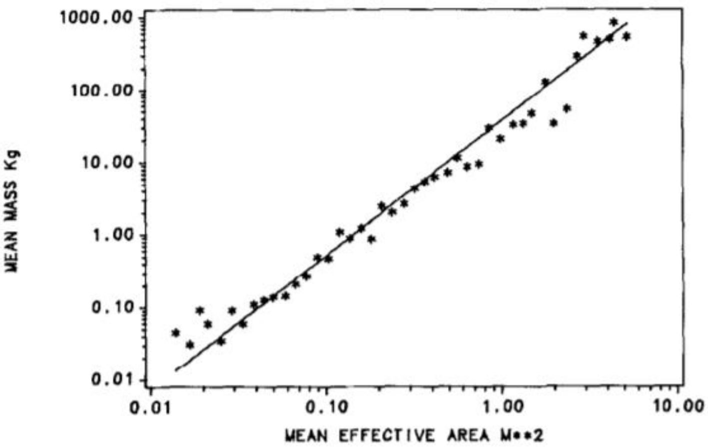


Figure 4.3: Mean debris mass and mean debris area based on the analysis of 24 satellite breakups and 200 pieces of debris with known properties (Badhwar and Anz-Meador, 1989)

Table 4.3: Summary of the uncertainty model of environment forces

Symbol	Parameter	Distribution
X_ρ	Observed to model density ratio	Log-normal (-0.08,0.25)
K	Ballistic coefficient ISS	Uniform (0.0025,0.011)
	Ballistic coefficient general spacecraft	Uniform (K-0.015,K+0.015)
	Ballistic coefficient general debris	Log-uniform (10^{-4},5)

The spacecraft in reality will be changing its attitude and in the case of space debris even tumbling. Assuming that the change in attitude (within a different range for each satellite) is constant, a uniform distribution is selected as the best choice to model this behaviour. This approach has been used by Bhusal and Subbarao (2019) to quantify the ballistic coefficient uncertainty of a CubeSat in LEO. Römgens (2011) assumed that the ballistic coefficient was an interval variable in the range of $10^{-4} - 1$ m^2/kg. In the mission under study, it is possible to extract more detailed information of the satellites. Three models will be provided to model the ballistic coefficient according to the reference mission: for the ISS, for generic LEO satellites and for a general piece of space debris or unknown active satellite.

The ISS has been studied in detail to provide precise orbit determination and orbit propagation. From initial studies, the engineering applications modelling the trajectory of the ISS such as the Station Reboost Analysis Program (STRAP) used a ballistic coefficient of $\frac{1}{12}$ft^2/lb $\approx$ 0.0035 m^2/kg. NASA contractors later reported that the ballistic coefficient falls in the range of $\frac{1}{14}-\frac{1}{15}$ft^2/lb $\approx$ 0.0028−0.003 m^2/kg (Rylaarsdam, 1996). This information provides a reference, but is not updated to the current configuration and orbital measurements of the ISS. According to ESA (2008) the C_D of the ISS is 2.07 and the cross-sectional area varies between 700 – 2300 m^2. For a mass of 419,725 kg, this results in a ballistic coefficient of $K \approx 0.0034 - 0.011$ m^2/kg. To include all this information and other potential sources of uncertainty, the ballistic coefficient is assumed to vary uniformly in the range (0.0025 – 0.011) m^2/kg.

For generic LEO satellites,it is common to assume that a spacecraft in LEO has a drag coefficient of 2.2. This assumption is based on spherical spacecraft, and when expanding to cuboid shapes it increases to 2.7. For this approximation the drag coefficient is assumed to have a uniform distribution in the range $C_D \sim [2\ \ 3]$. Assuming a spacecraft mass of 820 kg and 82 kg of fuel, the mass fluctuates in the range $m \sim [738\ \ 820]$ kg. Assuming that the change in frontal area is accounted in the variability of the drag coefficient, and with a frontal area of 18.86 m^2 the ballistic coefficient is assumed to have a uniform distribution in the range [0.046 0.076] m^2/kg.

Modelling the ballistic coefficient of an unknown piece of space debris requires considering a wide range of possibilities for the debris shape and mass. From the analysis of 200 pieces of debris and 24 satellite breakups by Badhwar and Anz-Meador (1989), the power law relation illustrated in Figure 4.3 is obtained. From this relation, the ratio $\frac{A}{m}$ lies in the range of $[10^{-3}\ \ 1]$ m^2/kg. Including the uncertainty in the drag coefficient, the ballistic coefficient of an unknown piece of space debris is assumed to have a log-uniform distribution in the range $[10^{-4} \;\; 5]$ m^2/kg. A log-uniform distribution represents that $\log(K)$ presents a uniform distribution in the range $[-4 \;\; \log(5)]$. The same distribution is assumed for a general spacecraft of which no information is known regarding the size and mass.

5

Uncertainty propagation with Hybrid DA-GMM method

This chapter focuses on the first phase of the collision probability calculation process, the propagation of uncertainty. As explained in Chapter 2, the hybrid Differential Algebra and Gaussian Mixture Model (DA-GMM) method is selected as the best candidate to perform this propagation. The objective of this algorithm is to provide a valuable alternative to a Monte Carlo analysis with a significant reduction of the computational time, which can be used during satellite operations. The method is clearly divided into two parts: the splitting of the initial distribution into a Gaussian mixture formed by N Gaussian Mixture Elements (GMEs) and the uncertainty propagation of each individual GME to time of closest approach. The former addresses the deformation of the initially Gaussian uncertainty into a non-Gaussian distribution due to the effect of nonlinear dynamics. The latter allows to propagate a Gaussian uncertainty through the nonlinear dynamics without simplifying to the state transition matrix.

In Section 5.1, the mathematical basis of the Gaussian Mixture Model is explained and applied to three cases: the Gaussian, uniform and log-normal distributions. The final step of the GMM splitting is a quadratic optimisation problem that solves for the weights of each GME. This process poses a computational challenge, and the methodology followed to solve it is explained in Subsection 5.1.2. Once the splitting is complete, the solution is tested to select the optimal number of elements. To explain the uncertainty propagation process, the theory of Taylor series integration that forms the basis of this method is introduced in Section 5.2. Related to this concept, the theory of differential algebra (i.e., the computational implementation of Taylor series integration) is explained in Section 5.3. The result of this integration is the Taylor expansion of the final state of the satellite at TCA with respect to a deviation of the initial state. The process to obtain the propagated uncertainty from this Taylor expansion is not trivial, and is explained in Section 5.4.

5.1. Gaussian Mixture Model

This section introduces the concept of a Gaussian Mixture Model and covers the topic of splitting the probability density function into a sum of Gaussian distributions. To correctly include the GMM in the differential algebra propagation tool, the initial probability density function must be a multivariate distribution including all the elements with uncertainty. Since the main source of uncertainty (i.e., the satellite position and velocity) is normally distributed, a multivariate Gaussian distribution is selected. However, as explained in Section 4.3, not all the elements in the model follow a Gaussian distribution. To compensate for this fact, a set of variable transformations is used. The concept of a GMM is first introduced and illustrated through Example 5.1. In Subsection 5.1.1 the mathematical description of these transformations and the splitting process is given. Once the theory is introduced, a description and comparison of the methods available to perform the Gaussian split is presented in Subsection 5.1.2. The results are compared with the available verification data and the approximation error is calculated. Once the splitting process is implemented, the method is tested by performing a Gaussian hypotheses testing on the propagated GMEs. This allows to select the required number of GMEs in Subsection 5.1.3.

Example 5.1 - Concept of a Gaussian Mixture Model

A Gaussian Mixture model is an approximation of a probability distribution as a weighted sum of N Gaussian distributions. This concept is formally expressed expressed by:

$$\hat{p}(t,\boldsymbol{x}) = \sum_{i=1}^{N} \omega_i p_g\left(\boldsymbol{x};\boldsymbol{\mu}_i,\boldsymbol{P}_i\right) \tag{5.1}$$

where N is the total number of Gaussian kernels, $\boldsymbol{\mu_i}, \omega_i$ and $\boldsymbol{P_i}$ are the mean, weight and covariance matrix of the i^{th} Gaussian density function $p_g\left(\boldsymbol{x};\boldsymbol{\mu}_i,\boldsymbol{P}_i\right)$. To illustrate this concept, consider the case of a satellite, whose position uncertainty follows a Gaussian distribution with covariance $\boldsymbol{P}_{init}$:

$$\boldsymbol{P}_{init} = \begin{bmatrix} 4.15 & -6 & 1.19 \\ -6 & 12.41 & -2.26 \\ 1.19 & -2.26 & 1.42 \end{bmatrix} \mathrm{m}^2 \tag{5.2}$$

In this example, a reduced number of kernels (e.g., five) is selected to better illustrate the concept. Then, the initial state uncertainty is approximated by a mixture of five Gaussian distributions with the same individual covariance and different weight. The mathematical theory and process behind this approximation are explained in Subsection 5.1.1 and Subsection 5.1.2, respectively. Figure 5.1 illustrates this concept. On the left, 5,000 samples drawn from the original Gaussian distribution are represented, together with the 3σ covariance ellipsoid. On the right, the equivalent Gaussian Mixture Model is represented through the samples drawn from the individual GMEs, illustrated in different colours. Note that the mean of each Gaussian mixture element is distributed along the principal axis of the ellipsoid and a larger weight is given to the central elements.

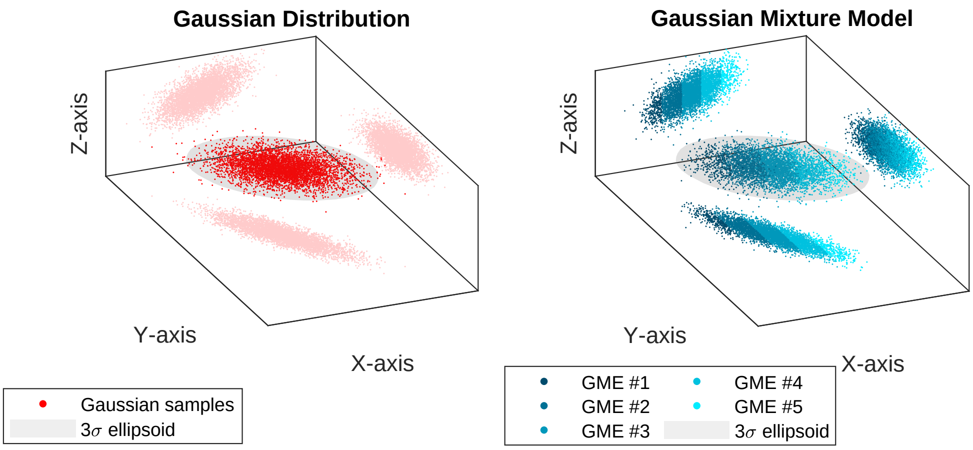


Figure 5.1: Illustration of the three-dimensional Gaussian distribution. *Left*: Samples drawn from a Gaussian distribution in the 3σ ellipsoid. *Right*: Samples drawn from the corresponding GMM with five elements in the 3σ ellipsoid.

This example aims to illustrate how a Gaussian distribution for the satellite position can be represented as a Gaussian Mixture Model. On the following sections, the methodology to transform a multivariate Gaussian distribution into a Gaussian Mixture Model is explained.

5.1.1. Mathematical description

From the literature study performed, it is concluded that the main sources of uncertainty in the propagation of the satellite trajectory come from: the orbit determination process, the atmospheric model and the physical characteristics of the satellite. These sources of uncertainty are included in the propagation through three categories: the satellite state including position and velocity ($\boldsymbol{x} = [\boldsymbol{r}\,\boldsymbol{v}]$), the atmospheric density (ρ) and the ballistic coefficient ($K = \frac{C_D A}{m}$). Although the satellite state can initially modelled by a Gaussian distribution, after the propagation of uncertainty through the nonlinear dynamics this distribution becomes non-Gaussian. To compensate for this effect, the Gaussian Mixture Model approach approximates the initial Gaussian distribution by a weighted sum of Gaussian distributions with smaller standard deviation. When these individual elements are propagated, their weighted summation will approximate the final non-Gaussian probability density function.

To implement this method, the initial state must be modelled by a single multivariate density function, which for simplicity is selected to be a multivariate Gaussian distribution. Referring back to the road-mao presented in Chapter 2, this corresponds to Step 1. The variables forming the satellite state are correlated and can be modelled as a multivariate Gaussian distribution, as explained in Section 4.2. However, the atmospheric density follows a lognormal distribution and the ballistic coefficient is uniformly distributed. To correctly include it in the model, a variable transformation is applied to convert the uniform and lognormal distributions into a Gaussian distribution that can be included in the multivariate model assuming no correlation with the satellite state. In this section, the theory to apply these transformations is derived and the mechanism for splitting a distribution into a Gaussian mixture is explained.

Uniform to Gaussian transformation

The objective of this section is to find a change of variable that can relate a uniformly distributed variable (which in practice will represent the ballistic coefficient) with a normally distributed variable. The latter will be used to obtain the GMM and the change of variable will be applied during the propagation in the DA environment. Assuming that a random variable X has a probability density function f(x), by definition, the probability that the random variable lies between two given values a and b is:

$$\mathrm{P}(a \leqslant X < b) = \int_a^b f(x)dx \tag{5.3}$$

By applying an arbitrary variable transformation $y(X)$, Equation (5.3) becomes:

$$\mathrm{P}(y(a) \leqslant Y < y(b)) = \mathrm{P}(a \leqslant X < b) = \int_a^b f(x)dx = \int_{y(a)}^{y(b)} f(x(y))\frac{dx}{dy}dy \tag{5.4}$$

By defining the right-hand integrand in terms of y, $g(y) = f(x(y))\frac{dx}{dy}$, the expression becomes:

$$\mathrm{P}(y(a) \leqslant Y < y(b)) = \int_{y(a)}^{y(b)} g(y)dy \tag{5.5}$$

which demonstrates that $g(y)$ is the probability density function of the transformed variable. In this scenario, the objective is to transform a uniform distribution $f(x) = U(a,b)$ into a normal distribution $g(y)$. To simplify this process, the normal distribution is set to have $\mu = 0$ and $\sigma = 1$, leaving:

$$f(x(y)) = \frac{1}{b-a} \tag{5.6}$$

$$g(y) = \frac{1}{\sqrt{2\pi}}e^{(-\frac{1}{2}y^2)} \tag{5.7}$$

Which, according to the definition of $g(y)$, allows to express the differential equation:

$$\mathrm{d}x = \frac{b-a}{\sqrt{2\pi}}e^{(-\frac{1}{2}y^2)}\,\mathrm{d}y \tag{5.8}$$

This function can be integrated with the use of the error function, defined as $\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}dt$ and matching the centre of both distributions as boundary condition: $x(0) = a + \frac{b-a}{2}$. The resulting variable transformation is:

$$x = \frac{b-a}{2}\mathrm{erf}(\frac{y}{\sqrt{2}}) + \frac{a+b}{2} \tag{5.9}$$

To illustrate this transformation, 10^7 samples are drawn from a unit normal distribution, and their histogram is plotted in Figure 5.2. As observed, the sample size is large enough to replicate the shape of the probability distribution. Then, Equation (5.9) is applied to the samples, and the histogram of the transformed data is also represented in Figure 5.2. As observed, the transformed data follows a uniform distribution, with the bounds specified by a and b.

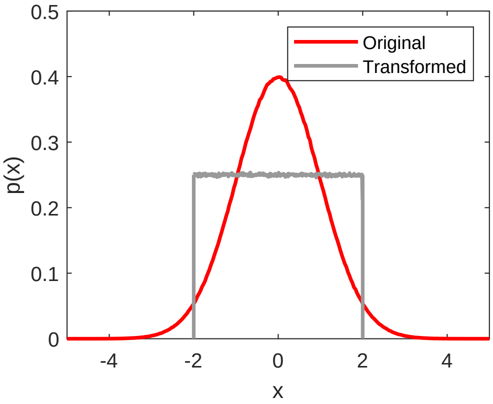


Figure 5.2: Transformation of data from original Gaussian distribution with $\mu = 0$ and $\sigma = 1$ to uniform distribution with bounds $a = -2$ and $b = 2$. Illustration with 10^7 samples.

Lognormal to Gaussian transformation

In the case of the lognormal distribution, the transformation function is intrinsically defined. A random variable X follows a lognormal distribution if:

$$\ln X \sim \mathcal{N}\left(\mu, \sigma^2\right) \tag{5.10}$$

Therefore, in this case the relationship between the normally distributed variable Y and the log-normally distributed variable X is given by:

$$x = e^y \tag{5.11}$$

Splitting theory

The approach to split a multivariate distribution into a GMM has been studied and implemented by Horwood et al. (2011), DeMars et al. (2013), Vittaldev and Russell (2016), and Psiaki et al. (2015). The available methods can be divided into two categories: the univariate splitting technique and the linear matrix inequality technique. The former method is based on producing the Gaussian sum approximation of the unit one-dimensional Gaussian distribution, $p_g(x; 0, 1)$. This approximation, the univariate split, is then scaled to fit any multivariate Gaussian distribution, $p_g(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{P})$. The main benefit of this approach is that it provides a high accuracy approximation and is valid for any number of GMEs. Moreover, the computational requirement is concentrated on the univariate split, which is only performed once. Therefore, this method is fast and can be used for any type of problem. The linear matrix inequality method introduced by Psiaki et al. (2015) has been used by Sun et al. (2019) for uncertainty propagation in the hybrid DA-GMM method. In this case, the Gaussian split is performed to the full multivariate probability density function at once, assuming a constant distribution of weights and means. In this method, the Gaussian mixture elements are selected so that their covariances lie below a linear matrix inequality upper limit. The main disadvantage of this method is that the accuracy depends on the number of Gaussian mixture elements, requiring more than 3,000 kernels for a good approximation. Based on this limitation, the univariate splitting method is selected to perform the Gaussian mixture splitting.

The univariate splitting method can be divided into two phases: optimisation and refinement. The former phase consists in splitting the univariate unit Gaussian distribution by solving a quadratic optimisation problem. The latter consists in scaling the univariate split to approximate the distribution in the remaining dimensions. The metric selected to evaluate the accuracy of the resultant mixture is the L_k, which measures the distance between two distributions p and q. For a given order k, the metric is defined as:

$$L_k[p\|q] = \int_\Omega \left|p(\mathbf{x}) - q(\mathbf{x})\right|^k \, \mathrm{d}\mathbf{x} \tag{5.12}$$

The distance metric L_2 of order $k = 2$ is chosen to assess the accuracy, since it can be calculated analytically. Although the concept is hard to interpret physically, a visual depiction of the relationship between L_2 and the difference in uncertainty distributions is illustrated in Chapter 8. Based on this definition, the objective function is set by Horwood et al. (2011) as $\frac{1}{2}L_2$:

$$E = \frac{1}{2}\int_{-\infty}^{\infty}\left(p_g(x;0,1) - \sum_{\alpha=1}^{N} w_\alpha p_g\left(x;\mu_\alpha,\sigma_\alpha^2\right)\right)^2 \mathrm{d}x = \frac{1}{2}L_2 \tag{5.13}$$

The optimisation problem requires to minimise E, subject to

$$\sum_{\alpha=1}^{N} w_\alpha = 1, \quad w_\alpha \geq 0, \quad \alpha = 1,\ldots,N \tag{5.14}$$

$$\mu_1 \leq \mu_2 \leq \cdots \leq \mu_N \tag{5.15}$$

$$\sigma_\alpha \leq \sigma < 1, \quad \alpha = 1,\ldots,N \tag{5.16}$$

Considering that both distributions are Gaussian, Equation (5.13) can be expressed as:

$$E = \frac{1}{2}\boldsymbol{w}^T\mathbf{M}\boldsymbol{w} - \boldsymbol{w}^T\boldsymbol{n} + \frac{1}{4\sqrt{\pi}} \tag{5.17}$$

where

$$(\boldsymbol{w})_\alpha = w_\alpha \qquad (\boldsymbol{n})_\alpha = p_g\left(\mu_\alpha;0,\sigma_\alpha^2+1\right) \qquad (\mathbf{M})_{\alpha\beta} = \mathcal{N}\left(\mu_\alpha-\mu_\beta;0,\sigma_\alpha^2+\sigma_\beta^2\right) \tag{5.18}$$

This optimisation problem is still difficult to solve. To simplify the process, all the GMEs are set to have a common standard deviation σ_α (homoscedastic assumption), and the means are evenly distributed along the $\pm 6\sigma$ range. Following these assumptions, Equation (5.17) can be minimised to find the weights of each GME, w_α.

Based on this theory, the following procedure is followed to approximate the initial multivariate Gaussian distribution $p_g(\boldsymbol{x};\boldsymbol{v},\boldsymbol{Q})$. To do so, first the GMM for a univariate distribution is created, forming a so-called univariate splitting library which approximates $p(x;0,1) \approx \sum_{\alpha=1}^{N}\omega_\alpha p_g\left(x;\mu_\alpha,\sigma^2\right)$ (Horwood et al., 2011).

1. Select the standard deviation of the univariate splitting library $\sigma \in (0,1)$, also referred to as the refinement parameter. This selection can be either manual or following

$$\sigma = \frac{2\sqrt{c-1}\,v^1}{3nt}\sqrt{\frac{Q^{66}}{Q^{11}}} \tag{5.19}$$

where $c > 1$ is a tuning parameter that controls the growth of the component covariances, v^1 is the mean of the first state variable, $n = \sqrt{\mu_E/(v^1)^3}$ and t is an upper bound for the total propagation time. The variable Q^{ab} refers to element in row a and column b of the $\boldsymbol{Q}$ matrix.

2. Compute the number of terms N in the Gaussian sum according to $N = 1+2m/\sigma$, where $m = 4$ if $\sigma \geq 1/2$ and $m = 6$ if $\sigma < 1/2$.

3. For $\alpha = 1,\ldots,N$, compute the mean of each GME of the univariate distribution $\mu_\alpha = -m+\sigma(\alpha-1)$

4. Compute the matrix $\boldsymbol{M}$ and the vector $\boldsymbol{n}$ formed by:

$$(\mathbf{M})_{\alpha\beta} = \mathcal{N}\left(\mu_\alpha-\mu_\beta;0,2\sigma^2\right), \quad (\boldsymbol{n})_\alpha = \mathcal{N}\left(\mu_\alpha;0,\sigma^2+1\right) \tag{5.20}$$

5. Minimise the objective function defined by Equation (5.17) subject to the constraints defined in Equation (5.14) to obtain the weights of the unit univariate distribution. w_α.

6. For $\alpha = 1,\ldots,N$, compute the weights $\tilde{w}_\alpha$, means $\tilde{\mu}_\alpha$ and variation $\tilde{\sigma}_\alpha^2$ of the unit univariate distribution refined along the first coordinate of $\boldsymbol{x}, x_1$

$$\begin{aligned} \tilde{w}_\alpha &= \tfrac{\sqrt{2\pi}}{\sqrt{1-\sigma^2}} w_\alpha \exp\left[\tfrac{\mu_\alpha^2}{2(1-\sigma^2)}\right] \\ \tilde{\mu}_\alpha &= \tfrac{\mu_\alpha}{1-\sigma^2}, \quad \tilde{\sigma}^2 = \tfrac{\sigma^2}{1-\sigma^2} \end{aligned} \tag{5.21}$$

7. For $\alpha = 1, ..., N$, the univariate distribution is further refined to be defined on the interval $(\nu^1 - m\sqrt{Q^{11}}, \nu^1 + m\sqrt{Q^{11}})$, by a GMM with weights $\hat{w}_\alpha$, means $\hat{\mu}_\alpha$ and variation $\hat{\sigma}^2_\alpha$ defined as:

$$\hat{w}_\alpha = \tilde{w}_\alpha, \quad \hat{\mu}_\alpha = v^1 + \sqrt{Q^{11}}\tilde{\mu}_\alpha, \quad \hat{\sigma}^2 = \tilde{\sigma}^2 Q^{11} \tag{5.22}$$

8. To obtain the GMM approximation of the multivariate distribution $p_g(\boldsymbol{x}; \boldsymbol{\nu}, \boldsymbol{Q}) \approx \sum_{\alpha=1}^{N} \bar{w}_\alpha \mathcal{N}\left(\boldsymbol{x}; \overline{\boldsymbol{v}}_\alpha, \overline{\mathbf{Q}}\right)$, the univariate distribution refined along x^1 is multiplied by $p_g(\boldsymbol{x}; \boldsymbol{\nu}, \boldsymbol{Q})$ and the resulting parameters of the GMM, $\overline{\mathbf{Q}}, \overline{\boldsymbol{v}}_\alpha$ and $\bar{w}_\alpha$ are computed as:

$$\begin{gathered} \overline{\mathbf{Q}} = \left(\hat{\sigma}^{-2}\boldsymbol{e}_1\boldsymbol{e}_1^T + \mathbf{Q}^{-1}\right)^{-1}, \quad \overline{\boldsymbol{v}}_\alpha = \overline{\mathbf{Q}}\left(\hat{\sigma}^{-2}\hat{\mu}_\alpha\boldsymbol{e}_1 + \mathbf{Q}^{-1}\boldsymbol{v}\right) \\ \bar{w}_\alpha = \hat{w}_\alpha p_g\left(\hat{\mu}_\alpha - \boldsymbol{e}_1^T\boldsymbol{v}; 0, \hat{\sigma}^2 + \boldsymbol{e}_1^T\mathbf{Q}e_1\right) \end{gathered} \tag{5.23}$$

where $\boldsymbol{e_1}$ is a unit vector along the direction of the first state component.

9. Finally, the weights are re-normalised for $\alpha = 1, ..., N$ following $\bar{w}_\alpha = \bar{w}_\alpha / \sum_{\beta=1}^{N} \bar{w}_\beta$

5.1.2. Gaussian Mixture Splitting Methodology

Although the splitting procedure requiring to solve a quadratic optimisation problem is apparently straightforward, it can present a computational challenge. The number of Gaussian mixture elements determines the number of unknowns in the optimisation problem, and therefore its complexity. To obtain an accurate solution, traditional optimisation methods using double precision floating-point arithmetic are insufficient. Instead, quadruple precision is required, which is not available in common software. In the literature consulted, this problem is usually eluded by making use of pre-calculated univariate splitting libraries, such as the one documented by Vittaldev and Russell (2016). However, these libraries have generated splitting data up to $N = 39$ Gaussian mixture elements. For this work it is expected that a higher number of elements will be required, and other alternatives are studied. In this section, different methods to solve the problem presented in Equation (5.17) are compared and the final results with the winning method are provided.

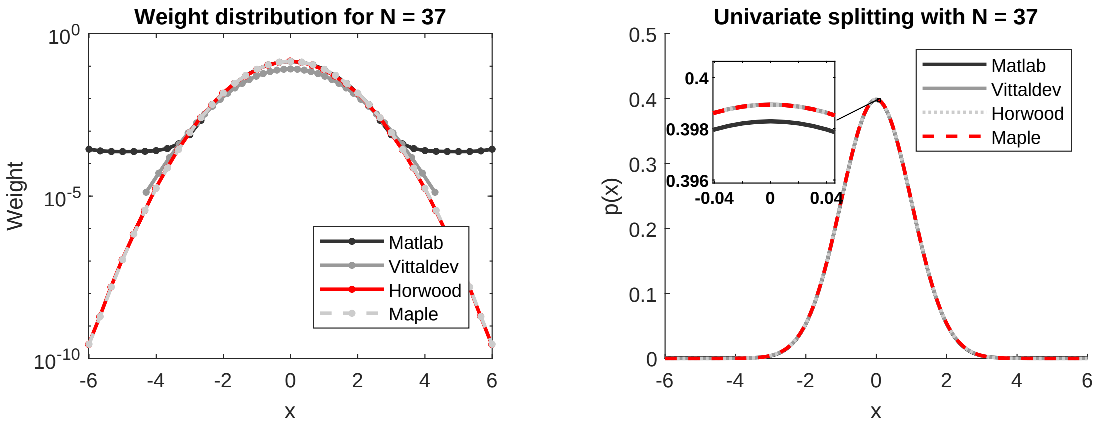
Optimisation methods comparison

Three methods were implemented in different software to solve the optimisation problem in Equation (5.17):

- Matlab's *fmincon* function: minimises constraint nonlinear multi-variable functions using the interior-point algorithm. Matlab uses double-precision floating point for this computation.
- Maple's *QPSolve* function: minimises a quadratic function subject to linear constraints using the active-set optimisation method. In this case, the floating point precision can be selected, allowing to compute using quadruple precision.
- Python's Parallel Global Multi-objective Optimiser library *pygmo*: This library offers multiple genetic algorithms to solve constrained and unconstrained optimisation problems. The Extended Ant Colony Optimisation (GACO) was selected as the best option considering the characteristics of the problem (single objective, constrained optimisation). The results of the algorithm were largely below the expected accuracy, and therefore the method was discarded and is not further analysed.

To analyse the results, the L_2 distance is again selected as the metric for accuracy of the split. Moreover, the resulting mixture (optimised weights) are compared to the results provided by Horwood et al. (2011) and Vittaldev and Russell (2016) for $N = 37$. Figure 5.3a illustrates the distribution of the optimised weights as a function of the mean location for each Gaussian Mixture Element. As observed, the reference univariate split from Vittaldev and Russell (2016) occupies the range $x \in [-4 \;\; 4]$ while Horwood et al. (2011) selects $x \in [-6 \;\; 6]$. As a consequence, the results from both methods are slightly different. Since the method will be applied with > 1000 GMEs, the wider range $x \in [-6 \;\; 6]$ is selected for our optimisation problem. Moreover, the weight distribution is strictly increasing for $x < 0$ and strictly decreasing for $x > 0$. In Maple, this can be set as an extra constraint that further improves the optimisation result.

It can be readily observed that both *fmincon* (Matlab) and *QPSolve* (Maple) provide an accurate result for the central weights. However, Matlab fails to correctly calculate the the extremes of the distribution, leading to errors in the approximation of the univariate distribution. This effect is visible in Figure 5.3b, where the reconstructed univariate probability density function is represented. As observed, the result from Maple perfectly match the univariate split used as reference while the result from Matlab shows a deviation that will affect the propagated probability distribution.



(a) Optimised weight distribution

(b) Reconstructed distribution from the GMM

Figure 5.3: Results of the univariate GMM splitting with N = 37 from the optimisation process in Matlab and Maple compared to the database from Vittaldev and Russell (2016), and Horwood et al. (2011).

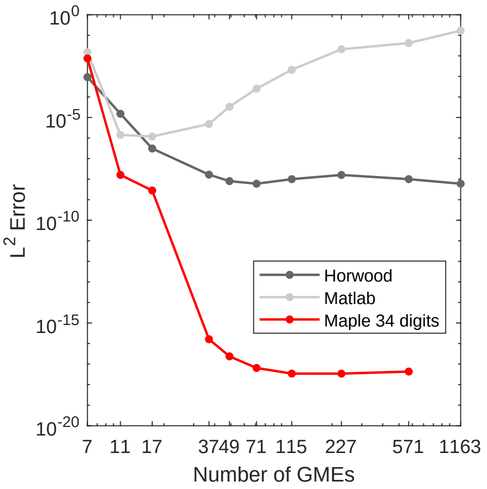


Figure 5.4: L^2 error between the univariate Gaussian distribution and its Gaussian sum approximation obtained using Matlab, Maple and the results from Horwood et al. (2011).

Results

For this research objective, it is expected to require a large number of GMEs, in the order of 1,000. Therefore, the method selected needs to perform well for a wide range of elements. To analyse this concept, the L_2 distance between the original one-dimensional univariate distribution and the Gaussian sum approximation is calculated for a selected number of GMEs and compared to the results provided by Horwood et al. (2011). As observed in Figure 5.4, for the reference data, the error decreases with the number of GMEs until it reaches a threshold value of 10^{-8}, which is considered acceptable. In the case of Matlab, the computational error derived from the limited floating point accuracy becomes more critical as the number of GMEs increases. As a result, a maximum accuracy of 10^{-6} is achieved for the scenario with 11 GMEs, and the method becomes invalid for higher number of elements. Finally, the computation with Maple using 34 digits provides the best results. Similarly to the reference results, the accuracy improves with the number of GMEs until it stabilises, in this case with an L_2 error of 10^{-18}. The computational time required for this calculation is significant (in the order of days) when > 500 elements are required. However, this optimisation problem is only solved once and the results are stored as a univariate splitting library for it use in any problem. Therefore, time is not a critical constraint in this scenario, and *QPSolve* from Maple is selected as the tool to compute the weights from the univariate split.

The univariate split is refined along the first coordinate to obtain its Gaussian sum approximation and then it is further refined along the remaining directions, as explained in Subsection 5.1.1. The result is a set

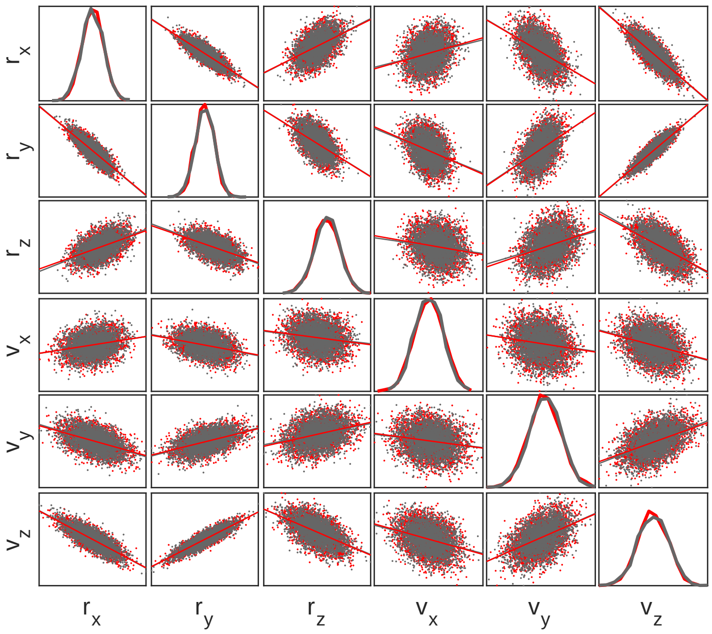


Figure 5.5: Correlation between state components derived from the Gaussian distribution (red) and Gaussian Mixture Model (grey).

of N weights and means, and a single covariance matrix that characterise the multivariate distribution as a Gaussian mixture. To illustrate this result, Figure 5.5 shows the histogram and correlation of two sets of data from a satellite state (position and velocity), drawn from the initial multivariate Gaussian distribution and its corresponding GMM. The data sets consist of 5,000 samples and it can already be observed that both distributions follow the same trend. For higher number of samples, both distributions agree, with a low error as expected from the low L_2 distance of $L_2 = 10^{-18}$ achieved.

5.1.3. Selection of required number of GMEs

The main benefit of modelling the initial distribution as a Gaussian Mixture Model is that the propagation of the individual elements through the nonlinear dynamics will correctly approximate the final non-Gaussian distribution. However, this implies that the individual GMEs are assumed to remain Gaussian during the propagation. The validity of this assumption will depend on the number of Gaussian Mixture elements and on the propagation time. In this section, the effect of these variables on the Gaussian hypothesis is analysed and related to the final error in the uncertainty distribution to select the optimum number of GMEs. This is done through two methods. First, the L_2 error between approximations for a different number of elements is calculated to evaluate the effect of adding more kernels on the final distribution. Second, several Gaussian hypothesis tests are performed to check the validity of the assumption. It must be noted that in any case, increasing the number of elements improves the accuracy of the propagated distribution with respect to the case where the uncertainty remains Gaussian. Therefore, although this section aims to verify that it is valid to assume that the individual GMEs remain Gaussian during the propagation, it is also valuable to prove that an accurate solution can be found when increasing the number of elements. This information allows to define a trade-off between computational time and accuracy.

Evaluation of L_2 error

To evaluate the effect of increasing the number of Gaussian mixture elements on the final distribution, the L_2 error is used, setting the distribution with 2001 elements as a reference. A nominal mission scenario is defined (labelled as NOM-CASE-1 hereafter) with the characteristics defined in Table 5.1. The initial state uncertainty is propagated during one and 30 orbital revolutions using the full dynamical model with perturbations. This process is repeated for 13 sets of GMMs with different number of elements ranging from seven to 2001. The computational time required to solve the optimisation problem increases quadratically with the number of elements. For this reason, the selected GMMs are not uniformly distributed, with a denser grid on the lower range of elements.

Table 5.1: Initial state orbital elements of NOM-CASE-1.

Element	a [km]	e [-]	i [deg]	ω [deg]	Ω [deg]	θ [deg]
Value	7178.137	0	60	75	32	40

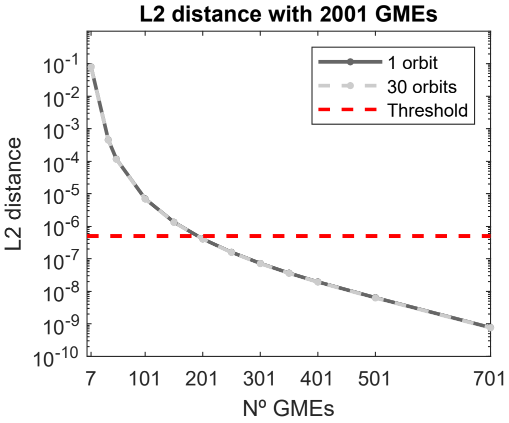


Figure 5.6: L_2 error between propagated distribution with 2001 Gaussian Mixture Elements and equivalent distributions for propagation of 1 orbit and 30 orbits.

Figure 5.6 illustrates the difference between the same initial distribution with different number of Gaussian kernels, with respect to the "reference" distribution with 2001 elements. As observed, the same trend is observed for the cases with one and 30 orbit propagation. This result simplifies the selection of the optimal number of elements, since it removes propagation time from the decision variables. The L_2 distance monotonously decreases with increasing number of elements, as expected. To select the optimal number of elements, considering the increase in computational time and the results from this analysis, a tolerance threshold is defined. Following the discussions by Horwood et al. (2011), a threshold L_2 distance of $5 \cdot 10^{-7}$ is selected. It is then concluded that the optimal number of Gaussian mixture elements is 201. However, it is important to note that selecting a higher number for cases where non-linearity in the propagation plays an important role is possible and will result in a higher accuracy. Similarly, a lower number can also be selected in cases where the available computational time is limited. In conclusion, the selection of the number of Gaussian Mixture elements is case dependent and this section is only meant to provide reference information on the effect of taking a given choice. Any Gaussian mixture model will represent an improvement with respect to assuming a Gaussian distribution.

Gaussian hypothesis test

Once an optimum number of 201 elements has been set, the assumption stating that the individual Gaussian Mixture Elements remain Gaussian during the propagation can be tested. To this end, several Gaussian hypothesis test algorithms are selected and applied different the NOM-CASE-1 scenario with different propagation times. Two types of hypothesis test are applied: single-variable and multi-variable. The former, only allow to test that a single variable is normally distributed. Therefore, the test is individually applied to every component of the propagated satellite state (i.e., every position and velocity component) and it is considered successful when every component passes the test for a given confidence level. To this end, three single-variable Gaussian hypothesis tests are applied: the Kolmogorov-Smirnov test (KS), the Anderson-Darling test (AD) and the Pearson's χ^2 test. The research performed by Razali and Yap (2011) concluded that the Anderson-Darling test provided the highest reliability amongst the chosen algorithms. This testing methodology assumes that the individual elements have to remain Gaussian, instead of testing the full multivariate distribution, which is not ideal. To solve this issue, two multivariate normality tests are applied, although these algorithms have been less researched and there is less information on their reliability. These are the Henze-Zirkler and the Roy normality tests (Henze and Zirkler, 1990; Silvey, 1959). All these tests have in common that they allow to reject the null hypothesis that the data does not come from a normal distribution, when the p-value overcomes a given threshold. More details on the mathematics behind these tests are provided in Subsection A.4.3. For this section, it the result of the test is just read as a pass/fail that determines whether the data can be assumed to be normally distributed.

Table 5.2: Gaussian hypothesis test on test case NOM-CASE-1. *Red:* failed test. *Green:* passed test.

nº GMEs	**201**					**2001**				
Propagation time (hours)	2	6	10	31	62	2	6	10	31	62
Single-variable hypothesis tests										
Kolmogorov-Smirnov	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
Anderson-Darling	🟩	🟩	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩
Pearson's chi-squared	🟩	🟩	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩
Multi-variable hypothesis										
Henze-Zirkler	🟩	🟩	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥
Roy	🟩	🟩	🟩	🟥	🟥	🟩	🟩	🟩	🟩	🟩

Table 5.2 presents the results of these tests according to a pass/fail criteria established relative to a p-value of 0.05. The tests are performed to individual GMEs coming from Gaussian mixtures of the same initial distribution and with 201 and 2001 elements respectively. For the Gaussian mixture element coming from a GMM with 201 elements, the tests reveal that the element remains Gaussian during six hours of propagation and start to deviate afterwards. For ten hours of propagation, only some of the state components remain Gaussian, and it is considered that the test is failed. The only exception is the Roy multi-variate normality test, which concludes that the GME remains Gaussian after 10 hours and starts to deviate afterwards. When increasing the model to 2001 GMEs, the elements remain Gaussian for longer propagation times. Excluding the result from the Kolmogorov-Smirnov test, which results in a failure for every scenario, the remaining tests allow to conclude that the individual elements remain Gaussian through the 62 hours propagation. Considering that AD presents higher reliability, the results from the KS method for this case are neglected.

This analysis provides an orientation on the validity of the Gaussian hypothesis assumption. The number of Gaussian elements considered optimum from the L_2 error analysis turn out to fail the normality test for propagation times longer than six hours. However, the computational burden of increasing the number of elements might turn complying with the Gaussian hypothesis into an unfeasible problem. Therefore, the recommended number of elements for a general case is 201, but selecting a larger number of elements might be beneficial in cases where the computational time is not an issue (e.g., problems with reduced collision interval that require high sensitivity and long propagation hours). This is further investigated in Subsection 8.3.2.

5.2. Taylor series

In this section, the theory of Taylor series and Taylor series integration is first introduced, with the help of Example 5.2. Then, the concept of optimal expansion order is discussed. This section provides the theory, and Section 5.3 provides the computational frame required to integrate the Taylor series expansion of the final state with respect to a deviation from the initial state. This corresponds to Step 2 from the road-map.

5.2.1. Theory

A Taylor expansion is a power series representation of a function as an infinite sum of terms formed by the function derivatives evaluated at a reference point, a. The series is named after Brook Taylor, who proposed the formulation in 1715. The Taylor series expansion of a function $f(x)$ which is infinitely differentiable at a real or complex number a is:

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n \tag{5.24}$$

where $n!$ is the factorial of n and $f^{(n)}(a)$ is the n^{th} derivative of f evaluated at a. When the Taylor expansion approximated by a finite number of terms, the truncation order k defines the maximum order of the derivatives evaluated. The resulting expression is a Taylor polynomial, commonly referred to as Taylor expansion of order k:

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \ldots + \frac{f^{(k)}(x(a))}{k!}(x-a)^k + R_k(x) \tag{5.25}$$

The remainder term $R_k(x)$ encloses the remaining terms and determines the approximation error. To apply this procedure to orbit propagation, consider the following ordinary differential equation (ODE) and initial value:

$$\frac{dx}{dt} = f(x(t)) \qquad x(t_0) = x_0 \tag{5.26}$$

The integration of this equation at time t is:

$$x(t) = x(t_0) + \int_{t_0}^{t} f(x(\tau))d\tau \tag{5.27}$$

If the Taylor polynomial defining $f(x)$ in Equation (5.25) is substituted in Equation (5.27). After analytical integration of the Taylor polynomial the state at time $t_1 = t_0 + \Delta t$ is:

$$x(t_0 + \Delta t) = x(t_0) + \Delta t f(x(t_0)) + \frac{1}{2!}\Delta t^2 f'(x(t_0)) + \ldots + \frac{1}{n!}\Delta t^n f^{(n-1)}(x(t_0)) + R_n \tag{5.28}$$

Example 5.2 - Taylor series

We consider the Taylor series expansion of $f(x) = \sin(3x) + x$ centred at $a = 0$ (MacLaurin series). To construct the series, we compute the derivatives of $f(x)$. The first five derivatives are given by:

$$f^{(0)}(x) = \sin(3x) + x \qquad f^{(0)}(0) = 0 \tag{5.29}$$

$$f^{(1)}(x) = 3\cos(3x) + 1 \qquad f^{(1)}(0) = 4 \tag{5.30}$$

$$f^{(2)}(x) = -9\sin(3x) \qquad f^{(2)}(0) = 0 \tag{5.31}$$

$$f^{(3)}(x) = -27\cos(3x) \qquad f^{(3)}(0) = -27 \tag{5.32}$$

$$f^{(4)}(x) = 81\sin(3x) \qquad f^{(4)}(0) = 0 \tag{5.33}$$

$$f^{(5)}(x) = 243\cos(3x) \qquad f^{(5)}(0) = 243 \tag{5.34}$$

The Taylor polynomial derived following Equation (5.25) is:

$$\sin(3x) + x = \frac{4}{1!}x + \frac{-27}{3!}x^3 + \frac{243}{5!}x^5 + R_6 = 4x - \frac{9}{2}x^3 + \frac{81}{40}x^5 + R_6 \tag{5.35}$$

Figure 5.7 illustrates the approximation of $f(x)$ by a MacLaurin series for different degrees of the Taylor polynomials. It is clear that the accuracy increases with increased expansion order. The 4^{th} order expansion diverges at approximately $x = \pm 0.7$, while the 16^{th} expansion correctly approximates the function for approximately two periods.

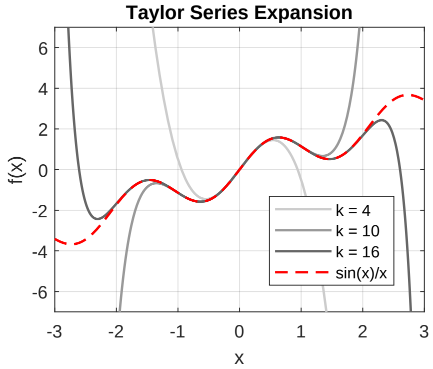


Figure 5.7: Taylor series approximation of order $k = 4, 10, 16$ of $f(x) = \sin(3x) + x$

5.2.2. Optimal expansion order

The expansion order determines the accuracy of the propagated uncertainty, and therefore of the overall collision probability calculation. An expansion order of one corresponds to the linear covariance propagation performed by traditional methods. By using a differential algebra technique, an arbitrary value can be set for the Taylor expansion order, at the expense of computational cost. For this application, it has been found that a expansion order of four is the maximum that could be used for operational collision risk estimation.

$$\begin{tikzcd}
a,b \in R \arrow[r, "\mathcal{T}"] \arrow[d, "\times"'] & \bar{a},\bar{b} \in FP \arrow[d, "\otimes"] \\
a \times b \arrow[r, "\mathcal{T}"'] & \bar{a} \otimes \bar{b}
\end{tikzcd}
\qquad
\begin{tikzcd}
f,g \arrow[r, "\mathcal{T}"] \arrow[d, "\times"'] & F,G \arrow[d, "\otimes"] \\
f \times g \arrow[r, "\mathcal{T}"'] & F \otimes G
\end{tikzcd}$$

Figure 5.8: Number representation in a computer environment (left) and function representation in a differential algebra framework (right) (Di Lizia et al., 2008)

Increasing the expansion order does not only affect the computational load of the propagation in differential algebra, but also the load of reconstructing the propagated covariance.

The effect of expansion order in computational load and accuracy is studied as part of the sensitivity analysis in Chapter 8. It is found that for most applications an expansion order of two or three should be used. By selecting a expansion order of three it is ensured that the accuracy with increasing lead time will not be compromised due to this parameter.

5.3. Differential algebra

The differential algebra approach was first studied by Berz (1999) to solve analytic problems with algebraic means. The result of this study was the code implementation of a differential algebra package: COSY INFINITY, which was applied in the field of particle beam physics. The basic principle behind the DA approach is to bring the treatment of functions to the computer environment in a similar way as the treatment of numbers.

The traditional approach to deal with functions in a computer environment is to evaluate the functions at specific points. Following this method, the differentiation of a function following a Taylor series expansion has a high cost and inaccuracies due to the multiple evaluations of the function derivatives and the accumulation of rounding error. With the DA technique, the objective is to extract more information from the function than just its numerical evaluation. To understand this concept, the analogy between the treatment of real numbers and the treatment of functions is introduced and illustrated in Figure 5.8.

Consider two real numbers a and b and their floating point representation $\bar{a}, \bar{b}$. We want to to apply an arbitrary mathematical operation such as addition or multiplication represented by ×. The first option is to perform this operation to the set of real numbers a and b and convert the result of this operation to its floating point representation (in Figure 5.8, left illustrated in the path from upper-left to lower-left and lower-left to lower-right. The second option is to transform the real numbers to their floating point representation $\bar{a}, \bar{b}$ and apply an adjoined operation to ×, ⊗ in the set of floating points. This is illustrated in the upper-left to upper-right and upper-right to lower-right path from Figure 5.8. In the end, the diagram commutes and the resultant operation from both options is the same. Now le us consider a similar scenario, with two functions f and g. In this simile, the transformation of real numbers to their floating point representation is analogue to the extraction of the Taylor series expansion of the functions F and G. If we want to apply an operation to these functions in a computational environment, two options can be considered. The first option, to perform the operation to f and g and then transform the result to its Taylor series expansion. The second option, to convert both functions to their Taylor series representation F, G and operate on them on the space of Taylor polynomials. Both options are equivalent and illustrated in the right diagram of Figure 5.8. In this approach, the Taylor coefficients of a function and its evaluation can be computed with a fixed effort. The implementation of the DA framework evolutioned to include complex operations on functions such as differentiation and integration.

The application of Taylor series expansion is applied in the DA framework as follows. To this end let us consider the ODE defined in Equation (5.26). To integrate the expression in a DA framework, the initial condition x_0 is initialised as a DA variable: $[x_0] = x_0 + \delta x_0$, where x_0 is the reference point of the expansion and δx_0 is the perturbation. To explain the integration process, let us consider the Taylor expansion of order one in Euler's integration scheme following $x_1 = x_0 + \Delta t \cdot f(x_0)$. If the variable is initialised in the DA framework:

$$[x_1] = [x_0] + \Delta t \cdot f([x_0]) \tag{5.36}$$

where the output of the first step, $[x_1]$ is the Taylor expansion of the flow of the ODE in x_0 for $t = t_1$. The procedure is applied repeatedly until the last integration step is reached and the resultant solution is represented by a Taylor polynomial map: $x_{tf} = \mathcal{T}_k(\delta x_0)$, where k is the order of the Taylor series expansion. The evaluation of the Taylor polynomial directly supplies the solution x_{tf} corresponding to the displaced initial condition δx_0. The same polynomial is evaluated to calculate the final state from any initial deviation.

Example 5.3 - Differential Algebra

To better understand the computational behaviour behind differential algebra, let us consider the computation of the following function for a perturbation x with respect to the central value of $a = 0$:

$$f(x) = \sin(x) + e^x + \log(1-x) \tag{5.37}$$

In traditional floating point computation, to evaluate this expression at a given point $x = 3$, the individual operations are computed and the floating point numbers are added with the defined floating point accuracy:

$$f(0.3) = \sin(0.3) + e^{0.3} + \ln(0.7) = 0.29552 + 1.34985 - 0.35667 = 1.28870$$

However, in differential algebra the operations are not performed using floating points, but using the Taylor expansion of the expressions. In this case, the Taylor expansion of the individual components is given by:

$$\begin{aligned}
\sin x &= \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots \\
e^x &= \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots \\
\log(1-x) &= -\sum_{n=1}^{\infty} \frac{x^n}{n}
\end{aligned}$$

Therefore, the Taylor expansion for $f(x)$ is:

$$f(x) = \sum_{n=0}^{\infty} \left(\frac{(-1)^n}{(2n+1)!} x^{2n+1} + \frac{x^n}{n!} \right) = -\sum_{n=1}^{\infty} \frac{x^n}{n} \tag{5.38}$$

In differential algebra, this computation is internally done, with the expansion coefficients of each individual operation up to the expansion order, k. For an example with $k = 5$, the expression obtained in this computational algebra method would be :

$$f(x) \approx 1 + x + \frac{x^3}{3} + \left(\frac{1}{4!} - \frac{1}{4}\right) x^4 + \left(\frac{2}{5!} - \frac{1}{5}\right) x^5 = 1 \cdot x^0 + 1 \cdot x^1 + 0 \cdot x^2 - 0.\widehat{3} \cdot x^3 - 0.208\widehat{3} \cdot x^4 - 0.18\widehat{3} \cdot x^5$$

which evaluated at $x = 3$ gives:

$$f(3) \approx 1.288867$$

The result from the differential algebra software would be the Taylor expansion coefficients of the solution, corresponding to each element:

$$f(x) = \sum_{0}^{k} c_i \cdot x^i \tag{5.39}$$

For this example, the differential algebra software would provide as an output a file with the Taylor expansion coefficients summarised in Table 5.3.

Table 5.3: Taylor expansion coefficients of Polar to Cartesian conversion

Coefficient	$\mathbf{c_0}$	$\mathbf{c_1}$	$\mathbf{c_2}$	$\mathbf{c_3}$	$\mathbf{c_4}$	$\mathbf{c_5}$
Value	1	1	0	$-0.\widehat{3}$	$-0.208\widehat{3}$	$-0.18\widehat{3}$

5.4. PDF reconstruction

The process followed to reconstruct the mean and covariance of the propagated state $\boldsymbol{y}$ is based on the derivation provided by (Valli et al., 2013), and corresponds to Step 3 from the road-map. The final state is expressed as the Taylor series expansion with respect to the deviations $\delta\boldsymbol{x}$ of the initial state $\boldsymbol{x}$:

$$[\boldsymbol{y}] = \boldsymbol{f}([\boldsymbol{x}]) = \sum_{p_1+\cdots+p_n \leq k} \boldsymbol{c}_{p_1 \ldots p_n} \cdot \delta x_1^{p_1} \cdots \delta x_n^{p_n} \tag{5.40}$$

where $\boldsymbol{c}_{p_1 \ldots p_n}$ are the Taylor coefficients of the polynomial expansion. The mean of the probability distribution of a random variable $\boldsymbol{x}$ is defined by its expectation ($\mu = E\{\boldsymbol{x}\}$). To compute the expectation it is required to use the Hafnian operator, which is a challenge to computationally optimise. The numerical details of this operator and its computation are given in Subsection 7.2.1. The linearity property of the expectation operator satisfies that for N random variables X_i and constants a_i:

$$\mathrm{E}\left\{\sum_{i=1}^{N} a_i X_i\right\} = \sum_{i=1}^{N} a_i \mathrm{E}\{X_i\} \tag{5.41}$$

Applying this property to the mean of the final state expressed by its Taylor series expansion yields:

$$\boldsymbol{\mu}_i = E\{[\boldsymbol{y}_i]\} = E\left\{\sum_{p_1+\cdots+p_n \leq k} \boldsymbol{c}_{p_1 \ldots p_n} \cdot \delta x_1^{p_1} \cdots \delta x_n^{p_n}\right\} = \sum_{p_1+\cdots+p_n \leq k} \boldsymbol{c}_{i,p_1 \ldots p_n} E\{\delta x_1^{p_1} \cdots \delta x_n^{p_n}\} \tag{5.42}$$

Where $\boldsymbol{\mu}_i$ is the mean of the i^{th} component of the final state, $\boldsymbol{y}_i$. The covariance of two random variables, $\boldsymbol{x}, \boldsymbol{y}$ is defined by $\boldsymbol{P}_{xy} = E\{(\boldsymbol{x} - E\{\boldsymbol{x}\})(\boldsymbol{y} - E\{\boldsymbol{y}\})\}$. By using the linearity property of expectations, this expression can be reduced to:

$$\begin{aligned} \boldsymbol{P}_{xy} &= \mathrm{E}\{\boldsymbol{xy} - \boldsymbol{x}\mathrm{E}\{\boldsymbol{y}\} - \boldsymbol{y}\mathrm{E}\{\boldsymbol{x}\} + \mathrm{E}\{\boldsymbol{x}\}\mathrm{E}\{\boldsymbol{y}\}\} \\ &= \mathrm{E}\{\boldsymbol{xy}\} - \mathrm{E}\{\boldsymbol{x}\}\mathrm{E}\{\boldsymbol{x}\} - \mathrm{E}\{\boldsymbol{x}\}\mathrm{E}\{\boldsymbol{y}\} + \mathrm{E}\{\boldsymbol{x}\}\mathrm{E}\{\boldsymbol{y}\} \\ &= \mathrm{E}\{\boldsymbol{xy}\} - \mathrm{E}\{\boldsymbol{x}\}\mathrm{E}\{\boldsymbol{y}\} \end{aligned} \tag{5.43}$$

Substituting for the covariance $\boldsymbol{P}_{ij}$ between the final state components $[\boldsymbol{y}_i], [\boldsymbol{y}_j]$, and using the expression of the mean from Equation (5.42) results in:

$$\boldsymbol{P}_{ij} = E\left\{\left(\sum_{p_1+\cdots+p_n \leq k} \boldsymbol{c}_{i,p_1 \ldots p_n} \cdot \delta x_1^{p_1} \cdots \delta x_n^{p_n}\right)\left(\sum_{q_1+\cdots+q_n \leq k} \boldsymbol{c}_{j,q_1 \ldots q_n} \cdot \delta x_1^{q_1} \cdots \delta x_n^{q_n}\right)\right\} - \mu_i \mu_j \tag{5.44}$$

Applying the distributivity property of the summation operator, which satisfies that $\left(\sum_{i=0}^{n} a_i\right)\left(\sum_{j=0}^{n} b_j\right) = \sum_{i=0}^{n}\sum_{j=0}^{n} a_i b_j$, the expression can be rearranged as:

$$\boldsymbol{P}_{ij} = E\left\{\sum_{\substack{p_1+\cdots+p_n \leq k \\ q_1+\cdots+q_n \leq k}} \boldsymbol{c}_{i,p_1 \ldots p_n} \boldsymbol{c}_{j,q_1 \ldots q_n} \cdot \delta x_1^{p_1+q_1} \cdots \delta x_n^{p_n+q_n}\right\} - \mu_i \mu_j \tag{5.45}$$

Finally, applying again the linearity of the expectation operator, the propagated covariance matrix can be calculated as:

$$\boldsymbol{P}_{ij} = \left(\sum_{\substack{p_1+\cdots+p_n \leq k \\ q_1+\cdots+q_n \leq k}} \boldsymbol{c}_{i,p_1 \ldots p_n} \boldsymbol{c}_{j,q_1 \ldots q_n} \cdot E\left\{\delta x_1^{p_1+q_1} \cdots \delta x_n^{p_n+q_n}\right\}\right) - \mu_i \mu_j \tag{5.46}$$

The following example demonstrates the output and result of the covariance propagation process for a simple coordinate conversion.

Example 5.4 - Polar to Cartesian coordinate conversion

To exemplify the propagation of covariance, the two-dimensional example of a conversion from Polar to Cartesian components is introduced. For a given Polar state $\boldsymbol{x} = [x_1\ x_2] = [r\ \theta]$, the equivalent state in Cartesian components $\boldsymbol{y} = [y_1\ y_2]$ is defined as:

$$y_1 = x_1 \cdot \cos y_1$$
$$y_2 = x_1 \cdot \sin y_1$$

For a polar vector $\boldsymbol{x} = [1\ \pi/3]$, the nominal transformation to Cartesian components is $\boldsymbol{y} = [0.5\ \sqrt{3}/2]$. Consider that the measurement in Polar coordinates presents an uncertainty with a standard deviation of 0.02 me in x_1 and $\frac{15\pi}{180}$ rad in x_2. The conversion of Polar to Cartesian components in this example with Taylor expansion order of four can be expressed in the format from Equation (5.40). This operation yields as an output the Taylor expansion of the converted state with respect to a deviation of the nominal initial state displayed in Table 5.4.

Table 5.4: Taylor expansion coefficients of Polar to Cartesian conversion

Coefficient	**Order**	**Exponents**		**Coefficient**	**Order**	**Exponents**	
y_1		δx_1	δx_2	y_1		δx_1	δx_2
$5.0 \cdot 10^{-1}$	0	0	0	$8.6602 \cdot 10^{-1}$	0	0	0
$-8.6602 \cdot 10^{-1}$	1	0	1	$5.0 \cdot 10^{-1}$	1	0	1
$5.0 \cdot 10^{-1}$	1	1	0	$8.6602 \cdot 10^{-1}$	1	1	0
$-2.5 \cdot 10^{-1}$	2	0	2	$-4.3301 \cdot 10^{-1}$	2	0	2
$-8.6602 \cdot 10^{-1}$	2	1	1	$5.0 \cdot 10^{-1}$	2	1	1
$1.4433 \cdot 10^{-1}$	3	0	3	$-8.3333 \cdot 10^{-2}$	3	0	3
$-2.5 \cdot 10^{-1}$	3	1	2	$-4.3301 \cdot 10^{-1}$	3	1	2
$2.0833 \cdot 10^{-2}$	4	0	4	$3.6084 \cdot 10^{-2}$	4	0	4
$1.4433 \cdot 10^{-1}$	4	1	3	$-8.333 \cdot 10^{-2}$	4	1	3

To compute the mean and standard deviation of the converted variable, the Taylor expansion coefficients and the initial covariance are input to Equations (5.42) and (5.46). The result of the conversion of uncertainty from Polar to Cartesian coordinates assuming that the final distribution is also Gaussian is observed in Figure 5.9. In this case, the mean and covariance of the overall final distribution assuming Gaussian uncertainty is correctly computed. However, the final distribution cannot be approximated as a Gaussian, due to the non-linearities in the propagation. To solve this issue, either a larger expansion order is required (which is not compatible with the computation of the Hafnian) or a Gaussian Mixture model can be used. The second option is selected for this work. In the propagation of uncertainty for satellite states, the effect of non-linearities in the dynamics are smaller and an accurate propagation of uncertainty can be achieved with Taylor expansion order of two to three.

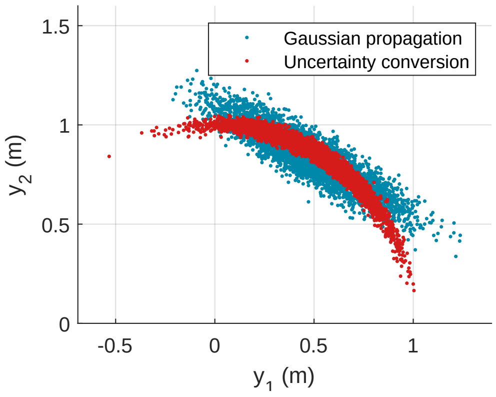


Figure 5.9: Conversion of uncertainty in Polar coordinates to Cartesian coordinates for 5000 samples, analytically and using differential algebra with a single Gaussian distribution

6

Collision probability calculation methods

The hybrid Gaussian Mixture Model and Differential Algebra method introduced so far propagates the initial uncertainty of the satellite state in time. Although this technique informs to an extent about the position of the objects at the time when a collision is suspected, it is not sufficient to quantify the collision risk. As discussed in Chapter 2, collision probability is the preferred parameter to evaluate the risk of a fatal encounter. This chapter is dedicated to theoretically introducing and discussing the implementation of the collision probability calculation methods selected to work together with the DA-GMM uncertainty propagation, which corresponds to Step 4 in the road-map. The conventional methods used to calculate the collision probability and their main assumptions are introduced in Section 2.2. By understanding the limitations of these techniques, the hot-spots for improvement are identified and taken into account in the development of collision probability calculation algorithms. A summary of the main characteristics of conventional methods for collision probability calculation is presented in Section 6.1. In this work, two methods of collision probability calculation are developed. First, the direct method relying on the calculation and time integration of the collision probability rate is introduced in Section 6.2 based on the work by Coppola and McAdams (2012) and extended to a mixture of Gaussian distributions. While this method overcomes important limitations from the conventional methods, it still assumes that the bodies are spherical. Since the effect of this assumption is not always negligible, a variation of the method that considering a complex geometrical model of the body is developed here, applied to the International Space Station, and discussed in Section 6.3.

6.1. Conventional methods

The conventional methods for collision probability, divided into short-term and long-term encounters have been discussed in Section 2.2, where their assumptions and limitations are explained. Table 6.1 summarises these techniques and their characteristics. The column short-term assumptions refers to assumptions A_1 to A_4 introduced in Subsection 2.2.1. Short-term extension refers to the long-term methods that perform the two-dimensional integration for the short-term assumption followed by a one-dimensional integration over the relative velocity. The relative velocity column refers to methods that apply assumptions on the relative velocity (either that it is non-zero or that it does not present any uncertainty). The body shape column specifies that types of encounter geometries that can be considered in any method and the last three columns indicate whether the method is included in the main collision-detection software. As observed, each collision probability calculation software contains at least one short-term and one long-term encounter algorithm, to adequately adapt to different scenarios. While the short-term methods are efficient in computing the collision probability for these encounters, the methods for long-term collision probability are limited in terms of computational time and accuracy. The assumptions on relative velocity cause the methods to fail in certain encounter geometries. Moreover, the limitation to spherical shapes can greatly impact the risk estimate. From these methods, the one developed by Coppola and McAdams (2012) presents the largest advantages, with the limitations that it assumes a Gaussian uncertainty and assumes a spherical shape. The first limitation is solved by extending the method for a mixture of Gaussian distributions, as derived by DeMars et al. (2014). The second, is solved by applying the multi-sphere method, developed and presented in this work.

Table 6.1: Summary of characteristics of existing collision probability calculation methods. Body shape can be a sphere (S), parallelepiped (P) or any cross section using the method of equivalent cross-section area (MECSA).

Reference	**Short-term assumptions**	**Short-term extension**	**Relative velocity**	**Body shape**	**NASA CARA**	**ESA CORAM**	**STK**
Short-term encounter methods							
Foster and Estes, 1992	X		X	S			
Patera, 2001	X		X	P, S		X	
Alfano, 2005	X		X	S			
Chan, 2008	X		X	MECSA	X		X
Long-term encounter methods							
Patera, 2003		X	X	P, S		X	
Chan, 2004		X	X	MECSA		X	
Alfano, 2006		X	X	P, S			X
Alfano, 2007		X	X	P, S			X
Coppola and McAdams, 2012				S	X		

6.2. Time integration method

This section introduces the time integration method for collision probability calculation. The algorithm was originally developed by Coppola and McAdams (2012) to compute the collision probability between objects with Gaussian uncertainty. This method offers two main improvements with respect to the conventional techniques described. First, it takes into account the uncertainty in the object's velocity, therefore dropping assumption A2. This is particularly important for the case of a collision space debris. While the velocity uncertainty of a satellite with the current orbit determination techniques, this does not apply to space debris. The effect of velocity uncertainty becomes non-negligible in this case, especially for long encounters. Second, the method is applicable for short and long terms encounters, and the encounter time becomes a free parameter. This expands the applicability of this collision risk assessment method, not only to GEO encounters but also to specific cases of LEO encounters, which require a long integration time. The time integration method was extended by DeMars et al. (2014) to include include Gaussian Mixture Models. The computational load of the model increases quadratically with the number of GMEs, which can become an issue for long-term encounters. However, even if the optimal number of GMEs cannot be included in the collision probability calculation, it already represents an improvement with respect to the Gaussian propagation scenario. In this section, the direct method for collision probability is mathematically described. In Subsection 6.2.2, a simplified example is provided to visualise the implementation and results of the method. Finally, the free parameters that can tune the collision probability calculation are identified and analysed in Subsection 6.2.3.

6.2.1. Theory

The direct method is mathematically based on the time integration of the collision probability rate over the encounter time. The collision probability rate calculated from the influx and outflux of the uncertainty distribution over the hard-body sphere. This concept is similar to Reynold's transport theorem, although applied to the artificial variable of collision probability rate at a given time.

The probability that a collision occurs by time t, $P_c(t)$ can be obtained by the time integration of the collision probability rate $p_c(t)$ assuming that at time $t = 0$ the collision probability is zero:

$$P_c(t) = \int_0^t p_c(\tau) d\tau \tag{6.1}$$

Based on the direct method of collision probability introduced by Coppola and McAdams (2012), and Akella and Alfriend (2000), the collision probability rate is given by:

$$p_c(t) = \int_S \int_{v_n \leq 0} |v_n| \, p\left(\mathbf{x}_t\right) d\mathbf{v}_t dS \tag{6.2}$$

where $v_n = \boldsymbol{v}_t \cdot \hat{\boldsymbol{n}}$ is the relative velocity ($\boldsymbol{v}_t$) normal to the surface area element dS, S is the area of the hard ball and $p(x_t)$ is the joint probability density function of the relative position $\boldsymbol{r}(t) = \boldsymbol{r}_t$ and relative velocity $\boldsymbol{v}(t) = \boldsymbol{v}_t$, with the satellite state defined as $x_t = [\boldsymbol{r}(t)\ \boldsymbol{v}(t)]$. The expression can be rearranged using that $p(x_t) = p(\boldsymbol{v}(t)|\boldsymbol{r}(t))p(\boldsymbol{r}(t))$ as:

$$p_c(t) = \int_S \left[\int_{v_n \le 0} |v_n|\, p\left(\mathbf{v}_t \mid \mathbf{r}_t\right) d\mathbf{v}_t\right] p\left(\mathbf{r}_t\right) dS \tag{6.3}$$

The problem reduces to compute the joint PDF, $p(x_t)$, which following the derivation from DeMars et al. (2014), simplifies for a GMM to:

$$p\left(\mathbf{x}_t\right) = \sum_{i=1}^{N_a}\sum_{j=1}^{N_b} w_{a,t}^{(i)} w_{b,t}^{(j)} p_g\left(\mathbf{x}_t; \boldsymbol{\mu}_t^{(ij)}, \Sigma_t^{(ij)}\right) \tag{6.4}$$

where N_a and N_b are the number of GMEs describing the PDF of satellites A and B respectively, $w_{a,t}^{(i)}$, $w_{a,t}^{(i)}$ are the weight of the i^{th} GME of satellite A and B respectively at time t. $\mu_t^{(ij)} = m_{a,t}^{(i)} - m_{b,t}^{(j)}$ and $\Sigma_t^{(ij)} = \mathbf{P}_{a,t}^{(i)} + \mathbf{P}_{b,t}^{(j)}$ where $m_{a,t}^{(i)}, m_{b,t}^{(i)}, \mathbf{P}_{a,t}^{(i)}, \mathbf{P}_{b,t}^{(i)}$ are the mean and covariance matrices of the i^{th} GME of satellites A and B respectively at time t. To simplify the calculation of Equation (6.2), the joint mean and covariance matrix $\mu_t^{(ij)}, \Sigma_t^{(ij)}$ can be decomposed into the position and velocity components as:

$$\boldsymbol{\mu}_t^{(ij)} = \begin{bmatrix} \boldsymbol{\mu}_{r,t}^{(ij)} \\ \boldsymbol{\mu}_{v,t}^{(ij)} \end{bmatrix} \quad \text{and} \quad \Sigma_t^{(ij)} = \begin{bmatrix} \Sigma_{r,t}^{(ij)} & \Sigma_{rv,t}^{(ij)} \\ \Sigma_{vr,t}^{(ij)} & \Sigma_{v,t}^{(ij)} \end{bmatrix} \tag{6.5}$$

which substituted into Equation (6.4) and then into Equation (6.2) becomes:

$$p_c(t) = \sum_{i=1}^{L_a}\sum_{j=1}^{L_b} w_{a,t}^{(i)} w_{b,t}^{(j)} \int_S p_g\left(\mathbf{r}_t; \boldsymbol{\mu}_{r,t}^{(ij)}, \boldsymbol{\Sigma}_{r,t}^{(ij)}\right) v\left(\mathbf{r}_t\right) dS, \tag{6.6}$$

$$v\left(\mathbf{r}_t\right) = \int_{v_n^{(ij)} \le 0} \left|v_n^{(ij)}\right| p_g\left(\mathbf{v}_t'; \left(\boldsymbol{\mu}_{v,t}^{(ij)}\right)', \left(\Sigma_{v,t}^{(ij)}\right)'\right) d\mathbf{v}_t' \tag{6.7}$$

with $v_n^{(ij)} = \mathbf{v}_t' \cdot \hat{\mathbf{n}} + \hat{\mathbf{n}}^T \Sigma_{vr,t}^{(ij)} \left(\Sigma_{r,t}^{(ij)}\right)^{-1} \mathbf{r}_t$. Assuming that both bodies are spherical, with a joint hard-body radius R, the collision probability rate is simplified to:

$$p_c(t) = R^2 \sum_{i=1}^{N_a}\sum_{j=1}^{N_b} w_{a,t}^{(i)} w_{b,t}^{(j)} \int_0^{2\pi}\int_{-\pi/2}^{\pi/2} p_g\left(R\hat{\mathbf{n}}; \boldsymbol{\mu}_{r,t}^{(ij)}, \Sigma_{r,t}^{(ij)}\right) v(\hat{\mathbf{n}}) \cos\theta\, d\theta\, d\phi \tag{6.8}$$

where θ and ϕ are the elevation and azimuth angles respectively. For a spherical hard-ball, the integration over velocity can be performed analytically, and $v(\hat{\mathbf{n}})$ becomes:

$$v(\hat{\mathbf{n}}) = \frac{\sigma(\hat{\mathbf{n}})}{\sqrt{2\pi}} \exp\left\{-\frac{v_0^2(\hat{\mathbf{n}})}{2\sigma^2(\hat{\mathbf{n}})}\right\} - \frac{v_0(\hat{\mathbf{n}})}{2}\left[1 - \operatorname{erf}\left\{\frac{v_0(\hat{\mathbf{n}})}{\sigma(\hat{\mathbf{n}})\sqrt{2}}\right\}\right], \tag{6.9}$$

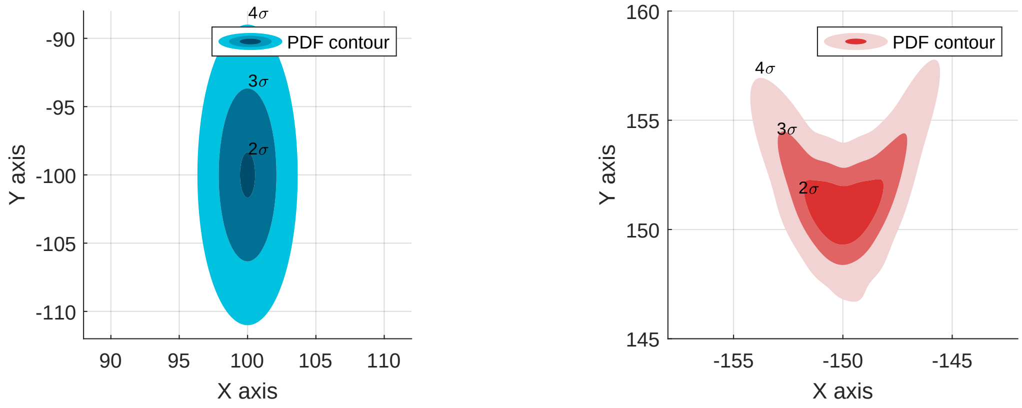
where $v_0(\hat{\mathbf{n}})$ and $\sigma^2(\hat{\mathbf{n}})$ are:

$$\begin{aligned} v_0(\hat{\mathbf{n}}) &= \hat{\mathbf{n}}^T \left[\boldsymbol{\mu}_{v,t}^{(ij)} + \boldsymbol{\Sigma}_{vr,t}^{(ij)} \left(\boldsymbol{\Sigma}_{r,t}^{(ij)}\right)^{-1} \left(R\hat{\mathbf{n}} - \boldsymbol{\mu}_{r,t}^{(ij)}\right)\right] \\ \sigma^2(\hat{\mathbf{n}}) &= \hat{\mathbf{n}}^T \left(\Sigma_{v,t}^{(ij)} - \Sigma_{vr,t}^{(ij)} \left(\Sigma_{r,t}^{(ij)}\right)^{-1} \Sigma_{rv,t}^{(ij)}\right) \hat{\mathbf{n}} \end{aligned} \tag{6.10}$$

This expression can then be incorporated into Equation (6.8) and the remaining step is to perform the flux integral over the surface of the sphere. This integration cannot be conducted analytically, and numerical quadrature techniques are required instead. The most common numerical approximation is Lebedev's quadrature (Lebedev, 1976), which is further discussed in Subsection 7.2.2. This method selects a set of N_k quadrature points on the surface of the sphere and assigns weights to approximate the integral by a weighted summation of the function evaluation. These quadrature points are represented by unit vectors on the unit sphere, $\hat{\mathbf{n}}_k$. The resulting expression to evaluate the collision probability rate as a function of time is:

$$p_c(t) = R^2 \sum_{i=1}^{N_a}\sum_{j=1}^{N_b} w_{a,t}^{(i)} w_{b,t}^{(j)} \sum_{k=1}^{N_k} w_k p_g\left(R\hat{\mathbf{n}}_k; \boldsymbol{\mu}_{r,t}^{(ij)}, \Sigma_{r,t}^{(ij)}\right) v\left(\hat{\mathbf{n}}_k\right) \tag{6.11}$$

This result is then numerically integrated over the encounter time interval (which can be found by specifying a threshold value) to obtain the collision probability.



(a) Car A

(b) Car B

Figure 6.1: Initial position uncertainty

6.2.2. Implementation

This section aims to introduce the implementation of the direct method for collision probability through a simplified two-dimensional example representing a car crash. This example allows to visualise the geometry of the encounter and better understand the results of the risk assessment. The temporal evolution of the collision probability rate and the total collision probability are then presented.

Car crash

Consider the scenario of two cars moving at a constant speed on a two-dimensional grid in perpendicular trajectories, such that their nominal path will collide at a certain time. However, there is an uncertainty in the measurement of the initial position and velocity. In this case, it cannot be stated that a collision will occur, and this introduces the need to calculate the collision probability. For this example, car A presents a Gaussian uncertainty and car B is modelled as a Gaussian Mixture Model with five elements for demonstration purposes.

Figure 6.4 illustrates the uncertainty contours of the initial position of both cars. Car A is located at a mean position of $x_{0,A} = 100$ m, $y_{0,A} = -100$ m and car B at $x_{0,B} = -150$ m, $y_{0,B} = -150$ m. Knowing that the cars move in the $+y$ and positive $+x$ directions respectively, the nominal collision would occur at $x_{TCA} = 100$ m, $y_{TCA} = 150$ m, at $t = 180$ s. The temporal evolution of the uncertainty is observed in Figure 6.2. It can be noted that the size of the covariance contour increases with time. This is due to the uncertainty in the velocity, which then affects the uncertainty in the position. If there was no velocity uncertainty, the shape of the contours would remain constant. From this figure, it can readily be understood that although fast, the collision does not occur instantaneously when there is uncertainty involved. It can be observed that there are several time frames that overlap and therefore collision probability must be studied as a function of time.

The results of applying the direct method of collision probability to this example, assuming a hard-body radius of 30 cm are displayed in Figure 6.3. The size of the hard-ball radius shall be at least one order of magnitude smaller than the position standard deviation, to avoid approximation errors in the Lebedev quadrature. This is not an issue in the case of satellite conjunction assessment, but it forces to set a small radius for this car crash example (which then becomes a remote control car crash example). In Figure 6.3a, it is observed that the maximum collision probability rate is found at $t = 180$ s, which coincides with the expected collision time. However, considering only that instant for the collision probability would bring a large error. It is required to integrate the collision probability along the complete encounter interval to obtain an accurate estimate of the collision risk, as observed in the evolution of the collision risk with time illustrated in Figure 6.3a.

6.2.3. Method parameters

In this section, the free parameters from the direct method of collision probability calculation are identified and discussed. Although the effect of these parameters on the final result are heavily dependent on the case under study, general trends and guidelines can be defined.

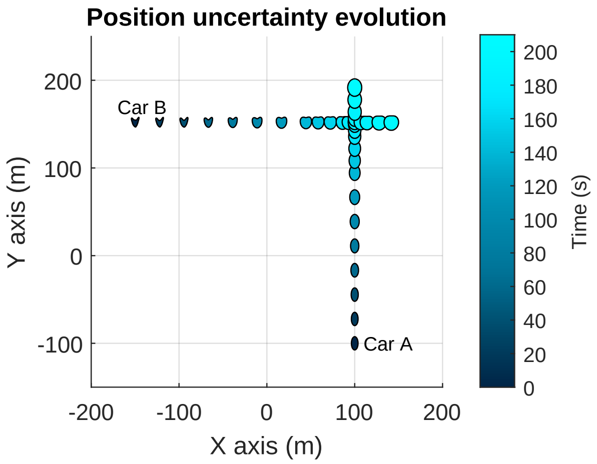
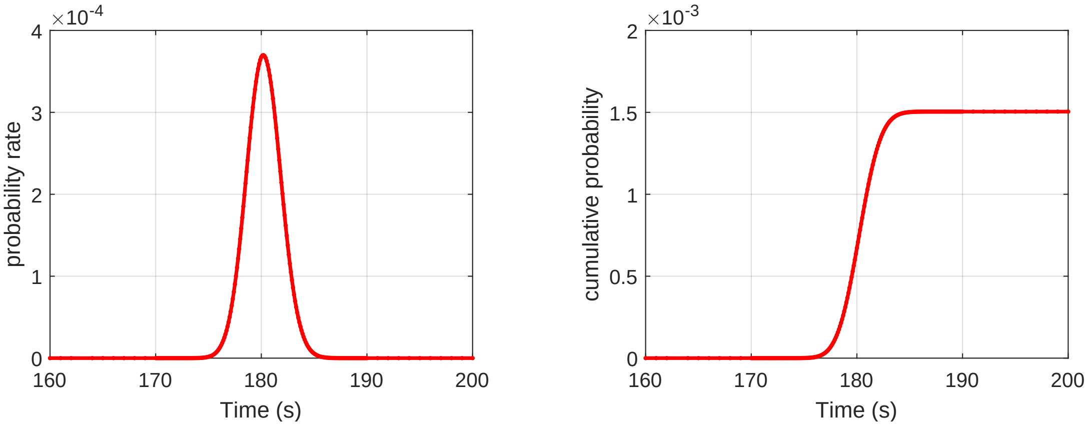


Figure 6.2: Uncertainty evolution of cars A and B.



(a) Collision probability rate

(b) Cumulative collision probability

Figure 6.3: Time evolution of collision risk in car crash example.

Integration time

The direct method for collision probability relies on the time integration of the collision probability rate. Therefore, selecting the correct time period is crucial to correctly estimate the collision probability. In regular scenarios where both satellites have different orbits and a punctual collision in time is detected, the collision probability rate will monotonously increase until reaching the maximum value at time of closest approach and then decrease, as observed in Figure 6.3a. In this case, selecting the integration time is straight forward, and only requires setting a threshold value for the minimum probability rate that should be considered. For this analysis, a threshold value of 10^{-12} is recommended. Another option is to select the integration time based on a threshold value for the nominal separation distance. This allows to decide on the integration period before performing the simulations, therefore minimising the use of unnecessary computational load. Once the uncertainty propagation is done and the direct method is applied, it can be checked whether the probability rate threshold is violated, and perform extra simulations, if required.

However, this is not always the case and there are scenarios, where the collision risk is not continuous and the integration time cannot be simply set by a threshold. The criteria to select the integration time will then depend on multiple case-dependent parameters such as tracking station passes and available computational load.

Time step

The required time step is also a case-dependent variable that will be selected as a function of the shape of the probability rate curve and the integration time. For very short term encounters, a shorter time step will be required. For short term encounters (typically intervals of less than five seconds) a time step of 0.1 seconds is

Table 6.2: Collision probability with change in time step with 1454 quadrature points.

Time step (s)	1	10	100	500
Collision probability (%)	0.24443	0.2442	0.2432	0.2281
% Error w.r.t $\Delta t = 1$ s	-	0.09	0.5	6.6

Table 6.3: Collision probability with change in quadrature points with $\Delta t = 10$ s.

Quadrature points	50	590	1000	1454	3074	5810
Collision probability (%)	0.2766552	0.244930	0.244217	0.2442129	0.2442169	0.2442170
% Error w.r.t 5810 points	13.282	0.291	0.0405	$6.5777 \cdot 10^{-4}$	$2.89 \cdot 10^{-4}$	-

recommended by Coppola and McAdams (2012). In these scenarios, using a time step of 1 second can lead to errors of 10% in the collision probability. However, in long term encounters the collision probability is spread over a longer time period, which allows to select longer time steps in the order of 10-60 seconds. As a general rule, it is recommended to select 100-300 integration points along the integration time span and refine the time step in fast changing areas if required.

To evaluate the effect of time step on a nominal satellite mission, the co-located satellite example that will be discussed in Section 9.1, is considered. In this case, the encounter time is very long, due to the fact that both satellites follow the same distribution. Moreover, the collision rate does not present narrow spikes, which suggests that small step sizes (< 1 s) are not required. Table 6.2 presents the collision probability evaluating the time integral with different time steps ranging from 1 to 500 seconds. From the analysis of the error with respect to the 1 second computation, it can be concluded that for this scenario, up to 100 seconds would be a valid time step, resulting in a 0.5% difference. When increasing to 500 seconds, the difference increases to 6.6 %, which supposes a great increase compared to the change in computational load, but still provides a result within the same order of magnitude.

Hard-ball radius

The hard-ball radius will depend on the geometrical specifications of the problem (i.e., the size of the satellites). However, due to the approximation required to calculate the surface integration along the sphere with Lebedev's method, the method will fail when the sphere is too big compared to the position uncertainty. This is not usually a problem due to the characteristic magnitude of the satellite uncertainty with respect to the satellite size. However, it must still be taken into account when applying the method. For safety, the sphere radius must be at least one order of magnitude smaller than the standard deviation of the position uncertainty.

Quadrature points

The number of quadrature points selected will affect the accuracy of the flow integration over the surface of the hard-body sphere and therefore the overall accuracy of the method. For a more detailed explanation and visualisation of Lebedev's quadrature, refer to **??**. In general, the required number of quadrature points will depend on the relative size of the sphere with respect to the position uncertainty. Moreover, the number of quadrature points linearly influences the computational load of the collision probability calculation. The available computational time will depend on the encounter time, the number of GMEs, and the Taylor expansion order. The selection of the optimal number of quadrature points would be the result of a case specific trade-off between computational load and radius to standard deviation ratio. This analysis is too specific and would only be useful in extreme scenarios. Instead, a general number of quadrature points is proposed for the standard scenario, which can be increased if needed. Table 6.3 presents the results with a range of quadrature points from 50 to 5819, and the relative error with respect to 5810 points. From these results it can be observed that for this scenario, an accurate computation can be achieved with approximately 1000 points.

6.3. Multi-sphere method

While the assumption of a spherical hard-body for the collision is valid for generic satellites, it introduces a large error in the case of the International Space Station. In this particular case, collision probability calculation is of foremost importance, due to the safety requirement for the astronauts inhabiting the spacecraft.

In nominal scenarios the collision probability is calculated considering a safety zone of 2 km (local vertical) and 25 km × 25 km (local horizontal), and a manoeuvre is applied when the detected collision risk is greater than 10^{-5}. However, there are specific scenarios when applying a collision avoidance manoeuvre can pose a higher risk than the space debris itself. This is the case when astronauts are performing extra-vehicular activities and when there is a docking operation. In the unlikely event that these situations occur, the upper-bound estimate for the collision probability does not provide enough information to correctly assess the risk. Using a more detailed model of the vehicle offers a solution to this issue. Considering the case of a short-term encounter, with all the associated assumptions, Chan (2008) developed the method of equivalent cross-section area, in which the cross-section of the ISS is divided into basic geometrical shapes which are converted to circles with the same area. The limitations of this method are that it only serves for short-term encounters and that it only considers one orientation of the satellite with respect to the debris trajectory (i.e., only one cross-section). Following a similar principle, the multi-sphere method is introduced, aiming to apply the theory of the direct method for collision probability to a three-dimensional model of the vehicle formed by a mesh of spheres. To multi-sphere method is developed under the following assumptions:

A-MSPH-1 The effect of shadowing between components can be neglected.

A-MSPH-2 The orientation of the vehicle with respect to the orbital plane is constant.

A-MSPH-3 The body frame (*B*-frame) is aligned with the radial-transverse-normal frame (*RTN*-frame).

These assumptions can be avoided by developing specific geometrical models for each scenario and applying frame transformations to account for the orientation at every time. In this section, the mathematical description of the multi-sphere model for collision probability calculation is introduced in Subsection 6.3.1. In Subsection 6.3.2, the geometrical specifications of the ISS are discussed and the creation of the multi-sphere model is explained. The implementation of the method, illustrated with a basic example is presented in Subsection 6.3.3.

6.3.1. Theory

During the mathematical derivation of the direct method for collision probability, no assumptions are made on the shape of the object until Equation (6.8), where the integration for a sphere is presented. The fact that the surface integration cannot be performed analytically limits the applicability of the method to generic shapes. While numerical quadrature techniques for the sphere have been widely studied, the extension to other shapes is limited. Reeger et al. (2016) developed a method to extend the quadrature to arbitrary smooth, closed surfaces. However, the method does not work for complex shapes such as the volume of the ISS. In response to these limitations, a model is devised based on the additive property of the direct method for collision probability. Considering a hypothetical case where the hard-body volume was two adjacent spheres, and neglecting the effect of shadowing, the collision probability would be the sum of the individual probability for each sphere. If the hard-body was formed by two intersecting spheres, the collision probability would be calculated by the surface integration over the resulting surface. The benefit of using a quadrature technique is that the quadrature points in the boundary of the resulting volume can be readily identified, allowing to compute the surface integral over the desired portion of the spheres. This concept can be generalised to more complex structures, allowing to adapt the direct method of collision probability to any shape. The following steps are required to implement the multi-sphere method.

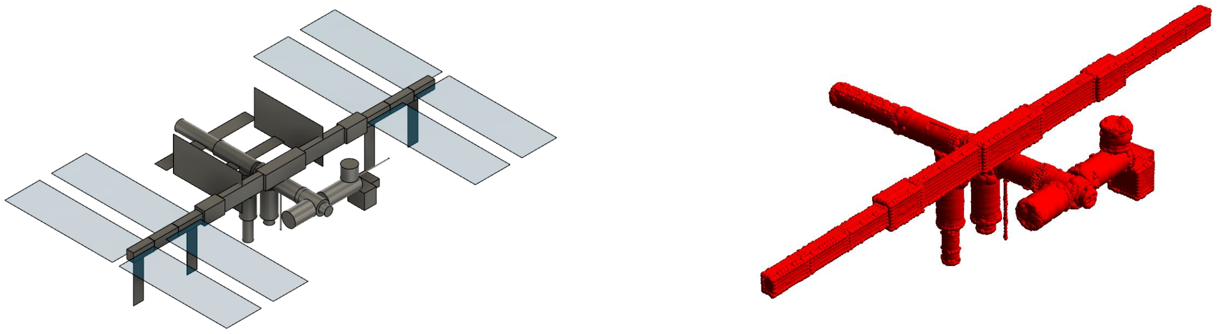
1. Select the number of spheres N_{sph} to fit in the volume to create a sphere mesh. This number will depend on the complexity of the shape and the computational load allowed. For this work, the ISS is modelled with a total of 60,000 spheres.

2. Create a sphere-mesh representation, with variable radius. The output is be the centre and radius of each sphere.

3. Sample N_k quadrature points from each sphere.

4. From the mesh of $N_{sph} \times N_k$ points, extract the external N_{bnd} boundary points which define the external surface of the volume.

5. Identify the associated Lebedev weights ($w_{bnd,k}$) and sphere radius (R_k) of the boundary points.



(a) Simplified ISS model in Autodesk Fusion

(b) Sphere tree of the ISS body with 50,000 spheres

Figure 6.4: Sphere modelling of the ISS

6. Calculate the collision probability rate with the adapted version of Equation (6.11):

$$p_c(t) = \sum_{i=1}^{N_a}\sum_{j=1}^{N_b} w_{a,t}^{(i)} w_{b,t}^{(j)} \sum_{k=1}^{N_{bnd}} R_k^2 w_{bnd,k} p_g\left(R_k \hat{\mathbf{n}}_k; \boldsymbol{\mu}_{r,k,t}^{(ij)}, \Sigma_{r,t}^{(ij)}\right) \nu\left(\hat{\mathbf{n}}_k, R_k\right) \tag{6.12}$$

   where $\boldsymbol{\mu}_{r,k,t}^{(ij)}$ is the relative distance between the secondary object and the centre of each Lebedev sphere, in the *RTN* reference frame.

7. Numerically integrate $p_c(t)$ over the encounter time to obtain the total collision probability rate.

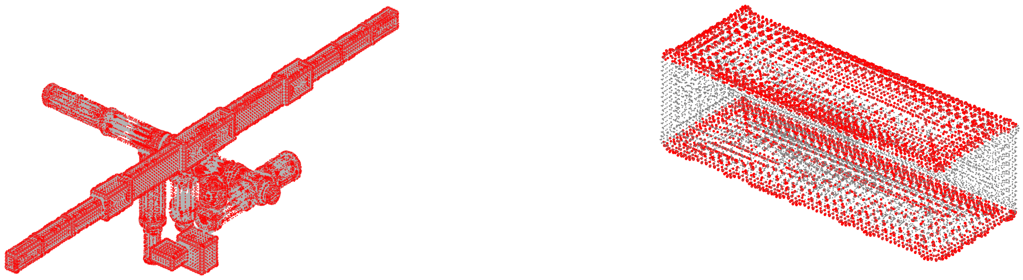
6.3.2. Geometry specifications

To apply the multi-sphere method, it is required to obtain a set of Lebedev points related to the spheres forming the external boundary of the model. To achieve this, a three-dimensional model of the vehicle under study is required (in this case, the ISS). Although multiple models of the ISS are publicly available, they are ill-suited for this purpose. This is due to the high degree of detail in small components that impede the generation of a sphere mesh. To solve this issue, a simplified three-dimensional model of the ISS is created in Autodesk Fusion, as illustrated in Figure 6.4a. This model is based on the component division devised by Chan (2008) for the method of equivalent cross-section area. It consists of 52 elements represented by simple geometrical shapes (parallelepipeds and cylinders) that approximate the ISS at one of its possible configurations.

To generate the sphere mesh, the software spheretree[1] is used, which includes adaptive medial axis algorithms for the construction of so-called sphere trees, developed by Bradshaw and O'Sullivan (2004). It was readily identified that the meshing of the solar pannels would pose a problem due to their large surface related to the thickness. As a consequence, a large number of spheres would be required to correctly fit the shape of the panel. This was computationally unfeasible, both for the computation of the mesh and for the later calculation of the collision probability. To solve this issue, the solar panels are studied separately. The body of the ISS is modelled as a sphere tree using 10,000 sample points and 50,000 spheres. The resulting sphere mesh is illustrated in Figure 6.4b. Despite the complex shape of the body, the sphere mesh produces an accurate replica of the real volume of the ISS.

For the solar panels, creating the sphere tree from the thin plates is unfeasible. Each panel has a surface area of 420 m^2 and a thickness of 10 cm can be assumed. Based on this, more than 50,000 spheres would be required to approximate the shape of each solar panel, adding up to a total of $> 420{,}000$ spheres for the set of eight solar panels. The root of the problem is the low thickness of the solar panel, which requires small spheres to fit inside the model. However, small spheres are not a requirement to approximate a flat surface. Therefore, an alternative solution is proposed, which consists in creating a thicker version of the solar panels (10 m thick) to fit the spheres. Once the (substantially smaller) sphere tree is created, the boundaries are extracted and the points forming the top and bottom faces of the parallelepiped are selected. The upper and lower boundaries are then shifted to recreate the shape of the solar panel with a thickness of 10 cm. Figure 6.5b illustrates this concept. In grey, all the quadrature points sampled from the sphere mesh in the thick

[1] Available at : `https://github.com/mlund/spheretree`. Date accessed: 01/12/2021



(a) ISS body

(b) ISS solar panels

Figure 6.5: Boundary extraction from multi-sphere model with 50 quadrature points per sphere. *Grey:* internal points. *Red:* boundary points.

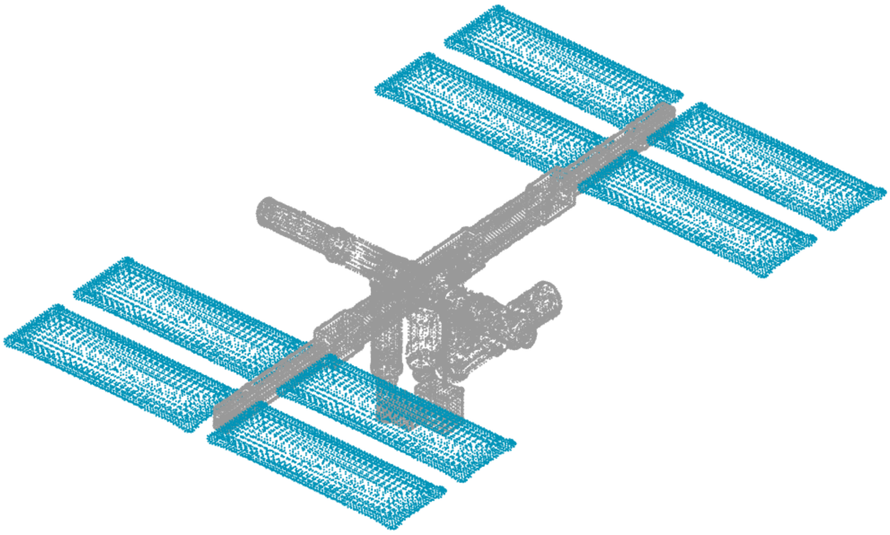
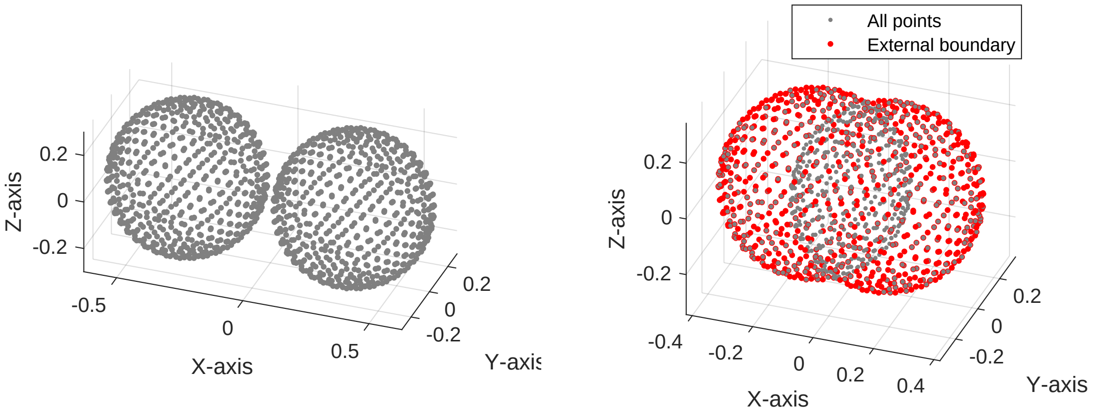


Figure 6.6: ISS representation with boundary points from multi-sphere model. *Grey:* main body formed by 50,000 spheres. *Blue:* solar panels formed by 5,000 spheres each.

parallelepiped are represented. These points fill the entire volume. A boundary extraction algorithm is then used to identify the external points, from which the upper and lower 5 cm slices are selected, illustrated in red. Essentially, these represent the quadrature points forming the external boundary of an equivalent parallelepiped with 10 cm thickness. In Figure 6.5a, the same concept is illustrated for the body of the ISS, with the internal points sampled from the sphere mesh in grey and the points forming the external boundary in red. Once all the boundary points are identified, the body and the solar panels can be combined as illustrated in Figure 6.6. Note that this figure does not represent a mesh point of the ISS. It represents a set of quadrature points, with associated weights, that can be used to numerically integrate the flow of a given variable over the surface of the ISS.

6.3.3. Implementation

To illustrate the implementation of the multi-sphere model for collision probability, the two-dimensional car crash example is again used. In this case, a simplified version of the multi-sphere model formed by only two spheres will be used, to easily exemplify the how the method works. The single sphere model can be easily understood, as the sphere represents the volume than can be occupied by the satellites for a collision to occur, and the probability is computed as the integration of the collision probability rate as a flux over



(a) Quadrature points from separated spheres.

(b) Quadrature points from intersecting spheres.

Figure 6.7: Simplified multi-sphere model with two spheres.

Table 6.4: Collision probability for single-sphere to multi-sphere example.

Geometry	$\mathbf{P_c}$
Single sphere	0.0015
Two separated spheres	0.0030
Two intersecting spheres	0.0019

this collision volume. To gradually understand the multi-sphere model, let us first consider a case where the collision volume can be defined by two spheres, separated by a distance of 66 cm from centre to centre as illustrated in Figure 6.7a. In such a case, the collision probability will be approximately double than the one in case of the single sphere. It is not exactly double because neither of the spheres are located at the centre anymore and therefore the distance between the uncertainty flux and the sphere is different. The collision probability in this case is equal to adding the individual probability of each sphere.

To understand the concept of the multi-sphere model, let us now assume that the combined shape of the two objects equals two intersecting sphere, with the centre of the spheres separated by 24 cm. To calculate the collision probability for this type of body, the flux of collision probability rate must be integrated along the surface of this volume. There is not an analytical way to perform this integration, and therefore quadrature methods must be used. However, there is no quadrature method that directly integrates the flux over this shape. Making use of the concept of Lebedev's quadrature for a single sphere, and the additive property of collision probability, the method can be extended to account for two intersecting spheres. Such a geometry is illustrated in Figure 6.7b. In grey, the quadrature points defining both spheres can be observed. In red, only the external quadrature points that form the surface of this new volume are visualised. By performing the surface integration considering only the external quadrature points and its associated weights, the flux over this shape can be computed. For this calculation, each sphere is shifted to the centre and the mean of each GME is moved accordingly. The collision probability for the car crash example described in Subsection 6.2.2 with the three geometries described is summarised in Table 6.4. As observed, the collision probability that results from considering two spheres is double than when considering only one. This result was expected since the separation of the spheres from the centre is very small compared to the magnitude of the position uncertainty.

When considering the intersection of two spheres, the collision probability represents 63.3% of the probability associated to the separated double sphere. To analyse whether this is feasible, two properties are inspected, the volume and the surface area of the volume. For the intersection of spheres, the combined volume represents 78.4% of the total volume of two spheres, while the surface area is 70%. Therefore, the magnitudes observed are reasonable. To understand why the exact value represents a lower percentage with respect to he surface area, it is recalled that the geometry is complex and the relation between flux integral and surface area is not linear, since it also depends on the orientation of the surface with respect to the flow.

7
Simulation, verification and validation

In this chapter, the simulation environment and verification and validation of the DA-GMM method are discussed. Section 7.1 covers the simulation process, by discussing the external software used and developed in this work, detailing the architectural design and the interaction between blocks of code and explaining the procedure followed for code optimisation and CPU time reduction. In Section 7.2, the numerical methods required for this work are introduced. These are the covariance propagation with the Hafnian operator and the Lebedev quadrature. Following, the verification process for the trajectory integration in the differential algebra software is treated in Section 7.3. This process includes unit, integration and system tests that ensure a correct propagation of the satellite orbits. In Section 7.4, the verification of the remaining pieces of code that form the pre- and post- processing of the method is explained. Finally, the guide for validation is presented in Section 7.5.

7.1. Simulation

This section covers the elements of the simulator developed to estimate the probability of collision between bodies in space. First, the external software is described and the main functionalities that have been used and developed are summarised. The architectural design is then introduced, distinguishing between three main elements of the simulator: pre-processing, propagation and post-processing. The latter is explained in more detail due to its complexity and relevance within the work. Finally, considering the importance of reducing the computational load in uncertainty propagation problems, a section is dedicated to software optimisation, covering the evaluation and steps applied o improve the performance of the code.

7.1.1. External software

To conduct the collision probability estimation using the DA-GMM method, three types of software are required. First, a differential algebra software that allows to obtain the Taylor series expansion of the flow by computing with Taylor expansion coefficients instead of floating points.The Differential Algebra Computational Engine (DACE), a differential algebra library developed in C++ and created by Dinamica srl. for ESA is selected for this purpose (Rasotto et al., 2016). This tool is open source and can be freely combined with other libraries. Second, an astrodynamics propagation tool is required to verify code developed and compare the results with traditional Monte Carlo techniques. The TU Delft Astrodynamics Toolbox (Tudat [1]. Accessed: 26-12-2021) is selected. It consists of a set of libraries developed in C++ that contain verified functions to numerically propagate trajectories and provide access to the required environmental and numerical models. Finally, a computer environment such as Matlab or Python is required for the pre- and post-processing of the propagation results and the calculation of collision probability. In this case, Matlab is selected as the software of choice, and Maple is used to compute the univariate splitting as explained in Section 5.1. In addition to these, a software is required to develop the multi-sphere ISS model, for which spheretree is selected and described in this section.

[1] Tudat software documentation available in: `https://tudat.tudelft.nl/`

DACE

The Differential Algebra Computational Toolbox (DACE) is a set of libraries coded in C++ that allow to operate with the Taylor series approximation of functions instead of using floating point numbers. The main advantage of using this tool with respect to other differential algebra software such as COSY Infinity is that it can be combined with other C++ libraries and interfaces. This is particularly helpful to develop the dynamical model of the satellite, including the libraries that allow to compute the perturbing accelerations. DACE has been validated with existing DA software with excellent results. In the recent years, this software has been the selected choice for most aerospace applications. Massari et al. (2017) relied on this software to propagate uncertainty in the unperturbed, two-body problem and Lunghi et al. (2018) used it for re-entry uncertainty propagation in Mars landing.

Within DACE, several functionalities have been developed to propagate the uncertainty of space objects in a perturbed dynamical model that includes spherical harmonics up to degree and order six, aerodynamic acceleration and third-body acceleration from the Sun and the Moon. Moreover, two existing libraries have been adapted. The following external libraries have been included in the DA-GMM application:

NRLMSISE-00 From the existing atmosphere models of Earth, the NRLMSISE-00 is selected and adapted to DACE. The model developed by Picone et al. (2002) is provided as C source code [2] and has been adapted to C++ by a C++ wrapper provided by Tudat.

SPICE The position of the Sun and the Moon required to compute the third-body acceleration is extracted from the SPICE toolkit, provided as a C library [3] and adapted to C++ by a C++ wrapper provided by Tudat.

Boost The Boost C++ library is required as part of the C++ wrappers that read the Spice and atmospheric interfaces, and provides multiple functionalities in the fields of linear algebra operations, pseudo-random number generation and unit testing.

Eigen The Eigen C++ library is required as part of the C++ wrappers that read the Spice and atmospheric interfaces to perform linear algebra operations.

The following main functionalities have been developed within the DA-GMM application to propagate the uncertainty of space objects.

Frame transformation To compute the total acceleration in the $I-$frame, several frame transformations are used, as explained in Section 3.2. The differential algebra computation requires all functions to be analytical, and therefore all frame transformations are individually coded as template functions that can be accessed with DA variables.

Integrator To integrate the trajectory, the RK78 integrator is used. Its code is adapted to access the integrator with all the required model and body parameters.

Legendre polynomials The Legendre polynomials are part of the computation of the spherical harmonics. While Legendre polynomials are usually recursively computed using floating points, this is not valid for the DA computation. The analytical expression of each normalised Legendre polynomial and its derivative up to degree and order six is coded and included in the DA software.

Spherical harmonics acceleration An individual function computing the spherical harmonics acceleration for Earth up to degree and order six.

Aerodynamic acceleration An individual function computing the aerodynamic acceleration assuming that the body frame is aligned with the trajectory frame.

Third body acceleration An individual function computing the third body perturbation from the Sun and the Moon central gravitational acceleration.

Total acceleration Function computing the total acceleration of the space object, including: spherical harmonics of Earth up to degree and order six, aerodynamic acceleration and central gravitational acceleration from the Sun and Moon.

[2] NRLMSISE-00 library available at: `https://github.com/magnific0/nrlmsise-00`; Accessed: 19-12-2021

[3] SPICE library available at: `https://naif.jpl.nasa.gov/naif/toolkit.html`; Accessed: 19-12-2021

Properties class All the dependent variables defining the body (e.g., mass, area, C_D, etc.) and the environment (e.g., start epoch) are collected in the properties class, which can be written and accessed from the main code.

Tudat

To conduct the code verification using Monte Carlo simulations, the TU Delft Astrodynamics Toolbox (Tudat) is selected [4]. Tudat is a set of C++ libraries, which combined offer powerful simulation capabilities for a wide set of astrodynamics problems, including the missions under study. This software has been developed by members of the Aerospace Engineering faculty at the Technical University of Delft, and is under continuous development. The propagation algorithms and libraries included are already verified, and therefore can be used as reference to test the output of the DA software. The main function of this software is to simulate the reference trajectories that will allow to verify the results of the DA-GMM method with a Monte Carlo approach.

Spheretree

To apply the multi-sphere method of collision probability, it is required to obtain a three dimensional mesh of the object under study, created with spheres. Fitting spheres to a body has been studied in the field of video-games and interactive simulations to study whether two objects intersect. Bradshaw and O'Sullivan (2003) studied this problem and combined several sphere-fitting algorithms into the *spheretree* software [5].

As input to the software, a three-dimensional model of the object to be fitted with spheres in .obj format is provided. Eight different algorithms are then available to perform the fitting, for which the branching factor of the sphere tree and the minimum number of spheres used can be selected. All the algorithms were tested with different settings based on the examples provided by the Spheretree software and tuning the values. Based on the results of this process, the Hubbard's medial axis algorithm algorithm is chosen. This is not the fastest method but provided the best approximation with less spheres outside the ISS core. Since the operation only needs to be performed once, computational time is not an issue. This algorithm produces tight fitting sets of spheres with high accuracy.

7.1.2. Architectural design

This section summarises the architectural design of the DA-GMM method to propagate the uncertainty in position an velocity of space objects and compute the probability of collision between two bodies. First, the architecture of the complete method is explained, keeping the uncertainty propagation in the differential algebra software as a black box. Second, the architecture of the propagation code block performed in DACE is summarised.

Software architecture

The DACE-GMM method of collision probability can be divided into five code blocks, which are illustrated in Figure 7.1. The inputs to the method are highlighted in yellow, and correspond to the settings of the algorithm (number of GMEs, Taylor expansion order, integration step) and the characteristics of the space objects (mass, area, encounter period, etc.). The blue boxes correspond to actions that only need to be performed once, and their output can be stored for later usage. The green boxes correspond to output of the method (both intermediate and final).

The first code block consists of the splitting of the initial distribution into a Gaussian mixture. This code block is extremely computationally efficient and outputs the Gaussian Mixture Model of the initial state, which is then an input to the uncertainty propagation in differential algebra. For every Gaussian mixture element, the state is propagated in the DA software and the output is the Taylor series expansion of each element at time t_i with respect to the initial time t_0. This information is then provided to the uncertainty reconstruction code block, which uses the Hafnian operator to obtain the mean and covariance of each Gaussian mixture element at time t_i. The combination of all these elements conforms the Gaussian mixture model of the satellite state at the final time. This information, together with the quadrature points (which are simply the Lebedev points in the case of the single-sphere model and the extracted boundary points in case of the multi-sphere model), is provided to the collision probability rate code block. Using the definition of surface flow integration, the collision probability rate at time t_i is calculated. When this process is finalised for all the points in the encounter time interval, the collision probability rate is integrated to obtain the collision probability.

[4] Tudat software documentation available at: `https://tudat.tudelft.nl/`. Accessed: 26-12-2021

[5] Available at: `https://github.com/mlund/spheretree`. Accessed: 31-12-2021

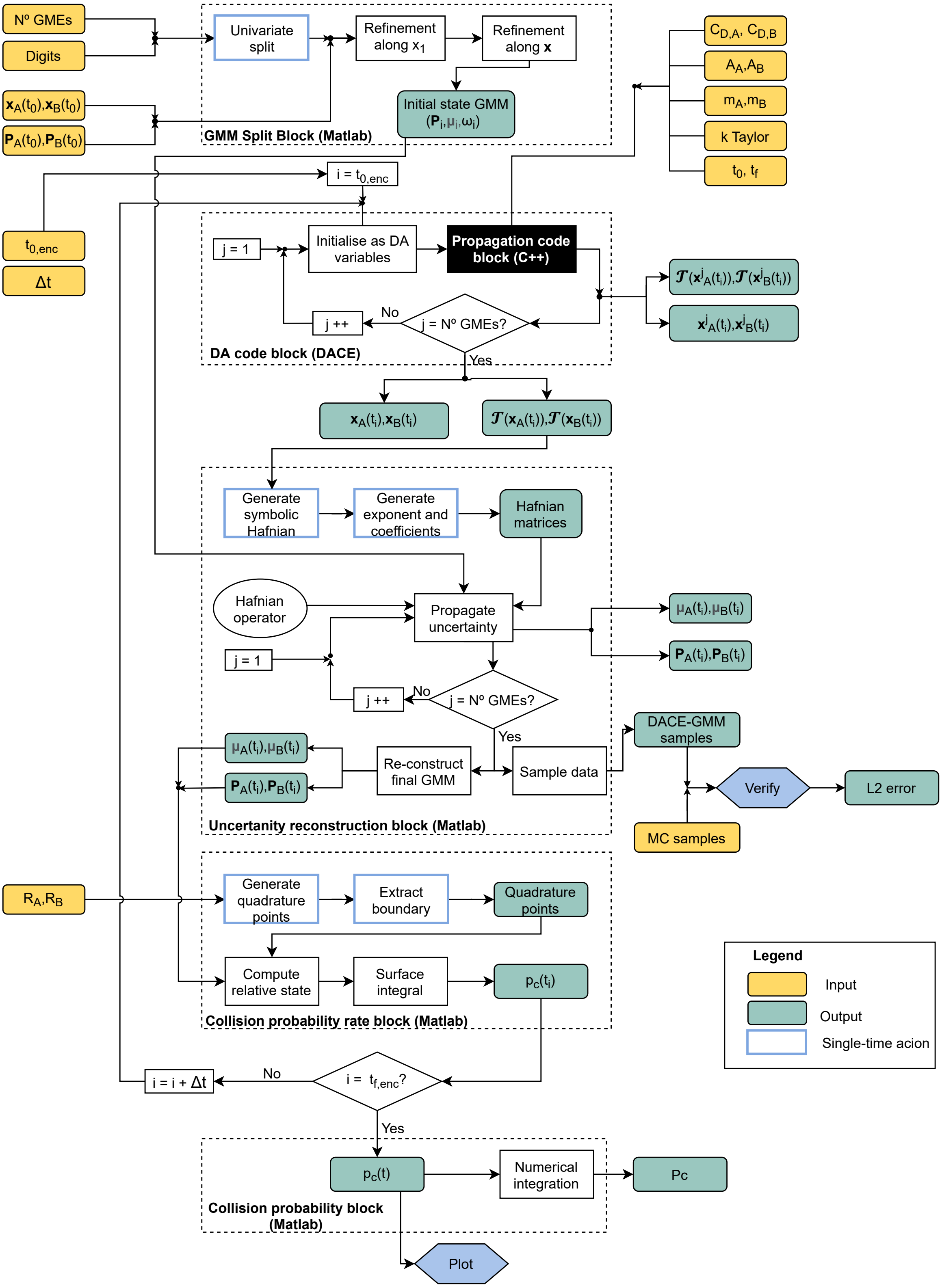

Nº GMEs
Digits
$\mathbf{x}_A(t_0)$,$\mathbf{x}_B(t_0)$
$\mathbf{P}_A(t_0)$,$\mathbf{P}_B(t_0)$
Univariate split
Refinement along x_1
Refinement along $\mathbf{x}$
Initial state GMM ($\mathbf{P}_i$,μ_i,ω_i)
GMM Split Block (Matlab)
$C_{D,A}$, $C_{D,B}$
A_A,A_B
m_A,m_B
k Taylor
t_0, t_f
i = $t_{0,enc}$
$t_{0,enc}$
Δt
j = 1
Initialise as DA variables
Propagation code block (C++)
$\mathcal{T}(\mathbf{x}^j_A(t_i))$,$\mathcal{T}(\mathbf{x}^j_B(t_i))$
$\mathbf{x}^j_A(t_i)$,$\mathbf{x}^j_B(t_i)$
No
j ++
j = Nº GMEs?
DA code block (DACE)
Yes
$\mathbf{x}_A(t_i)$,$\mathbf{x}_B(t_i)$
$\mathcal{T}(\mathbf{x}_A(t_i))$,$\mathcal{T}(\mathbf{x}_B(t_i))$
Generate symbolic Hafnian
Generate exponent and coefficients
Hafnian matrices
Hafnian operator
Propagate uncertainty
$\mu_A(t_i)$,$\mu_B(t_i)$
$\mathbf{P}_A(t_i)$,$\mathbf{P}_B(t_i)$
j = 1
j ++
No
j = Nº GMEs?
Yes
$\mu_A(t_i)$,$\mu_B(t_i)$
$\mathbf{P}_A(t_i)$,$\mathbf{P}_B(t_i)$
Re-construct final GMM
Sample data
DACE-GMM samples
Verify
L2 error
MC samples
Uncertanity reconstruction block (Matlab)
R_A,R_B
Generate quadrature points
Extract boundary
Quadrature points
Compute relative state
Surface integral
$p_c(t_i)$
Collision probability rate block (Matlab)
Legend
Input
Output
Single-time acion
i = i + Δt
No
i = $t_{f,enc}$?
Yes
$p_c(t)$
Numerical integration
Pc
Collision probability block (Matlab)
Plot

Figure 7.1: Software architecture of the DA-GMM method of collision probability calculation between two space objects, A and B. Black box represents propagation code block, whose architecture is illustrated in Figure 7.2.

DACE architecture

The propagation code block defined in Figure 7.1 as a *black box* plays an important role in the architecture of the DA.GMM method. While conventional applications that require the propagation of orbits in a perturbed environment can make use of already existing astrodynamics software, this is not possible for the DA-GMM method. The reason is that, to integrate in a differential algebra environment such as DACE, it is required to provide an analytical expression for all the functions involved in the propagation. This includes the calculation of the Legendre polynomials and the frame transformations. In Tudat, these operations are numerically evaluated, and therefore cannot be used in this context.

The software architecture for the propagation code block implemented in DACE is illustrated in Figure 7.2. As it can be observed, the architecture is divided into four blocks. The first three blocks contain the diverse dynamical models that are required to compute the total acceleration acting on the space object. The final block discusses the numerical integration of the equations of motion to obtain the Taylor expansion of the final state with respect to a perturbation from the initial state. In reality, the integrator will be continuously accessing the acceleration function to propagate the trajectory from a given time to the following time step.

The acceleration blocks work as follows. First, the initial state of the satellite is provided as a differential algebra variable. This state is transformed to the different reference frames that are required to calculate the parameters (orientation angles) involved in the calculation of environment variables. These include, calculating the longitude and latitude, the flight-path angle and the heading angle. With these variables, it is possible to extract the atmospheric density at a given location and to transform the aerodynamic forces from the aerodynamic reference frame to the inertial reference frame. With the remaining spacecraft parameters obtained from the properties class, the aerodynamic acceleration can be computed. Regarding the gravitational acceleration from the Earth considering the spherical harmonics up to degree and order six, the Legendre polynomials need to be computed analytically. This is done in an adding loop until the total potential gradient in spherical coordinates is calculated. It can then be converted to Cartesian coordinates to obtain the spherical harmonics acceleration in the correct form. Finally, the third body acceleration from the Sun and the Moon is easily calculated once the position of these bodies is retrieved from the ephemeris in the SPICE library.

7.1.3. Software optimisation

During the development of the DA code block in DACE, it was readily noticed that some operations in computational algebra were highly time consuming. To perform the uncertainty propagation, it is required to integrate the equations of motion along the desired time interval for every Gaussian mixture element. This entails multiple evaluations of the code. Therefore, a small reduction in computation of each operation will result in a large increase in computational efficiency. Considering this, code optimisation becomes of crucial importance to this work. To this end, the computational load related to each operation in the differential algebra software is evaluated, and alternatives for the same operation are considered. With the resulting information, several CPU-reduction techniques are devised and applied to the code.

CPU load of DACE operations

Performing operations in the differential algebra computational environment requires a large computational load compared to traditional floating-point computations. This method operates based on the Taylor series expansion coefficients of each expression instead of its evaluation. While this approach provides extra information on the derived expressions and allows to propagate the uncertainty without the need of a Monte Carlo simulation, the time required for a single propagation is higher. The software DACE includes a class *DA* where the mathematical expressions are redefined to operate with the Taylor expansion coefficients of the variables. To work within this environment, the *DA* class needs to be initialised with the expansion order and the maximum number of components that a variable can have. Then, the variables can be initialised as part of the DA class. The operations applied to these variables will be conducted with the expansion coefficients.

Due to the use of Taylor expansion-based arithmetic, the computational load does not follow the rules of traditional operations (e.g., a power can be evaluated in less time than its equivalent multiplication). To gain understanding on this logic and optimise the coding method, the computational time of each individual operation in the differential algebra environment is evaluated.

Table 7.1 presents the CPU time required to conduct 10^6 evaluations of several operations in the differential algebra environment. The computational time associated to operations in the differential algebra environment is immense compared to the same operation performed with a *double* variable. As a consequence, any detail in the code can have a large repercussion in its performance. From these results, several

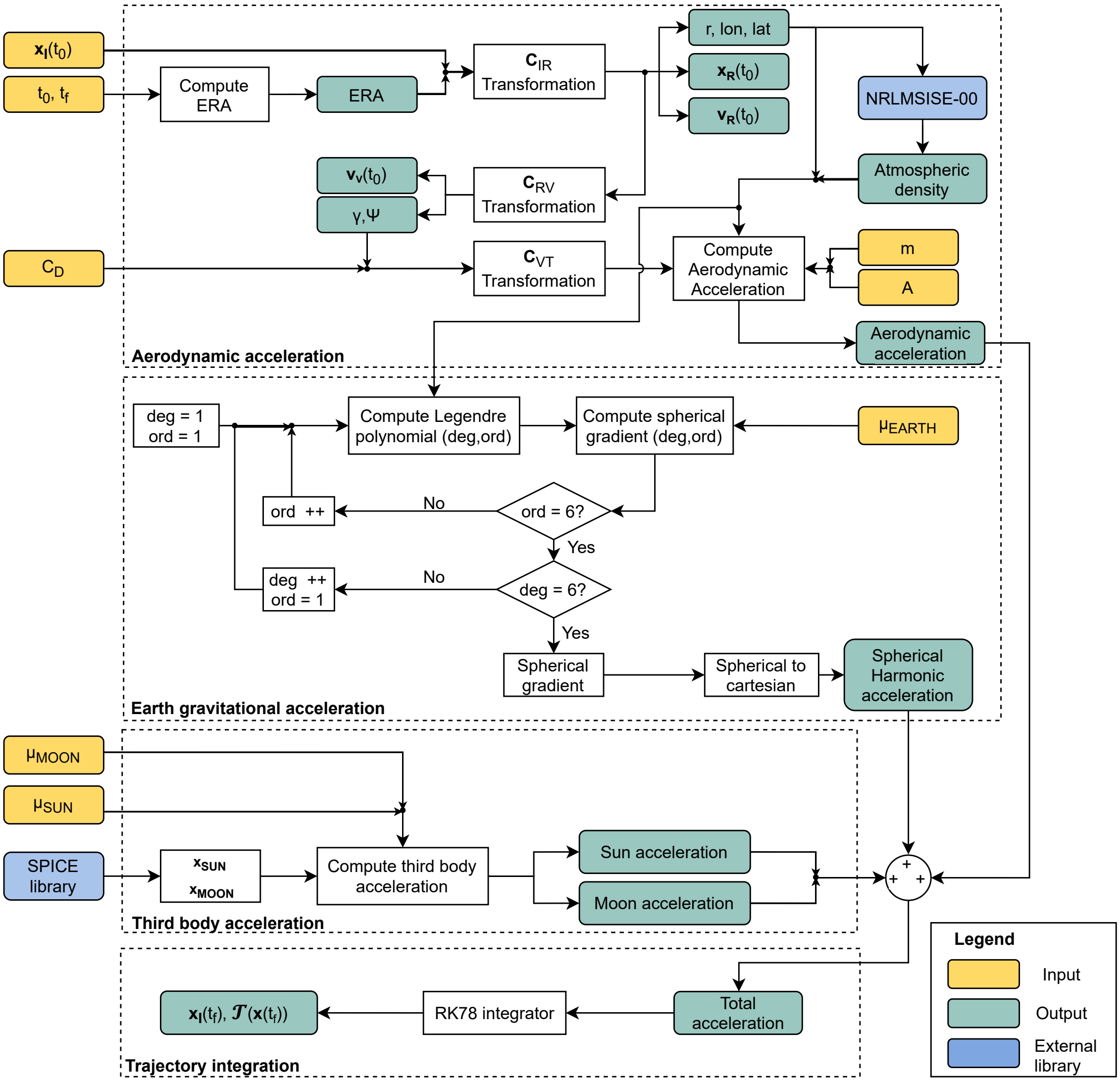


Figure 7.2: Software architecture of the propagation code block in DACE (C++)

observations can be made which are crucial to optimise the code.

First, the time required for the initialisation and declaration of a DA vector is proportional to the number of components. Therefore, no special attention should be paid to the vector size. Second, it is noted that the computational load for initialisation and declaration of a variable is very significant. In principle, this would mandate to minimise the number of variables to use. However, re-declaration of a variable takes approximately one tenth of the time of declaring the variable. The declaration of extra variables can therefore be an advantage when the same operation is used in multiple places of the code. As an example, if the term $sin^3(x)$ needs to be used in several equations, the operation will be performed each time at a cost of $4.3 \cdot 10^{-3}$ ns. However, if a separate variable is created to compute this operation, reusing the variable to evaluate each equation will only take $1.3 \cdot 10^{-5}$ ns. Finally, the fact that the operations are conducted based on the Taylor series expansion provides unexpected results regarding the computational time. For example, the time required to evaluate a power is always smaller than the time required to evaluate the equivalent product and a division takes more than three times longer than the equivalent product.

CPU reduction techniques

Based on the information summarised in Table 7.1 regarding the computational time associated to each expression, the following recommendations are issued and applied to the code:

Table 7.1: CPU time of 10^6 evaluations of operations in the DA environment.

Operation	Expression in DACE	Time [ns]
Initialise vector of 6 components	`AlgebraicVector<DA> vec6(6);`	460
Initialise vector of 3 components	`AlgebraicVector<DA> vec3(3);`	285
Initialise variable	`DA var;`	73
Declare vector of 6 components	`vec6[0] = 0.0 + DA(1);`	1050
Declare vector of 3 components	`vec3[0] = 0.0 + DA(1);`	520
Declare variable	`var = 3.0 + DA(1);`	168
Re-declare variable	`var = var2;`	13
Multiply with a double	`var = var2 * 0.5;`	85
Divide by a double	`var = var2 / 2.0;`	88
Multiply with a DA variable	`var = var2 * var3;`	291
Divide by a DA variable	`var = var2 / var3;`	911
Sine of a DA variable	`var = sin(var2);`	642
Cosine of a DA variable	`var = cos(var2);`	634
Tangent of a DA variable	`var = tan(var2);`	2053
Square root of a DA variable	`var = sqrt(var2);`	586
DA variable powered to 0.5	`var = pow(var2,0.5);`	587
DA variable powered to 2	`var = pow(var2,2.0);`	221
DA variable powered to 3	`var = pow(var2,3.0);`	414
DA variable powered to 4	`var = pow(var2,4.0);`	434
DA variable powered to 5	`var = pow(var2,5.0);`	688
DA variable powered to 6	`var = pow(var2,6.0);`	756
Initialise and declare $\sin^2(x)$	`DA var = pow(sin(var2),2.0);`	819
Declare $\sin^2(x)$	`var = pow(sin(var2),2.0);`	835
Re-declare $\sin^2(x)$	`var = var2;`	13
Initialise and declare $\sin^3(x)$	`DA var = pow(sin(var2),3.0);`	1090
Declare $\sin^3(x)$	`var = pow(sin(var2),3.0);`	4355
Re-declare $\sin^3(x)$	`var = var2;`	13

1. Remove repeated products: it is more efficient to declare an auxiliary variable defining the equivalent product.
2. Declare variables only once: ensure that no re-declaration of variables is present inside functions or loops.
3. Enter vector rotation functions with sin(angle) and cos(angle) to avoid trigonometric function repetitions.
4. Create a single function with all accelerations to avoid multiple evaluations of rotation functions.
5. Create the atmosphere model once and pass it as a pointer. The creation of the atmosphere model is extremely time consuming and therefore must be done in the main code and passed to the integrator as a pointer.
6. Substituting division by product: as observed, the equivalent product is more efficient than performing a division (e.g., coding $a \times 0.5$ instead of $a/2$).

The result of applying these recommendations on the CPU time of evaluating different functions is summarised in Table 7.2. As observed, the largest time reduction comes from the creation of the aerodynamic acceleration model, which was initially done inside the acceleration function. Besides this, with the code optimisation techniques previously described it is achieved to reduce the computational time of all functions.

Table 7.2: CPU time of 10^4 function evaluations before and after code optimisation.

Function	**Initial time [μs]**	**Optimised time [μs]**
computeSphericalHarmonicsAcceleration	441	302
computeThirdBodyMoonAcceleration	8.1	8.0
computeThirdBodySunAcceleration	8.1	8.0
computeAerodynamicAcceleration	3114474	90
computeTotalAcceleration	3114474	388
Creation of the atmosphere model	3114931	2883011
convertGradientSphericaltoInertial	15.62	12.31
computeNormalisedLegendrePolynomial	1.1-1.9	0.9
computeNormalisedLegendrePolynomialDerivative	1.1-1.9	1.1
convertInertialtoLlr	9.7	7.7
convertInertialtoBodyFixed	1.3	0.8
convertBodyFixedtoInertial	1.2	0.8
convertInertialVelocitytoBodyFixed	1.3	-
convertBodyFixedtoVertical	11	2.8
convertVerticaltoBodyFixed	11	2.4
convertTrajecorytoVertical	11	8.5
convertVectorTrajecorytoInertial	55	10

The final function required to integrate the trajectory in the differential algebra software, which combines all the perturbations is *computeTotalAcceleration*, which calculates the acceleration at a given time, adding the effect of the spherical harmonic gravitational acceleration, the third body acceleration from the Sun and the Moon and the aerodynamic acceleration. Each evaluation of this function is performed in approximately 39 ns, which is sufficiently fast to integrate the trajectory for multiple Gaussian mixture elements. Approximately 75% of the computational load of the method corresponds to the computation of the spherical harmonics acceleration, mainly due to the analytical evaluation of the Legendre polynomials. This implies that increasing the degree and order of the spherical harmonics would result in a large increase in computational load that could result unfeasible for operational collision probability estimation.

7.2. Numerical methods

To develop the DA-GMM method for collision probability calculation, several numerical techniques are required. These techniques can be divided into two categories. First, numerical methods that are widely used in most astrodynamics problems and are therefore treated in Appendix A. These include the root finding methods used for the optimisation of the GMM splitting; interpolation methods used to estimate the numerically integrated trajectory at mid-points; numerical integration techniques to propagate the trajectory and statistical sampling and solving methods. Second, there are two methods involved in methodologies developed in this work that require a more careful description. On one hand, the methodology developed to propagate the uncertainty using the Hafnian operator with pre-computed matrices and matrix algebra, which turns the method 3,700 times faster. On the other hand, the Lebedev quadrature for the flux integration over the surface of a sphere, which is used to compute the collision probability and is a basis for the multi-sphere model developed.

7.2.1. Covariance propagation with the Hafnian operator

One of the main benefits of using the Differential Algebra method for uncertainty propagation is that the final state uncertainty can be obtained from the Taylor series expansion of the initial state, following Equation (5.46), as demonstrated in Section 5.4. Although the expression appears simple, it involves the calculation of the expectation of a product of normal random variables, which is not straight forward. A formula for this expression was originally derived by Isserlis (1918), and involves the use of the Hafnian operator:

$$E\{x_1^{s_1} x_2^{s_2} \dots x_{n}^{s_n}\} = \begin{cases} 0, & \text{if } s \text{ is odd} \\ \mathrm{Haf}(\boldsymbol{P}), & \text{if } s \text{ is even} \end{cases} \tag{7.1}$$

where $s = s_1 + s_2 + ... + s_n$. Considering that the expression in this case comes from a Taylor series expansion, the maximum value of s coincides with the Taylor expansion order. When s is even, the expectation of that term is equal to the value of the Hafnian operator evaluated for the initial state covariance matrix $\boldsymbol{P}$. For this calculation, an expanded state vector needs to be defined from the original random vector $\delta\boldsymbol{x}$, as:

$$z^* = \left[z_1 1'_{s_1}, z_2 1'_{s_2}, \dots, z_n 1'_{s_n}\right]' \tag{7.2}$$

where $1'_{s_i}$ is a unit vector of size equal to the exponent of element $\delta\boldsymbol{x}_i, s_i$. The initial state covariance matrix $\boldsymbol{P}$ is now associated to this extended random vector and the Hafnian operator for a given matrix Σ defined by components $\sigma_{i,j}$ is calculated with (Valli et al., 2013):

$$\mathrm{Haf}(\Sigma) = \sum_{p \in \Pi_s} \prod_{i=1}^{\frac{s}{2}} \sigma_{p_{2i-1}, p_{2i}} \tag{7.3}$$

where Π_s is the set of permutations of $1, 2, ..., s$ that satisfy the property $p_1 < p_3 < p_5 < \cdots < p_{s-1}$ and $p_1 < p_2, p_3 < p_4, \dots, p_{s-1} < p_s$. Although this might seem complex, this operator is simply a sum of the product of the covariance matrix elements corresponding to every value of each permutation. The complexity of the operator does not come from the coding itself, but for the high number of function evaluations that are required as the number of permutations increases. As observed, there are multiple sum and product operators that indicate that each calculation needs to be performed multiple times. The Taylor expansion for each component is formed by a large number of terms. For the covariance propagation, the number of evaluations of the expectation operator is equal to the number of components of each coefficient to the square, for each element in the covariance matrix. Therefore, despite being a simple operator in terms of programming, it requires a large computational load, that becomes quickly unfeasible. It must be taken into account that for the DA-GMM method, the propagation of covariance is performed for every Gaussian mixture element at each time step. This can require up to 100,000 evaluations for a given encounter scenario. Therefore, having a low computational load for the evaluation of Equation (5.46) is of foremost importance. By evaluating the full expression for a state consisting of six components (three-dimensional position and velocity), the computational load for this evaluation is of 190 seconds and ten hours for Taylor expansion order three and four respectively. In any case, this is invalid for performing timely uncertainty propagation with the full Gaussian Mixture Model and considering multiple time stamps.

To reduce the computational time, it is noted that the computation of the expectation operator in this context is a function of the initial state covariance matrix and the Taylor expansion exponents of each random variable for each expansion term. While the former changes for every propagation, the latter it is constant. Based on this idea, it is here developed a method to divide the expectation and Hafnian operator into a constant part and a variable part, that can be expressed as two matrices. Therefore, the constant matrix can be loaded once and the full expectation can be computed as a simple matrix product. By computing the summation along all Taylor expansion coefficients as a matrix product as well, the full uncertainty propagation can be reduced to a few matrix operations. The mathematical theory behind this concept is easier explained with an example. For such, let's assume a simpler scenario with two-dimensional motion. For a given summation term in Equation (5.46), the expectation can be computed as a function of only the covariance matrix. In this example, the covariance matrix is $P_{4\times4}$ and for a given term of the summation, the evaluation of the expectation operator results in the expression $f(\delta\boldsymbol{x}, P)$:

$$f(\delta\boldsymbol{x}, P) = 3P_{1,1}P_{1,2} + 4P_{2,1}^2 P_{2,2} + 5P_{1,1}^3 P_{1,2} P_{2,2} \tag{7.4}$$

This operation can be performed for every summation term in the covariance propagation equation, resulting in a total of n_{coeffs}^2 expressions, where n_{coeff} is the number of terms in the Taylor series expansion of each component. The number of addends in these expressions will be equal or lower than the maximum number of permutations in Pi_s, n_{perms}. Based on this information, two matrices are defined, regarding the coefficients and exponents of these expressions. For each matrix, each row will correspond to an addition term in the summation operator of Equation (**??**), and the rows are linked to elements in the covariance matrix.

Table 7.3: Computational time associated to covariance propagation for six state components with the original and simplified methods of obtaining the Hafnian operator.

Taylor expansion order	Full Hafnian	Matrix simplification
3	190 s	0.051 s
4	>10 h	1.63 s

The coefficients matrix has n^2_{coeffs} rows and n_{perms} columns, where the value of each column, n represents the coefficient that multiplies the n_{th} addition term in the expression. The exponents matrix has n^2_{coeffs} rows and $n_{perms} \cdot n^2_{comp}$ columns, where n_{comp} is the number of state components, in this case four. In this example, the first row of the coefficient and exponent matrices is:

$$\text{coeffs} = [3\ 4\ 5] \tag{7.5}$$

$$\exp = [1\ 1\ 0\ 0\ 0\ 0\ 2\ 1\ 3\ 1\ 0\ 1] \tag{7.6}$$

To compute the propagated covariance with matrix operations, the initial state covariance needs to be redefined as a vector and repeated n_{perm} times, such that:

$$A_{vec} = [P_{11}\ P_{12}\ P_{21}\ P_{22}] \tag{7.7}$$

$$A = \underbrace{[A_{vec}\ A_{vec} \cdots A_{vec}]}_{n_{perm}}\ {}_{[1\times n^2_{comp}\cdot n_{perms}]} \tag{7.8}$$

With this format, the result of the expectation operator can then be computed with matrix operations as:

$$\boldsymbol{B} = \boldsymbol{A}^{\exp}\ {}_{[1\times n^2_{comp}\cdot n_{perms}]} \tag{7.9}$$

$$\boldsymbol{C} = \underbrace{\left[\prod_{i=1}^{n_{perms}} B_i \quad \prod_{i=n_{perms+1}}^{2\cdot n_{perms}} B_i \quad \cdots \quad \prod_{i=(n^2_{comp}-1)\cdot n_{perms}+1}^{n^2_{comp}\cdot n_{perms}} \right]}_{n_{comp}}\ {}_{[1\times n^2_{comp}]} \tag{7.10}$$

$$D_i = \text{coeffs}_i \cdot C_i\ {}_{[1\times n^2_{comp}]} \tag{7.11}$$

Matrix $\boldsymbol{D}$ contains in each row the expectation corresponding to each addend of Equation (7.4). The remaining addition can be very efficiently calculated in Matlab using matrix operations. By making use of the pre-computed elements of the Hafnian for a given expansion order and number of components in the coefficients and exponents matrices, the computational load of the uncertainty propagation is largely reduced.

As observed in Table 7.3, the computation of the uncertainty propagation is $> 3{,}700$ times faster using the simplified method for Taylor expansion $k = 3$, and $> 22{,}000$ times for $k = 4$. This has a huge impact on the computational efficiency of the DA-GMM method and on the settings that are allowed. Thanks to this reduction in computational time, it is possible to estimate the collision probability in a timely manner for complex models that include a large number of GMES and long encounter periods.

7.2.2. Lebedev's quadrature

Calculating the surface integral of a flow over a given volume cannot always be done analytically. As an extension of numerical integration techniques for one-dimensional problems, surface quadratures have been developed to calculate these type of surface integrals. These quadratures have been widely researched for spheres (Lebedev, 1976), and recently investigated for other closed, smooth surfaces (Reeger et al., 2016). From the available quadrature techniques for a sphere, Lebedev's method is the most widely used since it provides the best accuracy for a given number of quadrature points. The nodes and weights are pre-computed, which largely increases the computational advantage of using this numerical method to integrate the flow over a sphere.

The numerical quadrature is used to compute the surface integral of a function $f(\mathbf{x})$ over the unit sphere $\mathbb{S}^2 = \{\mathbf{x} \in \mathbb{R}^3 : \|\mathbf{x}\|_2 = 1\}$.

$$I[f] \equiv \int_{\mathbb{S}^2} f(\mathbf{x})\mathrm{d}\Omega = \int_0^{2\pi} \int_0^{\pi} f(\varphi,\theta)\sin\varphi \mathrm{d}\varphi \mathrm{d}\theta \tag{7.12}$$

where φ, θ are the polar coordinates of the sphere, the polar angle and azimuth angle respectively. A numerical quadrature approximates the integral $I[f]$ by a quadrature $Q[f]$ such that;

$$\int_{\mathbb{S}^2} f(\mathbf{x}) \mathrm{d}\Omega \approx \sum_{i=0}^{N-1} w_i f(\mathbf{x}_i) \equiv Q[f] \tag{7.13}$$

Therefore, in this case it is only required to evaluate the function at each quadrature point and perform the weighted sum of the result.

The calculation of the Lebedev quadrature relies heavily on the concept of spherical harmonics. It is assumed that the function on the unit sphere can be expanded in terms of the spherical harmonics orthogonal basis of the unit sphere (Atkinson and Han, 2012). By selecting a given degree for the spherical harmonics, the integration over the sphere becomes a system of non-linear equations. This yields a large system of equations of difficult solution. To reduce this computation, Lebedev bases the quadrature on the subset of the spherical harmonics according to the theorem proposed by Sobolev (1962), which states that a quadrature scheme is excact for a subset of rotations of the sphere if and only if it is exact for all the invariant harmonics of this degree. This allows to reduce the system of equations by computing the quadrature for symmetry groups of octahedron rotations. The weights and quadrature points for Lebedev's method are retrieved from the Matlab function *getLebedevSphere*[6].

7.3. DACE verification

This section describes the verification of the DA code block performed in DACE. Although this code block only represents one part of the collision probability calculation algorithm, it is a crucial element that must be carefully verified due to its high complexity. The DA code block is verified in three steps. First, a set of unit tests are performed that cover all the functions developed within this software. For comparison, analytical results are used when possible, and equivalent results from Tudat are used in the rest of cases. Second, integration tests are performed to ensure that the different functions are correctly related. This involves testing the treatment of time, since different time units are required as input to the different functions/libraries. Finally, system tests are performed to verify the computation of the total acceleration and the integrated trajectory.

7.3.1. Unit tests

The unit tests can be divided into three segments depending on the functions and capabilities under study. First, it is verified that the properties class is properly created and accessed. Second, that all the frame transformations are correct and output the required value. Finally, the integration of external libraries and custom functions is verified.

Properties class

The properties class includes all the variables of the problem that can be customised, both inputs related to the geometry of the satellite and the mission and model parameters such as the rotation rate of Earth. This class includes multiple functions that allow to set and get the values for these parameters. This functions can be manually tested to ensure that the values obtained match the expected results. The set of tests performed on the properties class are summarised in Table 7.4, showing the name of the test, the input and output, the expected outcome and the result. To pass these tests, the output must be equal to the expected outcome, since no computations are involved and no threshold is defined. All the tests are passed, which verifies that the properties class is correctly constructed and used within the method.

Frame transformations

Multiple reference frames are used in the method, as described in Section 3.1. Transformation between reference frames is required, not only for the position but also for the velocity and spherical gradient of the potential. These operations are highly sensitive since they include trigonometric operations and can result in singularities. To avoid these issues, each function is tested for eight initial states, spanning all four quadrants and checking singular values. For this tests, a threshold of 10^{-12} is defined as pass/fail criterion for the percentage error. To verify the results, two different methods are employed. First, by a manual calculation of the expected outcome. Second, the frame transformation is checked against the already verified functions available in Tudat.

[6] Available at: `https://www.mathworks.com/matlabcentral/fileexchange/27097-getlebedevsphere`.

Table 7.4: Unit tests of body properties class (PC)

ID	Test description	Input	Output	Expected outcome	Pass / Fail
Class construction					
PC - 1.1	Test successful default construction of the body class.	-	-	Class is constructed and can be accessed.	Pass
PC - 1.2	Test successful custom construction of the body class.	Start date, C_D, C_L, C_S, A, m	-	Class is constructed and can be accessed.	Pass
Get properties					
PC - 2.1	Test getVehicleMass	-	m	1 kg for default construction, custom value for custom construction	Pass
PC - 2.2	Test getVehicleArea	-	A	1 m^2 for default construction, custom value for custom construction	Pass
PC - 2.3	Test getEquatorialRadius	-	R_E	$R_E = 6378.137$ km	Pass
PC - 2.4	Test getMuEarth	-	μ_E	$\mu_E = 398600.4418\ \text{km}^3\text{s}^{-2}$	Pass
PC - 2.5	Test getMuSun	-	μ_S	$\mu_S = 1.327122 \cdot 10^{11}\ \text{km}^3\text{s}^{-2}$	Pass
PC - 2.6	Test getMuMoon	-	μ_M	$\mu_M = 4902.801076\ \text{km}^3\text{s}^{-2}$	Pass
PC - 2.7	Test getStartEpoch	-	Start date	0 JD for default construction, custom value for custom construction	Pass
PC - 2.8	Test getAerodynamicCoefficients	-	C_D, C_L, C_S	Custom value from custom construction	Pass
PC - 2.9	Test getSineCoefficient	degree (n), order (m)	$S_{n,m}$	Sine and cosine expansion coefficient match values of model EMG2008 (Pavlis et al., 2012)	Pass
PC - 2.10	Test getCosineCoefficient	degree (n), order (m)	$C_{n,m}$		Pass
Set properties					
PC - 3.1	Test setVehicleMass	m	m	Retrieved mass equals input	Pass
PC - 3.2	Test setVehicleArea	A	A	Retrieved area equals input	Pass
PC - 3.3	Test setMuEarth	μ_E	μ_E	Retrieved μ_E equals input	Pass
PC - 3.4	Test setMuSun	μ_S	μ_S	Retrieved μ_S equals input	Pass
PC - 3.5	Test setMuMoon	μ_M	μ_M	Retrieved μ_M equals input	Pass
PC - 3.6	Test setStartEpoch	Start date	Start date	Retrieved date equals input	Pass
PC - 3.7	Test setAerodynamicCoefficients	C_D, C_L, C_S	C_D, C_L, C_S	Retrieved coefficients equal input	Pass

In Table 7.5, the description and results of these unit tests are summarised. All the tests are successfully completed for all input values.

External libraries and custom functions

A third set of unit tests is devised to verify the integration of external libraries inside the main code. Although these libraries provide multiple functionalities, these are already verified. Therefore in this section, only the functionalities required by the DA-GMM method are tested. For the SPICE library, this involves the functionality of retrieving the position of the Moon and the Sun at a given ephemeris time, and the functionality of converting an ephemeris date into a Julian date. To verify this functionalities, the results obtained in DACE are compared to the results obtained in Tudat. The same method of verification is used to test the retrieval of atmospheric density from the NRLMSISE-00 model at a given location and time. Since these functionalities

Table 7.5: Unit tests of frame transformation functions (FT)

ID	Test description	Input	Output	Expected outcome	Pass / Fail
FT - 1.1	Test convertInertialtoLlr	$\boldsymbol{r}^I$, ERA	$\bar{\boldsymbol{r}}, \tau, \delta$	Output equals result from independent multiplication of each individual rotation matrix with input vector. Output also equals result from equivalent frame transformation in Tudat. Every function is called with eight different input values for each angle to verify all combinations.	Pass
FT - 1.2	Test convertInertialtoRotating	$\boldsymbol{r}^I$, ERA	$\boldsymbol{r}^R$		Pass
FT - 1.3	Test convertRotatingtoInertial	$\boldsymbol{r}^R$, ERA	$\boldsymbol{r}^I$		Pass
FT - 1.4	Test convertRotatingtoVertical	$\boldsymbol{r}^R, \tau, \delta$	$\boldsymbol{r}^V$		Pass
FT - 1.5	Test convertVerticaltoRotating	$\boldsymbol{r}^V, \tau, \delta$	$\boldsymbol{r}^R$		Pass
FT - 1.6	Test convertTrajectorytoVertical	$\boldsymbol{r}^T, \gamma, \chi$	$\boldsymbol{r}^V$		Pass
FT - 1.7	Test convertTrajectorytoInertial	$\boldsymbol{r}^T, \gamma, \chi, \tau, \delta$, ERA	$\boldsymbol{r}^I$		Pass
FT - 2.1	Test convertInertialVelocitytoRotating	$\boldsymbol{r}^R, \boldsymbol{v}^I$	$\boldsymbol{v}^R$		Pass
FT - 2.2	Test convertGradientSphericaltoInertial	$\Delta U^S, \boldsymbol{r}^R$, ERA	ΔU^I		Pass

do not involve any computations, a threshold error of 10^{-15} is set as acceptance criteria for these unit tests. Besides the fame transformations and the functions computing the acceleration on the space object, there are two other custom functions created for this method which need to be carefully verified. These are the calculation of the Legendre polynomial and its derivative for a given degree, order and latitude angle. The values obtained are verified with hand calculations and with an equivalent function publicly available on Matlab.

The results from these unit tests are found in Table 7.6. As observed, all the functions provide the expected result and the inclusion of external libraries and custom functions is verified.

7.3.2. Integration tests

Integration tests are performed to check that all components are correctly combined and the correct inputs are provided to every function. This is done first by a visual inspection of every function call. There is an element of the code to which special attention must be paid. It is the time variable, which is required by several functions in different formats. There are two main units of time used in the code. First, ephemeris time, which is defined as seconds passed since J2000 (January 1, 2000 at 12:00 TDB). This time variable is used to retrieve the position of the Earth and the Moon from the SPICE library and the density from the NRLMSISE-00 library. For other internal operations, time is expressed as a Julian date while the TLEs provide the observation time as the calendar day and the fractional portion of the day passed.

To ensure that time is being correctly handled, several integration tests are performed. In the DA code block, epoch time defined as seconds since J2000 is the main time variable used and is transformed to any other units when needed. First, the unit transformations from epoch time to Julian days or calendar date are coded and verified with the equivalent transformation in Tudat. In this case, the transformation must be exact and therefore, no threshold for the allowable error is defined. Second, it is inspected and checked that every function is accessed with the correct time unit.

7.3.3. System tests

The final verification test to be performed is the system test, which checks that the full DA code block works as expected and correctly integrates the Taylor series expansion of the final state with respect to a perturbation in the initial state. To do so, the individual acceleration functions for each perturbation are verified as well as the total acceleration function. Moreover, the integrated trajectory is also tested. To compare the results, the outcome of the same function in Tudat is used, and a percentage error threshold of 10^{-14} is selected. To verify that the results do not depend on the initial state, the test is performed for four different input values regarding the satellite state and initial time. The description and results of these tests are summarised in Table 7.7. Every test has been successfully performed, and it is concluded that the DA code block is correctly implemented and verified.

Table 7.6: Unit tests of external libraries (EL) and custom functions (CF)

ID	Test description	Input	Output	Expected outcome	Pass / Fail
			External libraries		
EL - 1.1	Test getting the Moon position from the SPICE library	Ephemeris time	$\boldsymbol{r}_M^I$	Equals value retrieved in Tudat	Pass
EL - 1.2	Test getting the Sun position from the SPICE library	Ephemeris time	$\boldsymbol{r}_S^I$	Equals value retrieved in Tudat	Pass
EL - 1.3	Test conversion of Julian date to ephemeris time	Julian date	Ephemeris time	Equals value retrieved from Tudat	Pass
EL - 1.4	Test getting the atmospheric density from the NRLMSISE-00 library	$\bar{\boldsymbol{r}}, \tau, \delta$, time from J2000	ρ	Equals value retrieved in Tudat	Pass
			Custom functions		
CF - 1.1	Test calculation of Legendre polynomial	degree (n), order (m), δ	P_n^m	Equals hand calculation	Pass
CF - 1.2	Test calculation of Legendre polynomial derivative	degree (n), order (m), δ	$\frac{dP_n^m}{d\delta}$	Equals hand calculation	Pass

Table 7.7: System tests for acceleration functions (ST).

ID	Test description	Input	Output	Expected outcome	Pass / Fail
ST - 1.1	Test third body acceleration Moon	$\boldsymbol{x}^I, t$	$\dot{\boldsymbol{x}}_{Moon}$	Output equals result from Tudat. Test performed with four different initial state of the satellite, $\boldsymbol{x}^I$	Pass
ST - 1.2	Test third body acceleration Sun	$\boldsymbol{x}^I, t$	$\dot{\boldsymbol{x}}_{Sun}$		Pass
ST - 1.3	Test aerodynamic acceleration	$\boldsymbol{x}^I, t$, atmosphere	$\dot{\boldsymbol{x}}_{Aero}$		Pass
ST - 1.4	Test spherical harmonic acceleration	$\boldsymbol{x}^I, t$	$\dot{\boldsymbol{x}}_{Earth}$		Pass
ST - 1.5	Test total acceleration	$\boldsymbol{x}^I, t$, atmosphere, Body	$\dot{\boldsymbol{x}}_{Total}$		Pass
ST - 1.6	Test trajectory integration	$\boldsymbol{x_0}^I, t$, atmosphere, Body	$\boldsymbol{x_f}^I$		Pass

7.4. Pre- and post-processing code verification

This section explains the verification of the pre- and post- processing code blocks that include the GMM splitting methodology, the covariance reconstruction and the collision probability calculation. In this case, verification has been performed by comparing the results of each function to results from equivalent methods found in research. In this section a brief explanation of how each process has been verified is provided, referring to the tests and comparisons provided in previous chapters and explaining additional checks performed.

7.4.1. GMM splitting

The Gaussian Mixture splitting methodology can be divided into two steps. First, the generation of a univariate splitting library and second, the refinement of the split over different dimensions.

The univariate split has been performed in Maple for up to 2001 Gaussian Mixture Elements. The verification of this procedure has been performed in two ways. First, by calculating the L_2 error, which determines the difference between two distributions, in this case the univariate Gaussian distribution and the univariate Gaussian mixture. Figure 5.4 shows the result of this procedure, and it is observed that the error obtained is smaller that in equivalent methods, such as the one presented by Horwood et al. (2011). Second, the results of the univariate splitting are compared to the ones from Horwood et al. (2011); Vittaldev and Russell (2016). The results follow the same curve, as observed in Figure 5.3a, with small differences due to the improved accuracy and different splitting range.

The split refinement over different dimensions is verified by calculating the L_2 distance between the multivariate Gaussian and Gaussian Mixture Model, result of the Gaussian splitting. In this case, the L_2 error obtained with 1001 elements is three orders of magnitude smaller than the one reported in research for similar methods. Moreover, the result from this split is visualised in Figure 5.5, by plotting a set of samples drawn from each distribution. From these tests, it is concluded that the splitting methodology is correctly verified.

7.4.2. Covariance reconstruction

The covariance reconstruction process allows to obtain the mean and covariance matrix of the propagated state uncertainty, knowing the covariance of the initial state uncertainty. To verify this process, the uncertainty propagation is first applied to the two-dimensional car example presented in Subsection 6.2.2. For this case, the uncertainty propagation can be calculated analytically, since the motion is linear. Comparing the expected results with the mean and covariance obtained with the propagation of uncertainty using the Hafnian operator, a perfect match is obtained.

To verify the covariance reconstruction with a more generic case, a sample from the initial state is created for the two-body problem. This case is very convenient since the analytical solution of the equations of motion is available, and therefore, the exact final state of each sample can be obtained. This allows to compute the final uncertainty of the propagated sample and compare it with the covariance reconstructed using the Hafnian operator. Following this process, it is concluded that the covariance reconstruction algorithm is verified.

The covariance reconstruction process can be easier visualised by drawing samples from the propagated GMM and comparing them to the results of a Monte Carlo analysis. This gives a better visual interpretation than the L_2 parameter. To do so, the propagation for nominal scenario #1 is considered, with the full dynamics model. Figure 7.3 presents the results for the samples drawn from the propagated uncertainty with 37 and 501 GMEs and for 1, 5, 15 and 30 orbital periods. It is observed that up to 5 revolutions, both distributions visually present the same direction and span. It must be considered that the number of samples drawn is limited from the Monte Carlo computational load, and therefore this is only a visual estimation. When increasing to 15 orbits, the uncertainty distribution presents a curvature (cannot longer be assumed to be Gaussian), due to the nonlinear dynamics. With the GMM, it is possible to replicate this curvature, even with only 37 elements. For 30 orbits it is observed that the data sampled from the Gaussian Mixture Model expands further than the one from the Monte Carlo simulation. In this case, it cannot longer be assumed that both distributions match.

7.4.3. Collision probability calculation

The final element to verify is the collision probability calculation using the direct integration method. From this algorithm, two elements are checked. First, that the time of maximum collision probability is correctly estimated. Second, that the collision probability estimated is correct. To do so, again the first test is performed with the two-dimensional, linear example provided by the car crash. In this case, the time of maximum collision probability is known, since it is the time of the nominal crash at $t = 180$ s. To compute the collision probability, a Monte Carlo simulation is performed, which consists in drawing a set of samples from the initial distribution of each car and propagating the trajectory of each sample to compute the ratio of encounters that result in a collision. With this procedure, it is verified that the method works for a linear encounter in two dimensions.

To verify the method for the calculation of collision probability between satellites in three-dimensional space, a large number of Monte Carlo samples (i.e., 10^7 are required to produce an accurate result. Due to the large impact that this has on computational load, it is chosen to follow the examples simulated by Alfano (2009) that provide the collision probability for given scenarios with very high accuracy. This allows, not only to verify the collision probability calculation performed in this work, but also to compare the results to alternative methods. This analysis is detailed in Section 9.1 for two test cases. The outcome of this analysis ensures that the collision probability calculation is correctly performed and it is concluded that the DA-GMM method is verified.

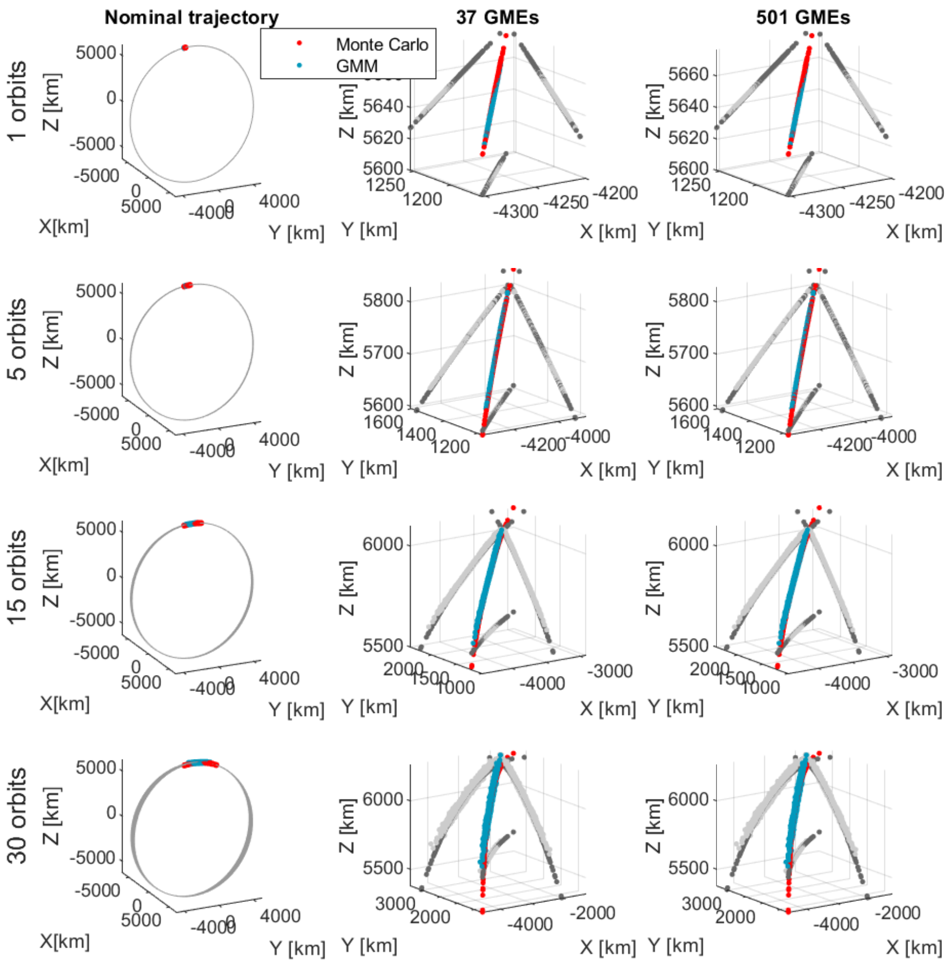


Figure 7.3: 3000 samples drawn from Monte Carlo simulation (red) and from the covariance propagation with the DA-GMM method (blue) for 31 and 501 GMEs, and 1,5,15,30 orbital revolutions.

For the multi-sphere model, the example case with two intersecting spheres discussed in Subsection 6.3.3 demonstrates that the method can be used to model a volume with a combination of two spheres. Due to the additive property of collision probability computed via integration of the collision probability rate, it is verified that the procedure can be applied to a given number, N, of spheres.

7.5. Validation

Once verification has been correctly performed, it is required to validate the applicability of the developed method for the application intended, and determine if it satisfies the specified requirements for its intended use. The ideal way to perform validation is by comparing the results of the model with real-life measurements.

With the combination of tests, which is presented in Chapter 8 and Chapter 9, it is concluded that the DA-GMM method is validated for its intended application. A brief description of the scenarios studied is provided here. From past and present close encounters, it is possible to test the model with three real-life collision scenarios. First, with the Cosmos-2251/Iridium-33 collision event that occurred in 2009 leading to the creation of thousands of new debris. This event is further discussed in Section 9.3. In this case, the time of collision is precisely known, which allows to validate the accuracy in predicting the encounter time. Second, the method is tested with a close encounter detected by LeoLabs in 2020 between the inactive satellite Cosmos-2004 and the rocket body Chang Zheng-4C, further discussed in Section 9.4. Although this encounter did not occur, the risk detected by the company LeoLabs was very high. With this analysis, it can be tested if such a high risk is also estimated and. Finally, the results from the anti-missile satellite test that occurred in November 2021 and destroyed satellite Cosmos-1408, threatening the ISS is used to validate the multi-sphere method of collision probability. It is still unclear what risk are these debris are posing to the vehicle crew and with this more precise method, the risk is estimated over a period of two weeks. For each of these test cases, tracking data is obtained from *SpaceTrack* in the form of Two-Line elements and is converted to account for special perturbations using the SGP4 model. However, the initial state uncertainty of this data is not optimum and does not provide the best results compared to radar tracking data such as the one obtained by LeoLabs. This must be taken into account when performing validation and comparing the results to the predictions performed by companies or space agencies.

Fortunately, the number of collision events that have been documented in the recent years is reduced but it still allows to test a variety of encounters. However, alternative methods have to be developed to complete model validation. First, to test the behaviour of the DA-GMM method of uncertainty propagation for different types of orbits and integration time, a sensitivity analysis is performed in Chapter 8. As a result, this analysis provides the limitations of the methods and a guideline the recommended settings to achieve the desired accuracy depending on the satellite orbit and integration time. Second, the tracking data used to validate the real-life scenarios is obtained from TLEs, which have the same accuracy. Therefore, an additional validation test is created to check the effect of changing the initial state uncertainty. To do so, a scenario where the nominal trajectory leas to a collision is simulated and the collision probability is calculated form different initial times and initial state uncertainties. The results from this analysis are then extrapolated to complement the discussion of the results from the real-life test cases.

8 Sensitivity analysis

This chapter presents the sensitivity analysis of the uncertainty propagation using the DACE-GMM method. The main objective of this analysis is to understand the effect of each parameter involved in the propagation on the accuracy of the results and identify its limitations. To this end, three nominal scenarios are introduced in Section 8.1. These cases represent the baseline for the sensitivity analysis and are closely related to the reference missions selected for Chapter 9. Moreover, the reference settings for the method are set. In Section 8.2, the effect of changing the trajectory is studied, with an independent analysis for each orbital element. In Section 8.3, the effect of modifying the settings of the method is analysed and related to the method accuracy and computational load. Finally, Section 8.4 treats the change in initial state uncertainty and how it affects the accuracy of the method for a given lead time. The chapter concludes with a summary of the findings of this sensitivity analysis and a guideline of the recommended settings for different applications.

8.1. Nominal scenarios

Studying the performance of the DACE-GMM method presents a challenge. First, the number of simulations that can be performed in the sensitivity analysis becomes limited due to the large computational time associated with the Monte Carlo verification technique. Moreover, the encounter between two satellites can present infinite geometries depending on the combination of orbital parameters and initial uncertainties. This complicates the sensitivity analysis and brings developing a detailed guide on the optimum parameters for each encounter out of the scope of this work. Instead, a reduced number of cases is studied, considering the boundary values for each parameter to understand the limits of the method. As a basis to perform the sensitivity analysis, three nominal scenarios are designed with an associated "baseline" choice of settings for the propagation with the DACE-GMM method. In this section, the geometry for the nominal scenarios is presented, together with examples of the propagated distribution compared to the equivalent Monte Carlo samples.

The selection of the nominal cases is based in two criteria related to the objectives of the sensitivity analysis. Table 8.1 presents the orbital parameters and the initial state in the Earth Centred Inertial Reference frame ($I-$Frame) defining these nominal scenarios. First, this analysis aims to reflect how the choice of settings for the DACE-GMM method affects the accuracy of the results, allowing to select appropriate values for the cases under study in Chapter 9. Therefore, the nominal scenarios must be similar to the reference missions (i.e., satellites in circular orbits at 800 km altitude and the ISS). This criterion determines the selection of the nominal scenarios #1 and #2. Scenario #1 represents a circular orbit at 800 km altitude and in the most crowded inclination, at 60 degrees. Scenario #2 matches the orbit of the ISS, with a slightly eccentric orbit, a semi-major axis altitude of 422 km and an inclination of 51.65 degrees. Second, the sensitivity analysis will uncover the limitations of the method and therefore a trajectory with higher altitude and larger eccentricity is selected for nominal scenario #3. In this case, the orbit is retrograde with an inclination of 98 degrees. It must be noted that the choice of nominal scenarios mainly affects the analysis in Section 8.3 and Section 8.4, for which the trajectory is fixed.

The geometry of these cases is illustrated in Figure 8.1, with the resulting trajectory after 1 orbital period (dark) and after 100 orbital periods (light). It can be observed how the change in orbital altitude affects the perturbing acceleration, therefore creating a larger spread in the propagated trajectory. The method set-

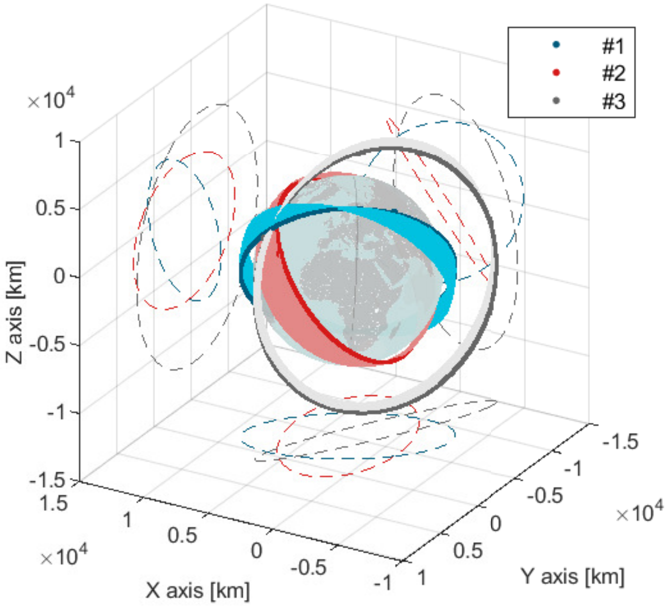


Figure 8.1: Illustration of the nominal scenarios propagated for 100 orbits (light trajectory) and for 1 orbit (darker trajectory),

Table 8.1: Orbital parameters and initial state of the nominal scenarios

Case	***h*** **[km]**	***e*** **[-]**	***i*** **[deg]**	**x [km]**	**y [km]**	**z [km]**	$\dot{\mathbf{x}}$ **[km/s]**	$\dot{\mathbf{y}}$ **[km/s]**	$\dot{\mathbf{z}}$ **[km/s]**
# 1	800	0	60	-4296.3705	1150.9621	5634.0161	-4.8929	-4.9142	-2.7273
# 2	422	0.00038	51.64	1844.5697	5763.9978	3103.7357	-4.3208	4.0133	-4.8801
# 3	3000	0.25	98	3469.5104	5030.4799	3482.5761	-1.2256	-4.1517	7.2180

tings for the nominal scenarios are introduced in Table 8.2 and will be kept constant through the sensitivity analysis unless specified otherwise. The number of samples drawn for the Monte Carlo simulation is limited by the computational load of this computation. The samples are generated using the Mersenne Twister pesudo-random number generator and the seed is changed and documented in each simulation. The number of Gaussian mixture elements is selected according to the analysis performed in Subsection 5.1.3 and the number of Monte Carlo samples is limited by the computational load. During the sensitivity analysis, the uncertainty will be propagated during 5 orbital periods, which corresponds to 8.4, 7.75 and 12.55 hours for nominal scenarios #1, #2 and #3 respectively.

For this analysis, the L_2 distance is used as the metric to evaluate the difference between the propagated Gaussian mixture and the Monte Carlo samples. This performance parameter includes more information than simply comparing the error in covariance between both distributions, since it also considers higher-order moments. In this case, to evaluate the L_2 distance, the Monte Carlo samples are fitted to a Gaussian mixture model, and it is compared to the GMM propagated with the differential algebra software. It must be noted that this parameter is therefore approximated. Moreover, the number of Monte Carlo samples also affects the accuracy of the propagated distribution and therefore the L_2 distance. Knowing the limitations of the calculated L_2 distance, only the order of magnitude will be studied. The L_2 error is not a quantity that can be directly related to any physical property. It is the integral of the squared difference between two distributions. As such, it is a non-dimensional quantity, hard to understand. To visualise what the L_2 magnitude represents, Figure 8.2 illustrates examples for distribution samples propagated with DACE and Monte Carlo, with a L_2 distance ranging from 10^{-5} to 2. It can be observed that for the sample sets with an error of 10^{-5} and 10^{-3}, both distributions, propagated following a Monte Carlo approach and with the DA-GMM method respectively, match and there is no visible difference. For $L_2 = 0.1$, the distribution is more elongated as it corresponds to a longer time propagation. In this case, both distributions follow the same direction and have

Table 8.2: Method parameters for nominal scenarios

Parameter	Value	Units
Nº GME	201	-
Nº Monte Carlo samples	10000	-
Taylor expansion order	3	-
Reference Area	1.5	m^2
Mass	3.3	kg
Radius	0.7	m
C_D	1.2	-
Start date	0	seconds from J2000
Propagation time	5	Orbital period
$[\sigma_R, \sigma_T, \sigma_N]$	[1,4,1]	m
$[\sigma_{V_R}, \sigma_{V_T}, \sigma_{V_N}]$	[0.01 0.0025 0.0025]	m/s

the same span. Again, it is noted that this visual inspection is performed with a limited number of samples, and therefore it is only for illustration purposes. For $L_2 = 0.5$, a similar behaviour is observed, where both distributions follow the same direction but some outlier points are present. When increasing to $L_2 = 1$, the difference between both distributions becomes clearly visible , as it can be appreciated a change in orientation, which is no longer acceptable. Finally, with $L_2 = 2$ the distributions follow different directions and the error becomes extreme. Based on this, it is recommended to keep the error below the threshold of $L_2 = 0.6$.

8.2. Effect of orbital elements

In this section, the uncertainty propagation with the DACE-GMM method will be compared to the equivalent Monte Carlo simulation by means of the L_2 distance for a range of mission geometries. The goal of this analysis is to identify the limitations of the method in terms of the orbital elements that can be propagated. To this end, independent simulations are performed to study the sensitivity of tree parameters: orbital altitude, eccentricity and inclination.

Through this analysis, the percentage change in L_2 with respect to the percentage change in input value from the nominal condition is used to evaluate the effect of each variable. The value of the variable is normalised in the range studied. This performance parameter is hereby defined as S_{L_2}, as:

$$S_{L_2} = \frac{\%\Delta L_2}{\%\Delta p} = \frac{\frac{|L_{2_i} - L_{2nom}|}{L_{2nom}}}{\frac{|\bar{p}_i - p_{\bar{nom}}|}{p_{\bar{nom}}}} \tag{8.1}$$

where $\bar{p}_i$ is the value of the independent parameter, and $p_{\bar{nom}}$ is the value of the independent parameter in the nominal setting. These parameters are normalised for the studied range. In this way, the S_{L_2} parameter quantifies the sensitivity of the error in the uncertainty propagation for each independent variable present in the model. A positive change represents a positive correlation, and vice-versa. By normalising the values, it is possible to compare the results between different parameters.

8.2.1. Change in altitude

The DACE-GMM method was originally conceived to cover the propagation of objects in LEO, which has determined the perturbing accelerations included in the dynamical model. To test if this requirement has been met, a sensitivity analysis is performed for the orbital altitude, keeping a constant eccentricity of zero and a constant inclination of 60 degrees. With the limitation in computational time for the Monte Carlo simulations used for verification, the altitudes must be carefully selected. The boundary for LEO is typically set at 200 km altitude, which then determines the lower bound for this sensitivity analysis. Due to the fast change in density at this altitude band, the second study point is set at 400 km altitude. With the nominal altitude set at 800 km, the remaining step is to define an upper limit. While the boundary for LEO is set at 2000 km altitude, a safety factor is added to test the limiting case, and an altitude of 3000 km is selected. Finally, to test the accuracy of the method outside the initially determined boundaries, an extreme case is

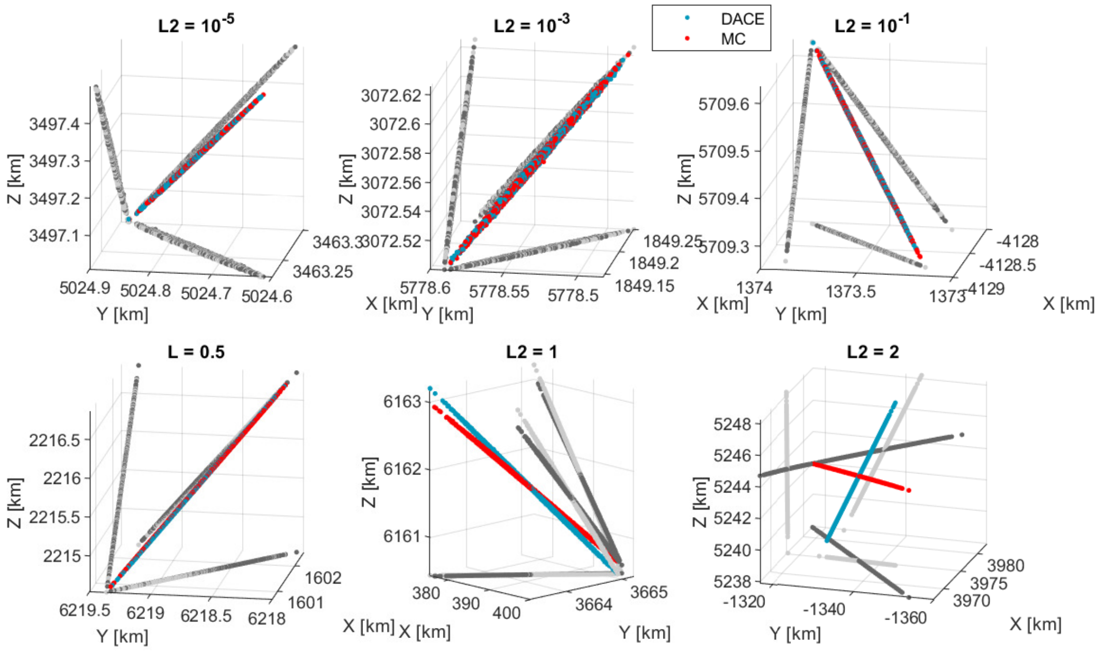


Figure 8.2: Example of L_2 error between Monte Carlo samples and DACE samples.

Table 8.3: L_2 error for changing altitude. Nominal altitude set at 800 km. Shaded cells indicate simulations that violate the threshold for accuracy.

Altitude [km]	**200**	**400**	**800**	**3000**	**35786**
Baseline dynamical model					
L_2 with Monte Carlo	1.3775	0.0522	0.0199	0.0093	0.0251
S_{L_2}	-68.1546	-2.4280	-	-0.1450	0.4808
Full dynamical model, non-displaced mean					
L_2 with Monte Carlo	1.3798	1.2137	1.2037	1.0770	1.253
S_{L_2}	-0.1463	-0.0125	-	-0.0287	0.0713
Full dynamical model, displaced mean					
L_2 with Monte Carlo	1.4190	0.0593	0.0214	0.0094	0.0273
S_{L_2}	-65.3090	-2.6584	-	-0.1529	0.4796

selected, corresponding to a satellite in GEO. The objective is to explore the capability to extend the method to higher altitudes than initially conceived. Two sets of Monte Carlo simulations are performed for this analysis: with the nominal dynamical model defined in Subsection 3.5.5 and with a "full" dynamical model. The latter includes, in addition to the nominal accelerations, the spherical harmonics up to degree and order ten, the solar radiation pressure and the third-body acceleration of Mars and Venus.

Table 8.3 summarises the results of the sensitivity analysis for the orbital altitude with respect to three dynamical models. First, with respect to the baseline, "reduced" dynamical model. Second, with respect to the "full" dynamical model. A third scenario is created by shifting the mean of the uncertainty propagated in DACE to match the nominal trajectory simulated with the full model. First, it is observed that the comparison with the baseline dynamical model results in the lowest L_2 error, as expected since both simulations are propagated with the same equations of motion. When comparing the results with the Monte Carlo simulation run with the full dynamical model, L_2 increases to the order of 1. However, this error does not come from the covariance but from a shift in position. By shifting the mean of the distribution to match the nominal trajectory, the error reduces again to the order of magnitude observed in the comparison with the baseline dynamical model. This reveals a valid alternative to increase the accuracy of the dynamical model without requiring additional Monte Carlo simulations besides the nominal trajectory.

Regarding the accuracy, it is observed that the DACE-GMM method presents a large error for the lower bound, with 200 km altitude. This error decreases below the threshold for an altitude of 400 km and becomes of the order of 10^{-2} for higher altitudes. The reason behind this large increase in error for low altitudes is the larger effect of drag. The drag acceleration is calculated by retrieving the density from the NRMSISE-00 aerodynamic model. This density is highly dependent on the position (e.g., for low altitudes of 400 km, a difference in 10^{-4} m in position can lead to a 3% change in density). For very low altitudes, drag becomes the main perturbing acceleration and this deviation propagates with every integration step, leading to large errors in the propagated uncertainty. For an altitude of 400 km, this effect becomes much smaller and the error is of the same order of magnitude as for higher altitudes. Regarding the upper limit, it is observed that the error is still well below the threshold for the geo-stationary orbital altitude. It can be concluded that the DACE-GMM method is valid for Earth orbits in LEO, MEO and GEO, when the perigee altitude is higher than 400 km.

8.2.2. Change in eccentricity

Considering the results from the altitude sensitivity analysis, two different scenarios are studied for the eccentricity analysis: changing eccentricity with a perigee altitude of 200 km and 600 km respectively. It has been observed a dependency between the uncertainty propagation accuracy and the orbital altitude, related to the aerodynamic acceleration. By selecting a set of orbits with perigee altitude of 600 km, it is possible to individually study the effect of changing the eccentricity in an altitude range that escapes this effect. However, it is still interesting to study how the change in orbital shape is correlated with the error derived from low altitudes. It is expected that for orbits with perigee altitude of 200 km, increasing the eccentricity would decrease the L_2 error, since a larger part of the orbit would be at higher altitudes. By selecting this second scenario, this hypothesis can be tested.

For every scenario, five values of eccentricity are selected, with the aim to test the method for the limiting cases and adding two intermediate cases that allow to study the evolution of the performance parameter S_{L_2}. For the lower threshold, a circular orbit is considered and taken as the baseline parameter. For the upper threshold, the Molniya orbit is taken as reference, with an eccentricity of 0.74. Since orbits with such high eccentricity are uncommon in LEO, the intermediate values selected for the eccentricity are: $0.05, 0.1$ and 0.2.

For the orbits with perigee altitude of 600 km, it is assumed that the results are uncorrelated with the error related to the atmospheric acceleration. This is a valid assumption since, according to Table 8.3 the effect of altitude change after 600 km altitude is smaller than the variations observed in eccentricity. In Table 8.4 it can be observed that for a perigee altitude of 600 km, the L_2 error increases with eccentricity. Up to an eccentricity of 0.2, the error remains in the order of 10^{-2} while for the Molnyia orbit it increases to 0.32. The reason behind this behaviour is the change in integration time related to the change in orbital period. Since every case is propagated for five orbits, the increase in semi-major axis results in an increase in the integration period, which affects the accuracy as further analysed in Subsection 8.3.1. The magnitudes observed for the change in eccentricity are consistent with the results obtained for circular orbits with the equivalent integration time. Therefore, it is concluded that eccentricity, as an isolated variable, does not influence the accuracy of the method.

Regarding the scenario with perigee altitude of 200 km, it is confirmed that the large error found for the circular orbit rapidly decreases when increasing the eccentricity. With an eccentricity of 0.05, the apogee altitude rises to 892 km and the portion of the orbit at lower altitudes decreases. After this initial decrease in error, the increasing error with increasing integration time trend becomes dominant again. It is observed that for $e = 0.1, 0.2, 0.7$ the error the scenarios with $h_p = 200$ km is approximately double the error when $h_p = 600$ km. This suggests that even with a very high apogee, the low altitude passage will cause a significant error.

8.2.3. Change in inclination

Beforehand, orbital inclination is not expected to have a large effect on the method accuracy. However, there are some magnitudes that depend on the orbital orientation, such as the value of the spherical harmonic coefficients and the atmospheric density, which does not only depend on the altitude. Since small errors in position are known to greatly effect the error in density, it is decided to briefly investigate the effect of inclination through five sample points. First, the inclinations of 0 deg and 90 deg are selected to verify that the method does not present any singularities. An inclination of 98 deg is included to test the method in retrograde orbits. Finally, an inclination of 60 deg is selected as the baseline, and an inclination of 23 deg

[1] Not normalised by nominal eccentricity to avoid division by zero.

Table 8.4: L_2 error for changing eccentricity. Shaded cells indicate simulations that violate the threshold for accuracy.

Eccentricity [-]	**0**	**0.05**	**0.1**	**0.2**	**0.74**
Perigee altitude of 200 km					
Integration time	7h 22’	7h 57’	8h 38’	10h 18’	55h 37’
L_2 with Monte Carlo	1.3775	0.0457	0.1274	0.1303	0.7612
S_{L_2}[1]	-	-19.3	-9.1	-4.5	-0.6
Perigee altitude of 600 km					
Integration time	8h 3’	8h 42’	9h 26’	11h 15’	60h
L_2 with Monte Carlo	0.0111	0.0315	0.0576	0.0681	0.3206
S_{L_2}[1]	-	27.2	31.0	10.0	27.9

Table 8.5: L_2 error for changing inclination.

Inclination [deg]	**0**	**23**	**60**	**90**	**98**
L_2 with Monte Carlo	0.015	0.0519	0.0199	0.0047	0.0139
S_{L_2}	0.2459	-2.6041	-	-1.5293	-0.4761

completes the set.

Table 8.5 summarises the results of the sensitivity analysis for the change in inclination. It is observed that the L_2 distance ranges from 0.01 to 0.037 for all cases. Taking into account the limitations of this L_2 calculation derived from the number of samples in the Monte Carlo simulation, these differences do not allow to infer any correlations between accuracy an orbital inclination. Focusing on the S_{L_2} parameter, there is no clear trend linking the accuracy values with the inclination. Therefore, it is concluded that the effect of this parameter on the accuracy of the results is negligible.

8.3. Effect of DA-GMM model parameters

This section examines the effect of model parameters, also referred to as model settings, on the accuracy of the results. Although the number of cases studied does not allow to provide the optimal settings for every geometry, this analysis intends to offer recommendations depending on the desired accuracy and available computational load. First, the effect of increasing the integration time is studied. This is a crucial step since it will determine the lead time for the encounter simulation. Second, the performance of the GMM method is tested by increasing the number of GMEs. The impact on computational time is stored for analysis. Third, the differential algebra method is analysed by changing the Taylor expansion order. Finally, the effect of integration tolerance is studied.

8.3.1. Lead time

During the uncertainty propagation, the Taylor expansion truncation error and the numerical integration error accumulate with time. It is crucial to investigate this effect and understand the limits of the DA-GMM method in terms of integration period, which directly effects the maximum lead time achievable for the collision probability calculation. The evolution of error with integration time will also depend on the simulated trajectory (mainly on the orbital altitude), as can be concluded from Section 8.2. Since it is not possible to conduct the analysis for every geometry, the three nominal cases are taken as reference. It must be noted that these cases were originally selected to cover a wide range of orbital parameters which are closely related to the reference mission.

Each scenario is propagated during 1, 5, 15, 30 and, 100 orbital periods, which cover a range of integration times from 2 hours up to 10.5 days. Taking into account that the second phase of collision warning, the Conjunction Approach Notification, is sent 3 days before TCA, achieving a lead time of 30 orbits (equivalent to 2 days for Scenario #2 and >3 days for Scenario #3) would allow to evaluate the risk of collision considering multiple scenarios and refining the estimation with newly acquired data.

In Table 8.6, the results for each scenario propagated with 1 and 201 GMEs are studied. The first observation made is that increasing the number of Gaussian mixture elements improves the accuracy in all the cases, leading to longer valid trajectory integration. For Scenario #1, this entails increasing the maximum computa-

Table 8.6: L_2 error for changing propagation time. Shaded cells indicate simulations that violate the threshold for accuracy.

Orbital periods [-]		**1**	**5**	**15**	**30**	**100**
Nominal Scenario #1						
L_2 with Monte Carlo	1 GME	0.0694	0.0861	0.2981	0.6456	1.0499
	201 GME	0.0033	0.0199	0.1586	0.4694	0.8432
S_{L_2}	1 GME	0.1942	-	0.9854	1.0401	0.4715
	201 GME	0.8336	-	2.7889	3.6142	1.7419
Nominal Scenario #2						
L_2 with Monte Carlo	1 GME	0.0902	0.1071	0.4238	1.0434	1.2340
	201 GME	0.0394	0.0446	0.3778	0.9132	0.9736
S_{L_2}	1 GME	0.1579	-	1.1821	1.3980	0.4428
	201 GME	0.1175	-	2.9879	3.1159	0.8769
Nominal Scenario #3						
L_2 with Monte Carlo	1 GME	0.1031	0.1653	1.1487	1.2236	1.2308
	201 GME	0.0016	0.0260	0.5495	0.5607	0.9108
S_{L_2}	1 GME	0.3764	-	2.3795	1.0243	0.2714
	201 GME	0.9393	-	8.0687	3.2968	1.4355

tional time from 15 orbits to 30 orbits. For Scenario #2, although the error is still smaller with 201 GMEs, the difference between both settings is smaller and with the selected propagation times, the increase in lead time is not visible and the limiting integration time is set at 15 orbital periods. This result is consistent with the previous analysis, since Scenario #2 corresponds to the ISS orbit at a 422 km altitude. Therefore, this scenario is affected by the error related to the atmospheric drag, resulting in a lower lead time. Regarding Scenario #3, the behaviour observed is similar to Scenario #1. In this case, the orbital period is longer, which explains the larger L_2 error for the same number of orbital periods integrated.

It is interesting to study the evolution of S_{L_2} as the propagation time increases. Firs, it is observed that this parameter is always positive, which confirms that the error always increases with increasing integration time. The interesting result is that S_{L_2} does not present an increasing trend with integration time. In all the scenarios, S_{L_2} for integration time of 100 orbits is lower than for integration times of 15 and 30 orbits. This implies that the evolution of the L_2 error with time is not linear. A direct consequence of this is that an improvement of X% in the L_2 error for 5 orbits will produce an increase the lead time > X%.

This analysis verifies that the DA-GMM method is valid to propagate the collision probability for > 60 hours when the orbital altitude is ≥ 800 km and for at least 30 hours for an altitude of 400 km.

8.3.2. Number of GMEs

The number of Gaussian mixture elements is a driver parameter on the accuracy and computational load of the DA-GMM method for uncertainty propagation. To study its effect, the nominal scenarios are propagated with $N = 1, 37, 51, 101, 201, 501$ GMEs. From the Gaussian hypothesis testing performed in Subsection 5.1.3 it was concluded that for 201 Gaussian mixture elements, the error with respect to a propagation with 2001 GMEs was $L_2 < 5 \cdot 10^{-7}$. Therefore, this number is selected as the nominal scenario. Due to the quadratic increase in computational load with respect to an increase in number of GMEs, 501 elements is selected as the upper bound for this analysis. As observed in Figure 5.6, the rate of error reduction with number of GMEs is very high for lower numbers of elements and becomes steadier as the elements increase. For this reason, the L_2 error is studied for 37, 51 and 101 sample points. The case with 1 sample point (i.e., Gaussian uncertainty propagation) is also added for comparison.

The results are summarised in Table 8.7. First, it is observed that for all scenarios, the largest reduction in error occurs when increasing the number of Gaussian mixture elements from 1 to 37. This leads to an improvement of one order of magnitude, which makes a large difference in the propagated uncertainty and in the potential to increase the lead time for the simulation. For scenarios #1 and #3, the difference in accuracy resulting from selecting 37 with respect to 201 GMEs remains significant and the error decreases when increasing to 51 and 101 GMEs respectively. By inspecting the sign of S_{L_2} it is concluded that this tendency

Table 8.7: L_2 error for change in number of GMEs. Nominal setting of 201 GMEs.

Nº GMEs	1	37	51	101	201	501
Nominal Scenario #1						
L_2 with Monte Carlo	0.0861	0.0254	0.0206	0.0199	0.0199	0.01758
S_{L_2}	-3.3266	-0.3371	-0.0469	$< 10^{-8}$	-	-0.0777
Nominal Scenario #2						
L_2 with Monte Carlo	0.1971	0.0465	0.0435	0.0404	0.0446	0.0446
S_{L_2}	-3.4193	-0.0520	0.0329	0.1883	-	$< 10^{-8}$
Nominal Scenario #3						
L_2 with Monte Carlo	0.1653	0.0334	0.0285	0.0261	0.0260	0.0255
S_{L_2}	-5.3577	-0.3471	-0.1282	-0.0077	-	-0.0128

does not occur in Scenario #2. In this case, the error obtained with 201 and 501 elements is larger than the error obtained with 101 elements. This can be related to the orbital parameters of this scenario and the resultant deviation in the drag acceleration. Imagine the error in acceleration as a *push* to the position mean of each Gaussian mixture element in a certain direction. Each mean will receive a different *push* depending on the location of the mean and this will change the direction of the propagated distribution with respect to a case without error. For a large number of Gaussian mixture elements, this effect accumulates as it occurs on every element, explaining the observed trend. This effect is analogous to the scattering of the space debris after a collision. Every piece of debris is initially in the same location, but receives a different velocity after the impact, which sets the pieces in different orbits. This effect propagates with time. Therefore, for cases with low altitudes (e.g., 300 - 600 km) it is counter-productive to select a larger number of elements.

Focusing on the magnitude of L_2 it is noticed that for 501 elements it is in the order of 10^{-2} in all cases, which questions the need to select more elements when considering the associated increase in computational load. Effectively, the difference between selecting 101 and 201 elements becomes insignificant for Scenarios #1 and #3, which renders impractical to select more than 101 GMEs.

Although in Scenarios #1,#3 the evolution of error with number of GMEs follows a slightly different trend compared to Scenario #2, due to the effect of atmospheric drag, a single conclusion can be made for all scenarios. Selecting more than 101 elements does not provide major benefits in terms of accuracy and it causes a large increase in computational load. Therefore, it is recommended to select 51 elements and increase to a maximum of 101 if a high accuracy or a long integration time are required.

8.3.3. Taylor series expansion order

The main parameter controlling the differential algebra computation is the Taylor series expansion order. This parameter sets the truncation order of the computation and therefore the accuracy of the Taylor expansion of the final state from a disturbance in the initial state. Moreover, this parameter has a large impact on the computational load, both during the integration process and during the post-processing of the Taylor expansion coefficients to obtain the final distribution. In this section, the effect of changing the Taylor expansion order on the accuracy of the propagated distribution is analysed, for the three nominal scenarios. Although the DACE software allows to perform the computation with very high expansion order (e.g., order 20), the computational load is a limiting factor that allows to use a maximum order of four. Therefore, the study is performed for Taylor series expansion order ranging from one to four.

Table 8.8 collects the L_2 error evolution with a change in Taylor series expansion order. First, it must be noted that in all cases the error has an order of magnitude of 10^{-2}, which is below the threshold set for a valid propagation. However, this analysis has been conducted for an integration of 5 orbits. As observed in Table 8.6, when $L_2 \approx 0.02$ after 5 orbital periods, the uncertainty propagation is still valid after 30 orbits. However, the cases with $L_2 \geq 0.04$ after 5 orbital periods fail before 30 orbits. Selecting an expansion order $k = 1$ with 201 GMEs has a similar impact on the accuracy as selecting a Gaussian propagation (1 GME). However, the reduction in computational load is negligible in comparison to the latter. Therefore, this option is discarded and will not be studied further.

The error shows a decreasing trend with increasing expansion order, as expected. However, analysing the S_{L_2} parameter it cannot be extracted a clear relation between the amount of change and the expansion

Table 8.8: L_2 error for change in Taylor expansion order. Nominal setting of $k = 3$.

Taylor expansion order	**1**	**2**	**3**	**4**
Nominal Scenario #1				
L_2 with Monte Carlo	0.0505	0.0205	0.0199	0.0152
S_{L_2}	-1.5377	-0.0603	-	-0.4724
Nominal Scenario #2				
L_2 with Monte Carlo	0.0896	0.0448	0.0446	0.0213
S_{L_2}	-1.0090	-0.0090	-	-1.0448
Nominal Scenario #3				
L_2 with Monte Carlo	0.0301	0.0284	0.0260	0.0292
S_{L_2}	-0.1577	-0.1846	-	0.2462

Table 8.9: L_2 error for change in integration tolerance. Nominal setting of $\epsilon = 10^{-12}$.

Tolerance	$\mathbf{10^{-8}}$	$\mathbf{10^{-10}}$	$\mathbf{10^{-12}}$	$\mathbf{10^{-14}}$
Nominal Scenario #1				
L_2 with Monte Carlo	0.0205	0.0203	0.0199	0.0197
S_{L_2}	-0.0302	-0.0402	-	-0.0201
Nominal Scenario #2				
L_2 with Monte Carlo	0.0446	0.0446	0.0446	0.0446
S_{L_2}	$< 10^{-8}$	$< 10^{-8}$	-	$< 10^{-8}$
Nominal Scenario #3				
L_2 with Monte Carlo	0.0286	0.0287	0.0260	0.0260
S_{L_2}	-0.1000	-0.2077	-	$< 10^{-8}$

order. For Scenarios #1 and #2, improvement from $k = 3$ to $k = 4$ is larger than the improvement from $k = 2$ to $k = 3$. However, for Scenario #3 the error actually increases when setting $k = 4$. Therefore, with this analysis it cannot be clearly stated that setting $k = 4$ will provide the best performance in every case. This will be further analysed in Subsection 8.3.5. For these scenarios, and considering the large jump in computational load required to increase from k_3 to $k = 4$, it is recommended to limit the settings to $k = 2, 3$.

8.3.4. Integration tolerance

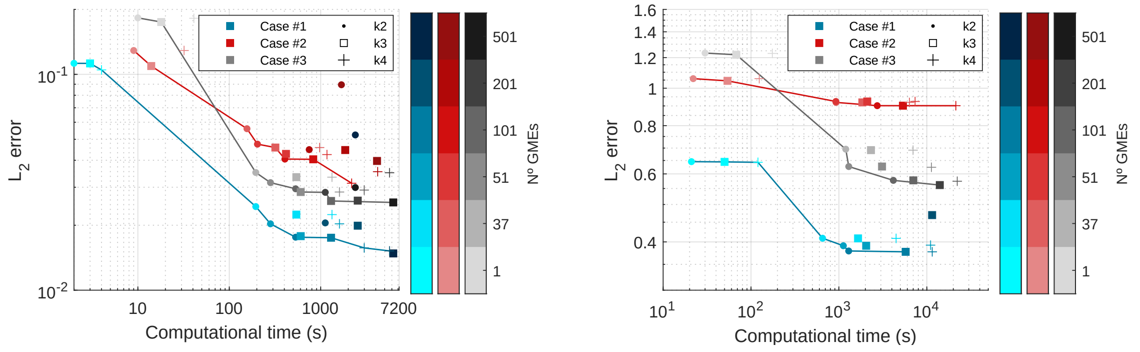
Finally, the effect of the tolerance for the numerical integration algorithm is studied. The objective of this analysis is to select a value that is suitable for any orbit in LEO. Since the number of cases that can be tested is limited, a conservative approach is taken while aiming to keep a low computational time.

For similar problems, the range of integrator tolerances that have been studied is $\epsilon = [10^{-9} \;\; 10^{-14}]$ (Valli et al., 2013; Jones and Doostan, 2013). Using this as a reference, and taking into account that the default setting for integration tolerance provided in the DACE software is 10^{-12}, the analysis will be performed for $\epsilon = 10^{-8}, 10^{-10}, 10^{-12}, 10^{-14}$.

The L_2 errors for the nominal scenarios with these settings are provided in Table 8.9. First, it is observed that the change in L_2 obtained by changing the integrator tolerance is very small compared to the variation of other parameters. Based on the three cases studied, it is expected that this parameter has a low sensitivity on the results. The general trend observed is a reduction of the error for stricter tolerances. The computational time required for tolerances of $10^{-8} - 10^{-12}$ is similar, while for a tolerance 10^{-14} there is a jump in computational time of one order of magnitude. Coupled with the fact that the improvement observed by reducing the tolerance from 10^{-12} to 10^{-14} is very small or even negligible, the setting of $\epsilon = 10^{-12}$ is selected.

8.3.5. Combined analysis

The analysis conducted so far provides a general idea of the effect of several orbital and model parameters on the accuracy of the DA-GMM method. It is concluded that from the orbital geometry, the altitude plays



(a) Uncertainty propagated for 5 orbital periods.

(b) Uncertainty propagated for 30 orbital periods.

Figure 8.3: Pareto front for L_2 accuracy against computational time for the uncertainty propagation of Nominal Scenarios #1,#2 and #3 with Taylor expansion order $k = 2, 3, 4$.

an important role in the accuracy of the results while the other variables do not have a significant effect. From the model parameters, the results match the expected behaviour. The model accuracy decreases with an increase in lead time.This effect can be counteracted by an increase in the Taylor expansion order and the number of Gaussian Mixture Elements. Since all these parameters are connected, a combined analysis is performed for the combination of parameters using the nominal scenarios described in Section 8.1. To account for the effect of lead time, two cases are considered by simulating the uncertainty propagation with an integration time of 5 and 30 orbital periods respectively.

Figure 8.3 illustrates the results of this combined analysis and presents the Pareto front of optimal settings for each scenario. The performance parameters considered for this analysis are computational accuracy as described by the L_2 error and computational time. The latter includes the time required for the Taylor series integration in the DA software and the reconstruction of the final state uncertainty. By analysing the Pareto lines of optimal results it is observed that for every scenario studied, nominal scenario #1 provides the best results. The main factor difference between nominal scenarios comes from the change in orbital altitude.

Focusing on the Taylor expansion order, it is observed that the optimal setting depends on whether the computational time or the accuracy are prioritised. If the computational time is prioritised, the optimal Pareto results are found for simulations with $k = 2$. For low integration times this setting provides an accuracy below the threshold of $L_2 = 0.6$ in all cases and therefore it is preferred. When moving to longer integration periods (e.g., 30 orbits as observed in Figure 8.3b) longer computational times are required to satisfy the accuracy threshold. For the settings tested, the threshold accuracy is not always met for long integration periods. If the accuracy is prioritised, it is required to increase the Taylor expansion order with the associated increase in computational time. Although an increase in Taylor expansion order is usually linked to a better accuracy, this does imply that a Taylor expansion order of four offers the optimal solution. Inspecting the optimal Pareto fronts it is observed that for higher accuracy the Pareto front is composed by Taylor expansion order three, since it is more efficient to increase the number of Gaussian Mixture Elements. Based on these results, it is recommended to use a Taylor expansion order $k = 2$, when computational load is a priority and $k = 3$, when accuracy is a priority or for long integration times.

Regarding the number of Gaussian mixture elements, represented by the colour of the markers, the conclusion from the individual analysis performed in Subsection 8.3.2 is confirmed. First, it is observed that going form a Gaussian uncertainty propagation to a GMM has the largest impact, both in accuracy and computational time. Therefore, if the integration time is low (less than ten hours), it is reasonable to prioritise computational time and perform a Gaussian uncertainty propagation. However, if a high accuracy is desired or the integration time is long it is required to use a Gaussian Mixture Model. Increasing the number of elements from 37 elements has a larger effect on the computational time than on the accuracy, as observed by the decrease in slope of the Pareto fronts. Therefore, in this case the recommendation of using 51 elements is kept.

Although every case is unique and this analysis cannot determine the optimal settings for every possible encounter geometry, it provides more insight on the interactions between settings. For the following analysis it is decided to select a Taylor expansion order of $k = 3$ and 51 Gaussian Mixture Elements, to stay in the upper bound of recommended settings that allow to obtain a good performance for a general scenario.

Table 8.10: L_2 error for change in initial state uncertainty with $k = 3$ and 201 GMEs.

Factor	**0.1**	**1**	**100**
Nominal Scenario #1			
L_2 with Monte Carlo	0.1940	0.0199	0.1178
S_{L_2}	-12.8046	-	0.0671

8.4. Effect of uncertainty

So far, the effect of the orbital parameters and the method settings on the accuracy of the propagated distribution have been analysed. There is one last parameter involved in the uncertainty propagation, which is the magnitude of the initial uncertainty. This section analyses the effect of the initial uncertainty for the position and velocity of the satellite and describes the expected effect of the uncertainty on the ballistic coefficient and atmospheric density.

8.4.1. Initial state uncertainty

The initial state uncertainty plays two roles on the DA-GMM method that could affect the accuracy of the resultant uncertainty propagation. First, the magnitude of the uncertainty determines the spread of the mean values for the position of the Gaussian mixture elements. For a larger initial state uncertainty and the same number of Gaussian mixture elements, the problem scales up and it becomes equivalent to having the smaller initial state uncertainty with a lower number of elements. Therefore, it is expected that increasing the initial state uncertainty will increase the error, but this can be avoided by increasing the number of elements. Second, the initial covariance is used by the Hafnian operator to obtain the uncertainty distribution of the final state with respect to the initial state. This process involves several products between the covariance components. If the covariance (of the Gaussian Mixture element) is very small, this can introduce numerical errors. In this case, it is recommended to select a lower number of GMEs.

To study the effect of a change in the initial state uncertainty, the nominal values for the standard deviation displayed in Table 8.2 are multiplied by a factor of 0.1 and 100. This covers initial state uncertainties in the range of 40 cm to 400 m, which covers the range of accuracy reached with satellite orbit determination techniques. Table 8.10 presents the results of this analysis for Scenario #1. The error in the propagated uncertainty increases for both deviations from the nominal initial state uncertainty, while staying within the acceptable region. Inspecting the S_{L_2} parameter it is observed that the effect is more pronounced for a decrease in initial state uncertainty than for an increase. From a practical point of view this is a positive result, since the error can be bypassed by selecting a smaller number of Gaussian mixture elements. Moreover, the typical accuracy reached for LEO is in the order of 100 meters, which can improve up to 3 cm if advanced tracking techniques are used Rongzhi and Kaizhong (2020a). This combination of factors allows to conclude that the method is valid for current operational scenarios.

8.4.2. Ballistic coefficient and atmosphere density uncertainty

As discussed in Section 4.3, the initial state is not the only variable of the dynamical model that is known with uncertainty. The two main parameters that can be included in the uncertainty model are the atmospheric density and the ballistic coefficient. The atmospheric density for a given location fluctuates with time, presenting variations of two orders of magnitude in the LEO region. Although these time variations have been modelled by the NRLMSISE-00 atmosphere, there are still discrepancies with respect to the measured data. Picone et al. (2002) concluded that the modelled atmospheric density can be assumed to have a standard deviation equal to 25% of the measured density. Regarding the ballistic coefficient, it combines the uncertainty on the mass, area and drag coefficient. The dynamical model presented in this work does not include the orientation of the satellite, in the orbit. This affects the effective area in the velocity direction relative to the atmosphere and therefore the drag acceleration. Therefore, both parameters are related to the drag acceleration.

The DA-GMM model has been developed with the possibility to add an uncertainty in these parameters. The uncertainty in the atmospheric density is modelled as a lognormal distribution, while the ballistic coefficient is assumed to follow a uniform distribution. To understand the effect of these parameters on the propagated uncertainty, a sensitivity analysis could be performed on these parameters. However, from the analysis performed in Section 8.2, it was concluded that a change in the atmospheric acceleration (in this case due to a numerical error), had a large effect on the error in the propagated uncertainty for low altitudes, but became acceptable for altitudes of 400 km and negligible after 800 km altitude. These results can be extrapolated to a change in atmospheric acceleration due to an imposed uncertainty instead of a numerical error. Following this reasoning and due to time constraints, it is decided to exclude the sensitivity analysis on these parameters.

8.5. Summary of findings

In this chapter, a detailed analysis of the uncertainty propagation for a range of orbit and model parameters. The results of the propagation of uncertainty with the DA-GMM method are verified, by computing the L_2 error with respect to the Monte Carlo analysis. The results from this analysis allow to identify the limitations of the methods and the recommended settings for different types of scenarios, which are hereby summarised.

- For very low altitudes (200 km), the error in uncertainty propagation is above the allowed threshold due to the larger effect of the atmosphere perturbation.
- For 400 km altitude, the method is already one order of magnitude more accurate than the threshold value. The method has a very high accuracy for LEO and MEO.
- Orbit eccentricity and inclination are not found to have a highly significant effect on the accuracy. For low perigee altitudes, higher eccentricity provides better results since a larger part of the orbit is spent at higher altitudes. Once this effect is bypassed by setting higher perigee altitudes, no correlation is observed.
- For inclination, no significant correlation with accuracy is observed.
- With a sufficient number of GMEs, a propagation time of 60 hours can be achieved for altitudes > 600 km. For orbits at 400 km, at least 30 hours are achieved with high accuracy.
- The accuracy increases by one order of magnitude when using a Gaussian Mixture Model with respect to Gaussian uncertainty propagation. This allows to double lead time with the same accuracy.
- Using 37-51 GMEs is sufficient to obtain a very high accuracy. Adding more elements increases the computational load without additional benefits in terms of accuracy.
- Using a Taylor expansion order of four is computationally unfeasible, and and expansion order one is insufficient in most cases. It is recommended to use an expansion order of two for applications where computational time is prioritised and an order of three when accuracy is prioritised.
- All the range of characteristic initial state uncertainty can be used in the model. However, an initial state uncertainty of one meter offers the best accuracy.
- Ballistic and atmosphere density uncertainty have been adapted to the model but have not tested in this sensitivity analysis due to the low effect for higher altitudes concluded from the altitude analysis and the time constraints.

9

Results

With the DA-GMM method for collision probability calculation fully developed and verified, it can be applied to several test cases to study the risk of past and present encounter events and determine whether the method supposes an improvement with respect to existing techniques. To do so, a combination of fictitious and real scenarios is analysed. Section 9.1 focuses on two test cases for which the collision probability with an accurate Monte Carlo analysis and alternative techniques has been reported in literature. This allows to assess the accuracy of the method developed in unique encounter scenarios. To evaluate the effect of the initial state uncertainty and lead time, Section 9.2 studies a fictitious case where the nominal trajectories lead to a collision. The outcome from this analysis are useful to interpret the results of real-life scenarios.

Section 9.3 documents the catastrophic encounter between satellites Cosmos-2251 and Iridium-33. The effect of the dynamics model, number of Gaussian mixture elements and lead time in the estimated risk is analysed. Following, the close encounter between the Chang Zheng-4C rocket body and Cosmos-2004 is analysed in Section 9.4. This encounter did not lead to a collision despite the very high risk predicted. It is interesting to research how the results with the DA-GMM method compare to these predictions and the real outcome of the encounter. Finally, Section 9.5 introduces the screening of close encounters between the ISS and the debris resulting from the Cosmos-1408 destruction. For the detected approaches, the collision probability is calculated with the single-sphere method, and the highest risk encounter is then simulated with the multi-sphere method to test the effect of modelling the real shape of the satellite. The summary of findings from this chapter is presented in Section 9.6.

9.1. Collision probability verification

The accuracy of a collision probability calculation method is typically assessed by comparing the results with Monte Carlo simulations. However, the accuracy of a Monte Carlo analysis is highly dependent on the number of samples simulated. To predict a collision probability of 10^{-N}, a number of at least 10^{N+1} samples are required. As explained, the number of samples that can be propagated with the available computational resources is limited to 10,000, which would only allow to predict, with a low accuracy, a collision probability of $P_c \geq 10^{-3}$. Instead, a database of satellite conjunction with Monte Carlo analysis performed by Alfano (2009) is used for verification. Alfano studied 12 different cases for close encounters at LEO, MEO, HEO and GEO with 10^8 Monte Carlo samples and considering only the central gravitational acceleration. Generating such a set of samples with the astrodynamics propagation software used in this work considering the full dynamics with perturbations would take more than one year, due to the limitations in available computational capacity. Considering that the two-body problem presents an analytical solution, it is expected that the Monte Carlo samples for this case can be obtained in a more reasonable computational time. However, since for each satellite the position is evaluated at 2840 time stamps, it is still expected that this computation will be time consuming. Alfano (2009) compared the results from five alternative collision probability calculation methods. These methods are the ones included in the NASA-CARA and ESA-CORAM software to compute the collision probability in non-linear encounters. By selecting these scenarios, the DACE-GMM method of collision probability can be verified and its accuracy can be compared to that of the main approaches currently used. In this section, the cases are first described and the input data for the simulations is presented. Finally, the results obtained with the DA-GMM method are provided and compared to the verification data.

Table 9.1: Verification scenarios: initial state (Alfano, 2009, p.34, p.49).

Initial Position		**x [km]**	**y [km]**	**z [km]**
Case 7	Satellite A	6337.9665365122	1889.3098033002	1889.3098033002
	Satellite B	6338.6705812339	1888.2177447937	1888.1503615517
Case 12	Satellites A & B	6878.137	0.0	0.0
Initial Velocity		**ẋ [km]**	**ẏ [km]**	**ż [km]**
Case 7	Satellite A	-2.9571994197397	4.9601786006789	4.9601786006789
	Satellite B	-2.9552934327179	4.9608101452466	4.9606245630827
Case 12	Satellites A & B	0.0	7.6126081732239	0.0

Table 9.2: Verification scenarios: initial uncertainty (Alfano, 2009, p.34, p.49).

Covariance	**Case 7, Sat A**	**Case 7, Sat B**	**Case 12, Sat A & B**	**Units**
$\sigma_{x,x}$	$3.4099738463705 \cdot 10^{-4}$	$3.4107282833791 \cdot 10^{-4}$	$8.578274700986899 \cdot 10^{-6}$	$[\mathbf{km}^2]$
$\sigma_{y,y}$	$2.3400130768148 \cdot 10^{-4}$	$2.3395737435097 \cdot 10^{-4}$	$4.4217252990131 \cdot 10^{-6}$	
$\sigma_{z,z}$	$2.3400130768148 \cdot 10^{-4}$	$2.3396979731112 \cdot 10^{-4}$	$4.0 \cdot 10^{-6}$	
$\sigma_{x,y}$	$9.8966443775773 \cdot 10^{-5}$	$9.8916238832886 \cdot 10^{-5}$	$-1.3895230358788 \cdot 10^{-6}$	
$\sigma_{x,z}$	$9.8966443775773 \cdot 10^{-5}$	$9.8912538411159 \cdot 10^{-5}$	0.0	
$\sigma_{y,z}$	$-1.6599869231852 \cdot 10^{-4}$	$-1.6603641405277 \cdot 10^{-4}$	0.0	
$\sigma_{\dot{x},\dot{x}}$	$8.5060718872296 \cdot 10^{-11}$	$8.5079820985814 \cdot 10^{-11}$	$1.0 \cdot 10^{-14}$	$\left[\frac{\mathbf{km}^2}{\mathbf{s}^2}\right]$
$\sigma_{\dot{y},\dot{y}}$	$5.7969640563852 \cdot 10^{-11}$	$5.7958516779401 \cdot 10^{-11}$	$1.0 \cdot 10^{-14}$	
$\sigma_{\dot{z},\dot{z}}$	$5.7969640563852 \cdot 10^{-11}$	$5.791662234785 \cdot 10^{-11}$	$1.0 \cdot 10^{-14}$	
$\sigma_{\dot{x},\dot{y}}$	$2.5057999830694 \cdot 10^{-11}$	$2.5045288093237 \cdot 10^{-11}$	0.0	
$\sigma_{\dot{x},\dot{z}}$	$2.5057999830694 \cdot 10^{-11}$	$2.5044351157813 \cdot 10^{-11}$	0.0	
$\sigma_{\dot{y},\dot{z}}$	$-4.2030359436148 \cdot 10^{-11}$	$-4.2039910463489 \cdot 10^{-11}$	0.0	

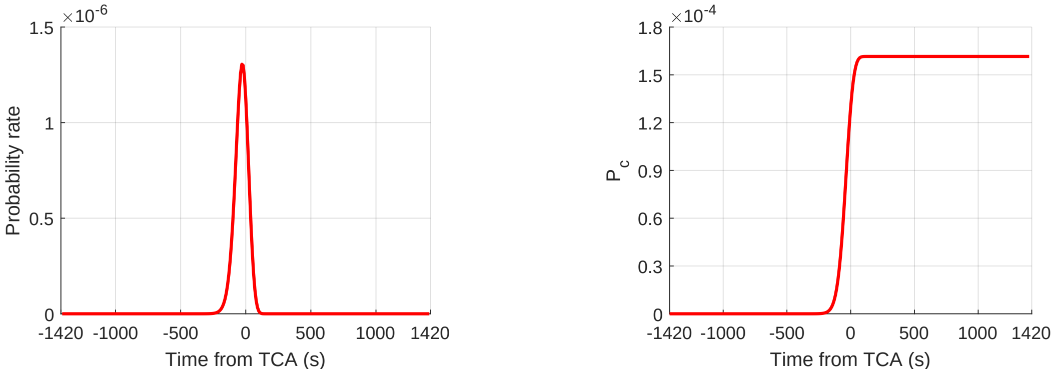
9.1.1. Mission description and initial state

From the cases studied by Alfano, cases 7 and 12 present special properties and are therefore selected. Case 7 involves the relative motion between two satellites in LEO which results in a low collision probability of $P_c \approx 10^{-4}$. This allows to test and compare the accuracy of the method to predict collisions with a very low risk. It must be noted that 10^9 Monte Carlo samples are required to compute this probability with sufficient accuracy. Case 12 presents a peculiar scenario of two satellites co-located in identical LEO orbits and with the same initial state uncertainty. The propagated state and uncertainty will be identical for both satellites, but a collision does not necessarily occur. This is a special event, since the encounter develops continuously. The collision probability methods selected by Alfano (2009) for comparison do not offer a good result for this scenario and therefore it is a good challenge for the algorithm developed in this work.

Table 9.1 presents the initial position and velocity of the satellites involved in the two cases selected. The initial state uncertainty is introduced in Table 9.2. It must be noted that the covariance elements excluded from the table are set to zero and that the covariance matrix is symmetric (i.e., $\sigma_{x,y} = \sigma_{y,x}$).

9.1.2. Simulation parameters

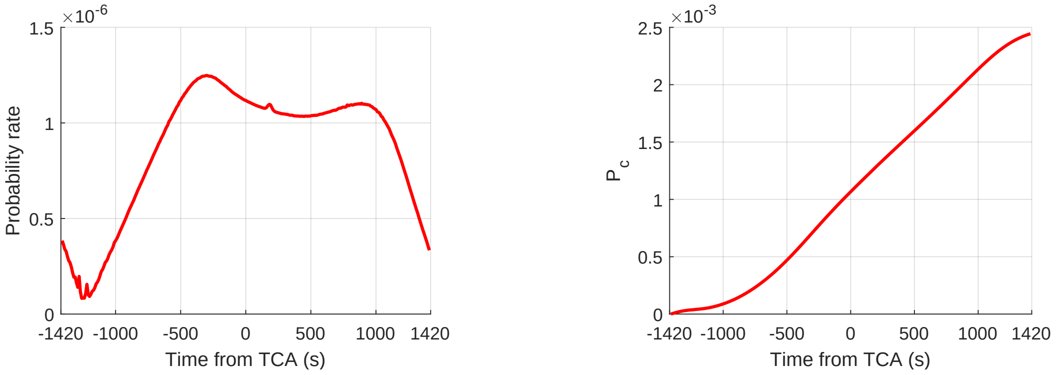
In the simulations run by Alfano (2009), the encounter time is set from TCA −1420 s to TCA+1420 s. To be consistent with the results from the Monte Carlo analysis, the same time span is selected for the collision probability calculation with the DA-GMM model. According to the results from the sensitivity analysis performed in Chapter 8, and considering that the collision probability has to be integrated over a long period, the simulations are performed with Taylor expansion order three. The main drivers for the computational load on the P_c calculation process using the time integration method are the number of GMEs and the number of integration steps (which depend on the encounter period and the time step). Following the recommendations from Chapter 8, a maximum of 51 elements is selected for the Gaussian mixture model. A second analysis with 37 GMEs is performed to test the sensitivity of the method. The time step is set to 10 seconds.



(a) Collision probability rate over time

(b) Cumulative collision probability over time

Figure 9.1: Evolution of collision risk for Case 7 with 51 GMEs.



(a) Collision probability rate over time

(b) Cumulative collision probability over time

Figure 9.2: Evolution of collision risk for Case 7 with 51 GMEs.

9.1.3. Results and discussion

With the specified settings, the DA-GMM method is run to compute the collision probability in Cases 7 and 12. This method not only provides the total collision probability, but also the evolution of the collision probability rate and the cumulative collision probability over time. This provides more information than simply outputting the risk as a single number. In Figure 9.1, the evolution of the risk for Case 7 with a lead time of 48 hours is presented. As observed, the real duration of the encounter is only about 300 seconds which is relatively small with respect to the 2840 second window studied. On Figure 9.2a it is observed that in Case 12, the risk of encounter spans through the entire time studied, which is expected since both satellites follow the nominal trajectory. Although the collision probability obtained for Case 12 is one order of magnitude higher than that of Case 7, the probability rate is in the same order of magnitude in both cases. This hints that only looking at the total collision probability might be insufficient in some scenarios. The results obtained are verified with the results from a Monte Carlo analysis, see Alfano (2009, Fig A7, Fig A12).

The interesting outcome of this analysis is that it allows to compare the final collision probability with the values obtained with the main conventional methods and to verify the results with the results of an extensive Monte Carlo simulation. Before discussing the results, it is worth to recapitulate the main characteristics of the alternative methods. The linear Alfano (Alfano, 2005) and linear Patera (Patera, 2005) assume that the relative velocity between the objects is constant over the collision interval, and that there is no error in the velocity. This allows to compute the collision probability as a two-dimensional integral, following different computational methods. These techniques are fast, but result in a low accuracy in long-term encounters with low relative velocity. On the other hand, the three-dimensional methods allow to introduce the effect of nonlinear motion: the voxels, the adjoining cylinders and the parallelepipeds. In these cases, each volume (e.g., cylinder) is broken into slices where the linear probability can be computed and its product is multiplied with the one-dimensional integral of the collision probability along the relative velocity. This process is highly time consuming, slower than real time and therefore only recommended for determining reference cases (Alfano, 2006).

Table 9.3: Results of collision probability calculation for reference methods and DACE-GMM (Alfano, 2009, p.32, p.48). Shaded cells represent results from the DA-GMM method that improve the accuracy compared to alternative methods analysed by Alfano.

Method	$\mathbf{P_c}$	$\Delta\mathbf{P_c}$(%)	$\mathbf{P_c}$	$\Delta\mathbf{P_c}$(%)
	Case 7		Case12	
Monte Carlo (10^6 samples)	$1.51 \cdot 10^{-4}$	-6.48	$2.55 \cdot 10^{-3}$	0.23
Monte Carlo (10^9 samples)	$1.61462 \cdot 10^{-4}$	-	$2.55595 \cdot 10^{-3}$	-
DA-GMM (51 GMEs)	$1.615292 \cdot 10^{-4}$	0.04	$2.444178 \cdot 10^{-3}$	4.37
DA-GMM(37 GMEs)	$1.609087 \cdot 10^{-4}$	-0.34	$2.443488 \cdot 10^{-3}$	4.40
Voxels $n = 50$	$1.64414 \cdot 10^{-4}$	1.83	$3.656683 \cdot 10^{-3}$	43.07
Voxels $n = 100$	$1.61719 \cdot 10^{-4}$	0.16	$3.079178 \cdot 10^{-3}$	
Adjoining cylinders $n = 50$	$1.61677 \cdot 10^{-4}$	0.13	0.0	100.00
Adjoining cylinders $n = 100$	$1.61677 \cdot 10^{-4}$	0.13	0.0	100.00
Parallelepipeds $n = 50$	$1.61761 \cdot 10^{-4}$	0.18	0.0	100.00
Parallelepipeds $n = 100$	$1.61701 \cdot 10^{-4}$	0.15	0.0	100.00
Linear Alfano $n = 50$	$1.58147 \cdot 10^{-4}$	- 2.05	$1.917488 \cdot 10^{-3}$	-24.98
Linear Alfano $n = 100$	$1.58147 \cdot 10^{-4}$	-2.05	$1.917487 \cdot 10^{-3}$	-24.98
Linear Patera $n = 50$	$1.58146 \cdot 10^{-4}$	-2.05	$1.917487 \cdot 10^{-3}$	-24.98
Linear Patera $n = 100$	$1.58146 \cdot 10^{-4}$	-2.05	$1.917487 \cdot 10^{-3}$	-24.98

Table 9.3 presents the resultant collision probability calculated with the method developed in this work (DA-GMM) and compared to alternative methods provided by Alfano (2009). Highlighted in green are the cases when the computed collision probability improves upon the alternative methods. For case 7, an error of 0.4% is achieved, which shows an improvement with respect to the linear and non-linear alternative methods. Moreover, the total time to compute the collision probability over the 2840 second interval with a time-step of 10 seconds is of three hours and less than one hour for the cases with 51 and 37 GMEs, respectively. This allows to compute the collision probability with sufficient time to conduct an avoidance manoeuvre and therefore it can be used during operations. For case 12, the results obtained by the alternative methods are greatly improved both using 37 and 51 Gaussian mixture elements. The poor performance of the alternative methods is due to the fact that there is no relative velocity, leading to a zero cumulative probability. A similar effect is produced in scenarios with low relative velocities, such in formation flying. For these scenarios, the non-linear methods fail and the linear methods provide a low accuracy due to the large encounter times. Moreover, these methods under-estimate the collision probability, which can lead to unexpected casualties. In comparison, the DA-GMM method estimates the collision probability with an error of 4.37%, providing the closest approximation to the real magnitude of the risk.

To summarise, the application of the DA-GMM method for these test cases has proven that:

- A collision probability error of 0.04% can be achieved with a lead time of 48 hours.
- The method is not limited to cases with large relative velocity
- A high accuracy estimate can be obtained with 37 Gaussian Mixture Elements.
- Very low collision probabilities ($P_c \approx 10^{-4}$) can be correctly predicted.
- The accuracy in P_c calculation is improved by > 70% with respect to the best alternative method.
- An accurate collision probability computation for the three-dimensional geometry and applicable to any encounter geometry can be obtained within five minutes for one GME and within one hour for the case with 51 GMEs, which allows to use the method for operational collision avoidance.

9.2. Crash scenario

With the sensitivity analysis performed in Chapter 8 and the collision probability calculations compared with Monte Carlo results, it is verified that the DA-GMM method correctly propagates the uncertainty and estimates the encounter risk. Before going into the simulation of real encounter scenarios, it is useful to gain

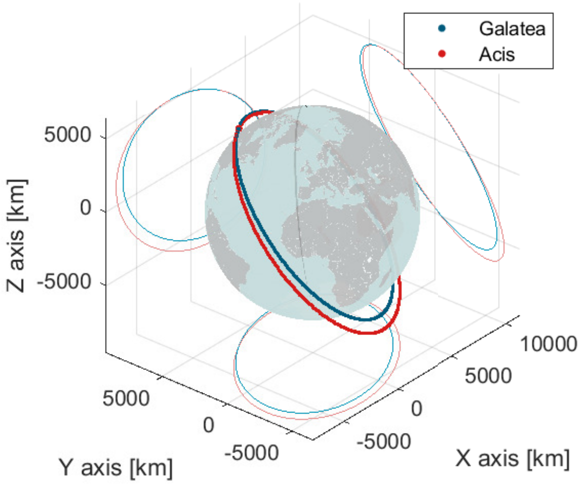


Figure 9.3: Illustration of nominal trajectories of Galatea and Acis leading to a collision.

more insight on what a given number of P_c represents and how it relates to the initial state uncertainty and lead time. To do so, the collision probability for a scenario where the nominal trajectory leads to a collision is simulated. In this section, the collision scenario is introduced and the simulation parameters are presented. The results from this simulation provide a guideline to interpret the risk estimation for real collisions.

9.2.1. Case summary

The main issue when computing the risk of collision is the uncertainty in the initial state that propagates to the encounter time leading to the phenomena of *uncertainty dilution* (Alfano, 2004). This occurs when the covariance of one or both objects is large, leading to a small collision probability due to the limited knowledge on the satellite position. The position accuracy provided by TLEs is in the order of 450 m, which already affects the estimated risk. Although most of the objects studied in this chapter are tracked through radar measurements by different space agencies and private companies, this information is not public. Therefore, the results of this analysis are limited by the accuracy of tracking data. To provide some baseline for comparison, a fictitious collision is simulated. Two satellites (named Galatea and Acis) in different, intersecting orbits are positioned initially in the same location. The position of both trajectories is propagated backwards with the perturbed dynamics model for 1, 3, 6, 12, 24, 48 and 72 hours. From these new initial states, the collision probability at TCA is calculated, assuming an initial state uncertainty. This analysis allows to inspect the effect of uncertainty dilution in a case where, with zero uncertainty, the collision probability would be $P_c = 1$. The simulation scenario is illustrated in Figure 9.3. Acis follows a retrograde orbit, which leads to an almost head-on collision. This is the worst case scenario for an encounter and it presents an interesting case for the collision probability. Despite the high relative velocity of the satellites, which would lead to short encounter, the uncertainty is distributed along the trajectory of the spacecraft, increasing the encounter period. This is complicated to model in short-term methods but it does not pose a challenge when using this method.

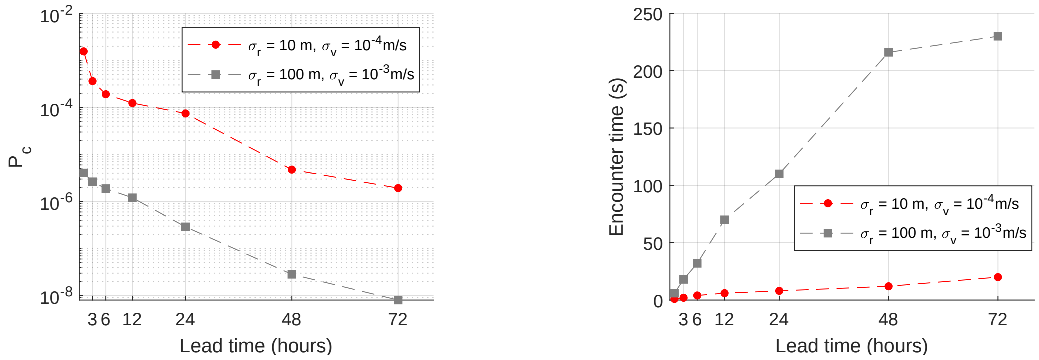
9.2.2. Simulation parameters

At the time of closest approach the nominal trajectories of both satellites intersect with an angle of three degrees, leading to an almost head-on collision. The encounter geometry is summarised in Table 9.4 and the initial state for each collision simulation is obtain by back propagating from this scenario.

For all the simulations it is selected to use a Taylor expansion order of $k = 3$ with 51 GMEs. According to the objective of this analysis, two different scenarios are considered for each simulation. First, a scenario with larger initial state uncertainty setting $\sigma_r = 100$ m and $\sigma_v = 10^{-3}$ m/s in each component. This scenario approaches the uncertainty provided by TLEs, which are the main source of tracking data used in this work. Second, a scenario with medium state uncertainty setting $\sigma_r = 10$ m and $\sigma_v = 10^{-4}$ m/s in each component. This approaches the accuracy achieved with radar measurements by tracking companies such as LeoLabs. For some satellites, more advanced tracking techniques are used which provide an orbit-determination ac-

Table 9.4: Kepler elements of the satellites at TCA.

Satellite	a [km]	e [-]	i [deg]	Ω [deg]	ω [deg]	θ [deg]
Galatea	7078.137	0.03	50	30	70	20
Acis	7441.707	0.0942	129.98	-152.05	128.44	-39.76



(a) Collision probability for change in lead time

(b) Encounter time-span for change in lead time

Figure 9.4: Evolution of collision risk for change in lead time

curacy under a meter. However, this is excluded from the analysis, since there are no results available for comparison that use such orbit determination techniques. Regarding the integration step for the collision probability determination, it is adjusted from 0.5 seconds to 10 seconds depending on the span of the encounter and the evolution of the collision probability rate.

9.2.3. Results and discussion

The evolution of the estimated risk as a function of lead time is illustrated in Figure 9.4. Focusing first on the predicted collision probability, it is observed that the estimated risk decreases as the lead time increases. This is expected since for the same initial state uncertainty, a longer lead time results in a higher final state uncertainty, therefore contributing more to the effect of uncertainty dilution. The decreasing trend approximates a linear relation in the logarithmic scale, which suggests that for this trajectory, the following relation can be established for the lead time measured in hours:

$$P_c(t_{lead}) = 10^{(C_1 - 0.0375\cdot t_{lead})} \tag{9.1}$$

$$P_c(t_{lead}) = 10^{(\log P_c(1) + 0.0375\cdot(1 - t_{lead}))} \tag{9.2}$$

Continuing with Figure 9.4a, it is surprising the effect of the difference in initial state on the collision probability. From the conclusion that both trends can be modelled as a line with the same slope in logarithmic scale, it follows that the difference in collision probability is approximately constant in this scale. In this scenario it is found that for the same lead time, the collision probability differs in two orders of magnitude for a decrease of one order of magnitude in the initial state uncertainty. This leads to a crucial conclusion: that the collision probability for a given encounter geometry using two sets of measurements made at the same time, but with different uncertainty, cannot be simply compared. The level of accuracy of the measurements must be taken into account when assessing the risk of collision. For this work, this implies that the results found for collision scenarios using two-line elements cannot be directly compared to the collision probability estimates performed by entities with accurate tracking data.

Focusing on Figure 9.4b, an increase in the encounter time is associated with an increase in lead time for the same initial state uncertainty. This is specially visible in this scenario since the encounter is head-on, and therefore the principal direction of the uncertainty ellipsoids are approximately aligned. In this case, the length of the encounter directly depends on the initial uncertainty, and therefore the curves do not have the same slope.

The main outcome of this study is a reference magnitude for the difference in collision probability associated to a change in initial state uncertainty, and how this value changes with lead time. It is concluded that, since there is a linear relation between lead time and collision probability in the logarithmic scale, this

Table 9.5: Time history of the predicted closest approach between Iridium-33 and Cosmos-2251 (Aida et al., 2009)

Days before TCA	7	6	5	4	3	2	1
Miss distance Iridium-33/Cosmos-2251 [m]	1752	1812	117	1243	688	984	584

Table 9.6: Collision probability for the Cosmos-2251/Iridium-33 encounter detected by NASA and the RSSS (Agapov et al., 2009).

Entity	**Time before TCA**	**Detected P_c**
NASA	1 day	10^{-10}
RSSS	10 minutes	$3 \cdot 10^{-5}$

difference is kept approximately constant and is equal to two orders of magnitude. Although the numbers found in this section are only valid for this particular case, this study provides a baseline to assess the validity of the collision risk computed in further sections.

9.3. Cosmos-2251/Iridium-33 collision

The Cosmos-2251/Iridium-33 is the largest collision event that has occurred, which resulted in more than 1,000 pieces of debris larger than ten centimetres in diameter (Kelso, 2009). The severity of the impact responds to the fact that both bodies were intact satellites heavier than 600 kg each, and the encounter occurred at an almost 90° angle and a relative velocity of nearly 11.65 km/s. Despite the fatal consequences of the encounter, the risk detected was not high enough to notify the satellite providers. By that time, the Iridium constellation was not included in the conjunction screenings of the U.S and Russian military. Moreover, the Iridium constellation was often found to have potential encounters with other objects, but due to the inaccuracy of the TLE data, most of these encounters were disregarded. After the collision, the security protocols were reinforced and the Iridium constellation was included in the high accuracy screening performed by the Joint Space Operations Centre (JSpOC). This collision event is an excellent source of data to predict a collision event and validate the collision probability calculations. In this section, the Cosmos-2251/Iridium-33 collision is simulated with the DA-GMM method using the publicly available information from the TLEs for different settings.

9.3.1. Case summary

The main issue that lead to the catastrophic event was the lack of warning. This was due to the fact that neither the U.S or Russian military screening programs included the Iridium-33 or Cosmos-2251. The protocol has improved and currently the U.S space Command (USspaceCOM) includes in the screening all objects larger than 10 cm in diameter. In the coming years, it is planned to include debris up to 5 cm in diameter (NASA, 2020).

However, the satellites were included in the screening process of the Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space (SOCRATES) software. SOCRATES is a tool provided by CelesTrack which uses STK to propagate the trajectory and provide the miss distance at TCA and the maximum collision probability. This software uses the public data from the TLE and therefore the results are treated as an initial estimation which should be reviewed by the satellite operator with more accurate tracking data. SOCRATES tracked the satellites and predicted the minimum distance from seven days prior to TCA, as presented in Table 9.5. Although a close approach of 117 m was predicted five days prior to TCA, this value increased to more than 1 km on the following data update, and therefore did not raise further concern. Moreover, it should be noted that during this period, the Iridium-33/Cosmos-2251 encounter was not the closest approach detected for the entire Iridium constellation, nor for the Iridium-33 satellite itself. As a result, the Iridium-33/Cosmos-2251 encounter did not rank high on the priority list and the potential collision was not notified to satellite operators.

After the collision, the scenario was run by NASA and the Russian Space Surveillance System (RSSS) with accurate radar surveillance data ten minutes and one day prior to TCA, respectively. The collision probability predicted by these entities is summarised in Table 9.6.

The collision took place on 10 February of the year 2009 at 16:55:59.806 UTC at an altitude of 788.6 km. The event occurred at 97.9° longitude, 72.5° latitude above the Taymyr peninsula in Siberia. This constituted the first accidental collision between catalogued satellites.

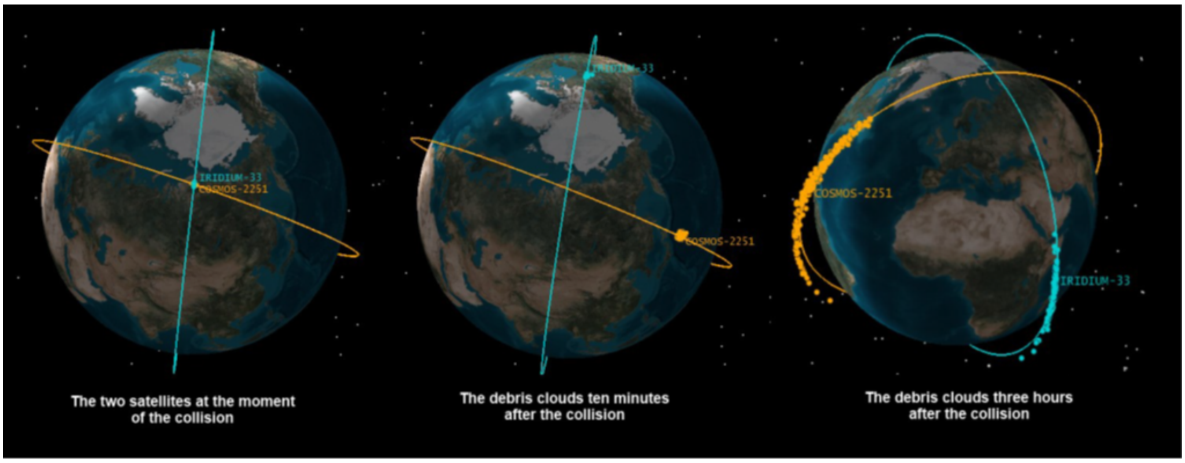

IRIDIUM-33
COSMOS-2251
The two satellites at the moment of the collision
IRIDIUM-33
COSMOS-2251
The debris clouds ten minutes after the collision
COSMOS-2251
IRIDIUM-33
The debris clouds three hours after the collision

Figure 9.5: Evolution of the Iridium-33/Cosmos-2251 debris cloud after collision (Kelso, 2009).

Table 9.7: Orbital characteristics of the satellites 10 minutes before the collision (Agapov et al., 2009)

Satellite	h_a (km)	h_p (km)	i (°)	e (-)	Ω(°)	Velocity (km/s)
Iridium-33	796	756	86.4	0.0028	121.3	7.46
Cosmos-2251	794	767	74.0	0.0019	17.3	7.45

Directly after the encounter, both satellites produced over 1,400 pieces of debris larger than 10 cm and nearly $3 \cdot 10^6$ pieces larger than 1 mm which scattered in a wide range of orbital altitudes. Figure 9.5 illustrates the evolution of the fragments after the collision. In only three hours the debris, which have different sizes and were oriented in a different direction after the collision, are already scattered close to the original orbit. With time, the scattering process continues, eventually covering the full orbital sphere at the collision altitude. The pieces of debris will collide with each other, reducing the number of larger pieces. Due to the scattering of the debris across altitude and the reduction of size due to collisions, the spatial density of debris larger than 1 cm in the collision altitude has decreased by a factor of three after ten years.

9.3.2. Simulation parameters

The satellites forming the Iridium constellation are located at a highly crowded orbital band, which caused multiple collision warning events. Table 9.7 provides the orbital characteristics of both satellites at the last radar measurement, at 16:46h on 10 February 2009, ten minutes before collision. Both orbits share a range of orbital altitudes and have a similar inclination.

The position of each satellite prior to collision is available as TLE on SpaceTrack, which are updated two times a day. The tracking accuracy of the radar measurements available to the satellite operators and space agencies was 2.6 m and 1.7 m for Iridium-33 and Cosmos-2251, respectively. However, this information is not publicly available. To recreate the collision event, the data from TLE will be used, taking into account that the resultant collision probability could be improved with higher-accuracy data. The initial state of the satellites retrieved from the last TLE published before the collision is presented in Table 9.8. Since the data comes from a TLE, the correlation between the uncertainty in each component follows the trend studied by Geul et al. (2017). The values selected are summarised in Table 4.1. Regarding the magnitude of the standard deviation for the initial state uncertainty, the values are selected following the results from Flohrer et al. (2008). The settings used for the simulation of the Cosmos-2251/Iridium-33 encounter summarised in Table 9.9.

Table 9.8: Cartesian state of the satellites in the Earth Centred Inertial reference frame at 00:00 UTC on 10 February 2009.

Satellite	x [km]	y [km]	z [km]	$\dot{x}$ [km/s]	$\dot{y}$ [km/s]	$\dot{z}$ [km/s]
Iridium-33	-3240.99	4759.60	4246.546	2.01107	-3.9914	5.98213
Cosmos-2251	2689.06	2726.88	6049.935	-6.53545	-1.16479	3.40109

Table 9.9: Method parameters for Cosmos-2251/Iridium-33 encounter. Uncertainty given in the *RTN*-frame.

Parameter	**Cosmos-2251**	**Iridium-33**
Nº GME	1. 37, 51	1, 37, 51
Taylor expansion order	3	3
Reference Area [m^2]	4.0	2.2
Mass [kg]	900.0	689.0
Radius [m]	2.5	2.5
C_D [-]	2.2	2.2
$[\sigma_R, \sigma_T, \sigma_N]$ [m]	[100,400,100]	[100,400,100]
$[\sigma_{V_R}, \sigma_{V_T}, \sigma_{V_N}]$ [m/s]	[1 0.25 0.25]	[1 0.25 0.25]

9.3.3. Collision probability estimation

Once the model parameters have been defined, several simulations can be executed to estimate the risk of collision with the available data. First, the effect of modifying the uncertainty model is studied, by simulating the encounter considering only the central body acceleration, and then including the full dynamical model developed. Second, the effect of using a larger number of Gaussian Mixture Elements is analysed. Finally, to assess the effect of lead time, the initial state from the TLE is propagated forward in time and the collision probability is calculated from that fictional scenario. Due to the large uncertainty in the TLE, the result of this analysis does not necessarily represent the real encounter. It would be more revealing to perform the same analysis with the real tracking data with lower uncertainty. However, with the available resources, this analysis still allows to study the evolution of collision risk as the encounter time approaches.

Effect of dynamics model

The trajectories of Cosmos-2251 and Iridium-33 are propagated from the last TLE, 16 hours and 56 minutes before the collision event. To study the impact of the dynamical model on the predicted risk, the simulations are run for a simplified model considering only the central gravitational acceleration from the Earth, and with the full model described in Subsection 3.5.5.

For the simplified model, the maximum collision probability is detected 75 seconds before the actual encounter time. The collision probability in this case is $P_c = \sim 10^{-51}$, which can be approximated as zero. Therefore, the simplified model fails to predict the risk of an encounter, and situates the encounter time more than one minute away from the real time of closest approach. This confirms that excluding the perturbing accelerations is unacceptable.

When adding the perturbation of the spherical harmonics up to degree and order six, the central gravitational acceleration from the Sun and the Moon and the atmospheric acceleration, the detected collision probability with a Gaussian uncertainty propagation becomes $P_c = 9.3511 \cdot 10^{-7}$. This result proves that the dynamics model has a large influence on the detected collision probability. Moreover, in this scenario the maximum probability is registered at 16:55:58 UTC, 0.8 seconds before the actual collision.

Effect of number of GMEs

Once it is established that the full dynamics model is required, the collision risk estimation is run with increasing number of Gaussian mixture elements to improve the prediction. Based on the results from the sensitivity analysis, it is decided to perform extra simulations with 37 and 51 elements respectively. The results are displayed in Table 9.10. First, it is noted that the predicted collision probability increases with increasing number of elements, but by small percentage. The discrepancy between using one and 51 elements is < 1%, which does not affect the order of magnitude of the predicted error. When increasing to 37 elements, the change with respect to 51 decreases to < 0.1%. Therefore, it is concluded that 37 elements are sufficient to correctly estimate the risk of collision.

In Figure 9.6, the time distribution of the probability rate for different number of elements is illustrated. First, it is noted that the distribution for one GME is displaced to the right with respect to the other two scenarios. By increasing to 37 elements, the difference between the maximum probability rate and the actual collision time reduces to 0.2 seconds. This result supports the importance of modelling for non-Gaussian uncertainties, since in a time-span of 1 second, the relative position of the satellites can change by more than

Table 9.10: Collision probability predicted 16h 55 min 58.806 s before the collision.

Nº GMEs	P_C	$\Delta P_c(\%)$
1	$9.3511 \cdot 10^{-7}$	0.9302
37	$9.4329 \cdot 10^{-7}$	0.0639
51	$9.4389 \cdot 10^{-7}$	-

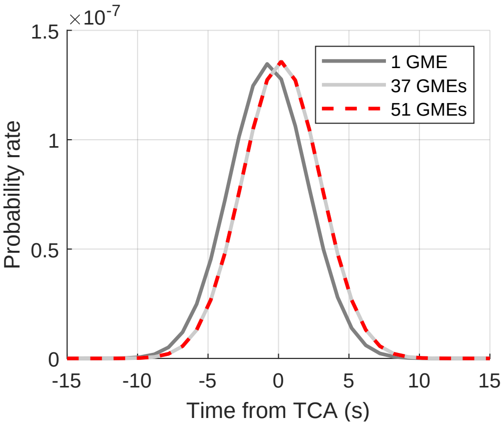


Figure 9.6: Evolution of collision probability rate for Cosmos-Iridium collision with 1, 37 and 51 Gaussian mixture elements.

10 km, which completely modifies the encounter geometry. It is also observed that the curves for 37 and 51 elements match, which supports the decision of limiting the model to 37 elements.

Effect of lead time

With the available data from the TLEs, the last information available is from 16 hours and 56 minutes before the collision. However, it is interesting to analyse how this prediction changes with a decrease in lead time. To do so, it is assumed that both spacecraft follow the nominal trajectory designated from the last TLE. This is not the ideal scenario, considering the low accuracy associated to these measurements. With new starting conditions defined 1, 3, 5 and 7 orbital revolutions before the collision, the simulation is repeated. Figure 9.7 shows the initial position of both satellites at these initial conditions. As it is observed, although in an unperturbed orbit all these points would match, in this case the initial conditions are displaced by hundreds of kilometres. This exemplifies the level of complexity of orbital dynamics when including the effect of perturbations.

The results from these simulations are illustrated in Figure 9.8. The evolution of the probability rate with time is depicted in Figure 9.8a. As expected, the encounter time-span increases with increasing lead time. Moreover, it is observed a change in the time of maximum collision probability rate. However, there is no clear relation between the total collision probability and the lead time. As observed in Figure 9.8b, the maximum collision probability is observed when starting the propagation three revolutions before time of closest approach. Starting the propagation from the same nominal orbit but one revolution before time of closest approach yields a collision probability one order of magnitude smaller. This confirms that extrapolating the initial state from a TLE is not a valid approach to create input data. The estimation of collision risk is an extremely sensitive process that requires adequate tracking data. This data is available to space agencies and satellite operators, but open to the general public, which limits the possibilities to study in depth the Cosmos-2251/Iridium-33 collision.

9.3.4. Results summary

The collision event between Cosmos-2251 and Iridium-33 changed the course of satellite operations and forced a tightening of the space debris mitigation guidelines. This event is an excellent source of verification, and allows to extract important conclusions regarding the accuracy of the method and its applicability to space operations. By analysing the effect of using a simplified model versus a full dynamics model it is confirmed that the two-body problem is not valid for collision probability estimation. A change of meters in the nominal position completely shifts the closest approach time and changes the encounter geometry. In this case, the collision probability detected for the simplified dynamics model is negligible. Regarding the

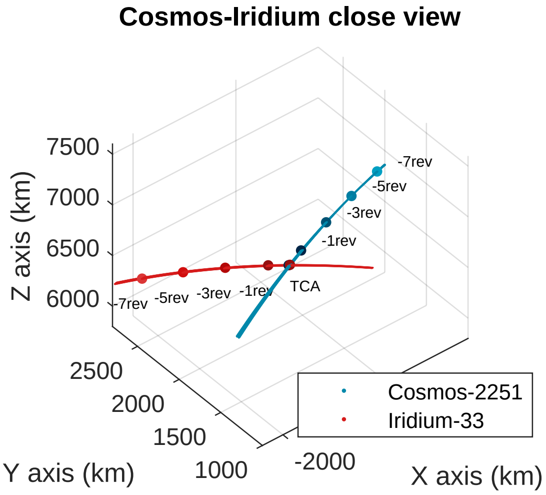
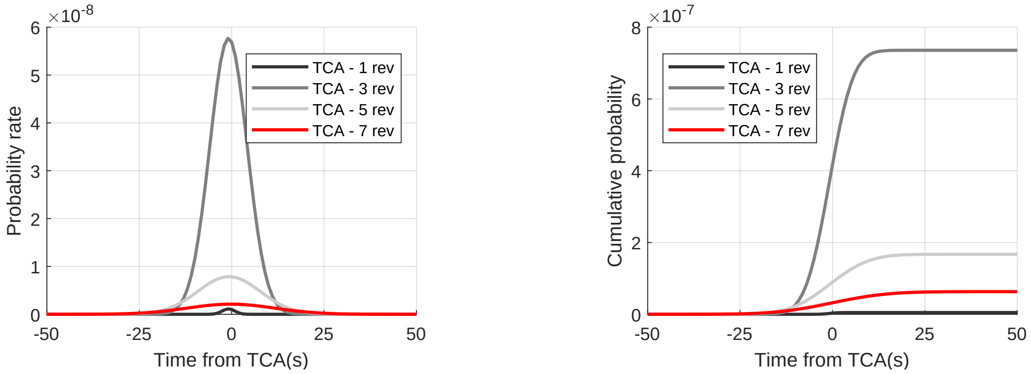


Figure 9.7: Position of the satellites 1,3,5 and 7 orbits before the collision event



(a) Collision probability rate over time

(b) Cumulative collision probability over time

Figure 9.8: Evolution of collision risk for Cosmos-2251/Iridium-33 collision with changing time from TCA

application of a Gaussian Mixture Model to represent the satellite uncertainty, it is concluded that adding 37 elements improves the accuracy, but for propagation times of 16 hours a Gaussian propagation would be sufficient. The addition of extra elements is useful for larger propagation times, as studied in Chapter 8. This is a fortunate conclusion, since longer lead times also allow for a larger computational load. Therefore, the method can be used with a higher accuracy and still obtain the result in time to decide on the application of a collision avoidance manoeuvre.

The collision probability estimated 16 hours before the time of closest approach is $P_c = 9.4329 \cdot 10^{-7}$. By itself, this value does not seem like a big threat to raise concern, However, it must be considered that this analysis has been done with TLE data, which have a low accuracy. From the analysis performed in Section 9.2 it is observed that for a scenario where the nominal trajectory leads to a crash, the TLE accuracy predicts a collision probability of $P_c = 1.2 \cdot 10^{-6}$, 12 hours before TCA. The conclusion from this analysis was that increasing the accuracy by one order of magnitude leads to a reduction in the estimated risk of 2-3 orders of magnitude. This indicates that the real collision probability estimated with higher accuracy measurements could be at least in the order of $P_c \approx 10^{-5}$. This estimate agrees with the estimates computed by Agapov et al. (2009) with accurate tracking data. Therefore, it can be concluded that the DA-GMM method is capable of accurately estimating the time of maximum risk of collision. The flexibility of the method allows to increase the accuracy to obtain a risk estimate for long lead times. Moreover, although it cannot be validated with real data, the verification of the method with Monte Carlo simulations and the simulations for the nominal crash scenario suggest that the DA-GMM method would have predicted a significant risk of collision for the Cosmos-2251/Iridium-33 encounter. With accurate tracking data, the method could have been subsequently used with new measurements to refine the risk prediction and decide on a collision avoidance manoeuvre with sufficient time.

Table 9.11: Orbital characteristics of the satellites at 22:56 UTC, 2 hours before TCA.

Satellite	h_a **(km)**	h_p **(km)**	i (°)	e (-)	Ω(°)	**Velocity (km/s)**
Chang Zheng-4C	1203.4	969.7	100.36	0.0156	201.1	7.30
Cosmos-2004	1012.8	969.9	82.95	0.0029	8.12	7.35

9.4. Chang Zheng-4C/Cosmos-2004 close encounter

The encounter between the upper stage of the Chang Zheng-4C rocket (CZ-4C herein after) and the Cosmos-2004 satellite offers an interesting case for study. In October 2020, the space monitoring company LeoLabs warned the space companies and satellite operators of a potential collision between CZ-4C and Cosmos-2004 (Corbett, 2020). Both objects were inactive and uncontrollable, which entailed that no mitigation action was applicable. The combined mass of both objects was 2,800 kg, twice that of the Iridium-33/Cosmos-2251 encounter. Moreover, the relative velocity at TCA was estimated to be 14.65 km/s as the collision geometry was head-on. These conjunction characteristics would result in fatal consequences in case of a collision. Fortunately, despite the high risk detected the collision did not occur. This encounter provides excellent data to replicate the event and estimate the collision probability. Although nowadays active debris removal is still under development and no measure could have been taken to prevent this collision, increasing the lead time will enable timely mitigation actions in the future.

9.4.1. Case summary

The CZ-4C and Cosmos-2004 objects have a slightly eccentric orbit with a perigee altitude of 969 km. This altitude band presents a lower density of satellites and space debris and therefore a lower risk of collision. However, an impact of such magnitude would create a large cloud of debris, altering this scenario. The orbital characteristics of both objects on 15 October at 22:56 UTC (two hours before TCA) are provided in Table 9.11. The data is obtained from the TLEs publicly available at SpaceTrack.

It should be noted that the CZ-4C upper stage orbit had a perigee-apogee altitude difference larger than 200 km, and the orbital eccentricity has continued to increase. The object is uncontrollable and at such a high altitude aerodynamic re-entry cannot be expected in the next decades. Therefore, this object does not comply with the IADC mitigation guidelines which state that satellites and bodies launched into LEO must be cleared from the protected region within 25 years of end of mission (IADC, 2020).

The scenario evaluated by LeoLabs at 17:24:52 UTC predicted a close approach of 25 m with a collision probability of $P_c = 3.8 \cdot 10^{-2}$. Leolabs has independent satellite tracking data measured from a network of four radar sites, which offers an average orbit determination accuracy of 53 m (*LeoLabs* 2021). This tracking data is private and therefore the same scenario cannot be replicated. The collision probability is calculated with the accuracy provided by the TLE and the behaviour observed in Section 9.2 is used to relate this results with the predictions from LeoLabs.

9.4.2. Simulation parameters

For this simulation, the TLE measured on 15 October at 20:03:30 UTC and 20:06:04 UTC for Cosmos-2004 and CZ-4C respectively will be used. The reason for discarding the last TLE measured before the collision is that with a lead time of less than two hours it cannot be assured that the collision risk will be calculated before the last station passage before the closest approach. Therefore, this does not allow to make a timely decision and the results would not be useful in an operational context. The initial state of the satellites used for this simulation, obtained from the TLE and converted to include the effect of orbital perturbations as modelled by the SGP4 model is presented in Table 9.12.

For this scenario, only one simulation will be performed, as the effect of changing the method parameters has already been discussed. To this end, the settings are selected according to the results of the sensitivity analysis and the collision scenarios simulated so far. Since only one simulation is being performed, computational time is not a leading factor and therefore it is chosen to use 51 GMEs. This leads to a simulation time of less than one hour and produces highly accurate results. Although at this high altitude the effect of the atmospheric acceleration is reduced, this perturbation is introduced in the model. The dimensions and mass of both satellites are retrieved from available catalogue data (NASA, 2021a) and completed with the ballistic coefficient information available from the TLEs. The combined object radius is nominally set to 16m, to account for the boom extending from Cosmos-2004. However, two other scenarios are studied considering that

Table 9.12: Cartesian state of the satellites in the Earth Centred Inertial reference frame on 15 October 2020 at 20:03:30 UTC and 20:06:06 respectively.

Satellite	x [km]	y [km]	z [km]	$\dot{x}$ [km/s]	$\dot{y}$ [km/s]	$\dot{z}$ [km/s]
Cosmos-2004	7312.25	1018.98	-14.55	-0.09857	0.89297	7.28887
CZ-4C	-7042.13	-2664.88	13.59	-0.53034	1.18656	7.12962

Table 9.13: Method parameters for Cosmos-2251/Iridium-33 encounter. Uncertainty given in the *RTN*-frame.

Parameter	Cosmos-2004	CZ-4C
Nº GME	51	51
Taylor expansion order	3	3
Reference Area [m^2]	13.8	2.2
Mass [kg]	825.0	1575.0
Radius [m]	2.5-10	10-20
C_D [-]	2.2	2.2
$[\sigma_R, \sigma_T, \sigma_N]$ [m]	[100,400,100]	[100,400,100]
$[\sigma_{V_R}, \sigma_{V_T}, \sigma_{V_N}]$ [m/s]	[1 0.25 0.25]	[1 0.25 0.25]

the elongated shape of the rocket body requires to account for different encounter geometries. The setting used for the simulation of the Cosmos-2004/CZ-4C encounter are summarised in Table 9.13.

9.4.3. Results and discussion

The collision probability calculated five hours before the encounter is presented in Table 9.14 for four different encounter radius. The encounter radius has a large effect on the computed collision probability. In this case, due to the cylindrical shape of the rocket and the boom extending out from the body of Cosmos-2004, there are multiple encounter geometries that should be considered. The best case scenario assumes a combined encounter radius of eight meters, which corresponds to adding the minimum dimensions of both bodies. On the other hand, a combined radius of 30 meters is set for the worst-case scenario considering the rocket height and the boom. Two intermediate values are selected to represent the nominal scenario.

Analysing first the magnitude of the collision probability, it is concluded that the risk is significant. Even with the accuracy provided by the TLEs, the DA-GMM method detects a collision with a maximum probability less than one second away from the time of closest approach. The results for different encounter geometries confirm that that the encounter radius has a large impact on the perceived collision risk. Failing to properly identify this radius can lead to fatal consequences, and overestimating the radius can largely reduce the confidence on the assessed risk. LeoLabs estimated a collision probability of $3.8 \cdot 10^{-2}$ for this encounter, which raised an alarm. The combined radius used by LeoLabs to compute this risk has not been published and the input data for the satellite states is taken from radar measurements with an average of 53 m error in accuracy. Therefore, the results from this study cannot be directly combined with the risk estimates from LeoLabs. However, the encounter geometry is heads-on, very similar to the event studied in Section 9.2. Using the crash scenario as a baseline, it can be assumed that a reduction in the initial state uncertainty of one order of magnitude will lead to a reduction in collision probability of two orders of magnitude. Following this principle, the collision probability for this event with the initial state uncertainty from the measurements by LeoLabs would be in the range $P_c = [2 \cdot 10^{-3} \;\; 2 \cdot 10^{-2}]$ depending on the encounter radius. The risk estimated

Table 9.14: Collision probability for Cosmos-2004/CZ-4C encounter evaluated 5 hours before TCA.

Combined radius [m]	P_c
8	$1.9824 \cdot 10^{-5}$
12	$3.9101 \cdot 10^{-5}$
16	$7.9293 \cdot 10^{-5}$
30	$2.7873 \cdot 10^{-4}$

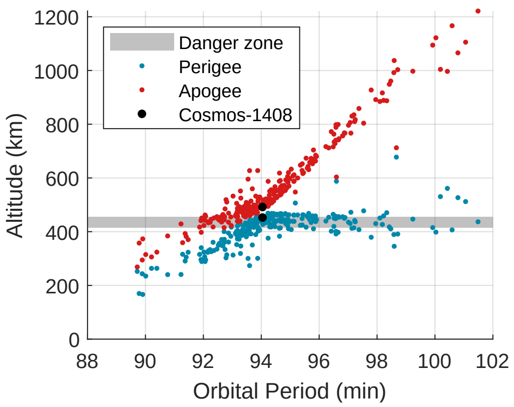


Figure 9.9: Distribution of space debris from Cosmos-1408 on November 30, 2021.

by LeoLabs is slightly larger than the estimated range. Considering the outcome of the encounter, this could imply that the results from LeoLabs slightly overestimated the encounter risk.

9.5. ISS screening

The final scenario to study with the DA-GMM method is between the International Space Agency (ISS) and a screening of selected space debris. On November 15, 2021 the inoperative intelligence satellite Cosmos-1408 was destructed in a Russian anti-satellite test (NASA, 2021a). This event produced over 1,500 pieces of debris which after two days span up to altitudes of 800 km. The crew sheltered at at the Dragon and Soyuz MS-19 capsules due to the unknown risk that the debris posed. Fifteen days later, an extra-vehicular activity was postponed due to "the lack of opportunity to properly assess the risk it could pose to the astronauts”(NASA, 2021b). This scenario offers a perfect opportunity to test the DA-GMM method to compute the collision probability with the space debris resulting from this event.

First, all the available space debris tracking data resulting from this event as published by December 13, 2021, will be screened to find potential objects that could intersect the ISS orbit. To do so, all the objects that have a perigee and apogee altitude both lower or higher than $h_{ISS} \pm 20$ km are discarded. The remaining objects are propagated using the perturbed dynamical model and their closest approach with the ISS are calculated. All objects that violate the "pizza-box" safety distance of 25 km from the ISS are selected for further study. This process is summarised in Subsection 9.5.1. For these scenarios, the collision probability with the ISS is calculated considering a the body as a single sphere. For the scenario with the largest collision probability, a refinement of the risk calculation is performed using the multi-sphere model of the ISS with the aim to obtain a more accurate estimate.

9.5.1. Case summary

On December 2, 2021, the tracking data of 300 pieces of debris resulting from the anti-missile satellite test destroying Cosmos-1408 were published as TLEs by Space-Track[1]. From these objects, 179 were located at orbits passing comprising altitudes within 20 km of the ISS orbit.These objects are selected for further study. Figure 9.9 presents the Gabbard diagram of the destruction event. The collision occurred at an altitude 50 km higher than the ISS orbit. As observed, this causes the debris cloud to scatter along altitude bands that comprise the station, leading to potential threats. It is observed that most pieces of debris have perigee altitudes within the *danger zone*. From the all the debris studied, only 15 pieces have perigee and apogee altitude below the danger zone. This entails that most of the debris is at altitudes that will continue to pose a threat. Moreover, as the debris in lower altitudes start to decay, the pieces that have been ejected into higher altitudes will begin to pose a threat to the ISS. Therefore, this collision event will continue to pose a hazard to the ISS and it is crucial to properly assess the risk of collision accurately and timely. For the 179 pieces of debris that have been found to pose a potential threat to the ISS, the following process is followed. First, the nominal trajectory of all the objects is propagated using the full dynamical model from the successive TLEs. With the

[1] `https://www.space-track.org`. NORAD IDs 49516 - 49721, 49781 - 49807 and 49820 - 49999. Some of the objects in this range had already decayed by the time of TLE publication.

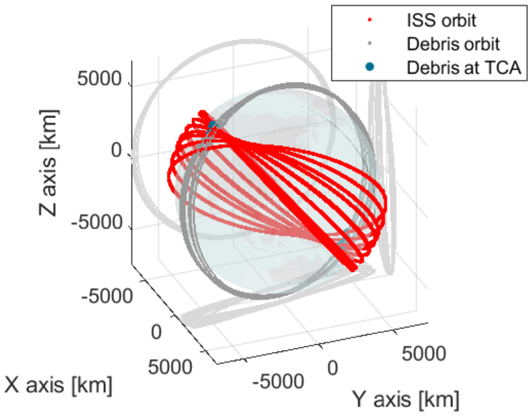


Figure 9.10: Orbit of the ISS (red) and Cosmos-1408 debris (grey) when close approach of <25 km are detected. Position of debris at TCA presented in blue.

Table 9.15: NORAD ID and characteristics of close encounters with the ISS. Shaded cells represent encounters that pose a significant risk to the ISS.

NORAD ID	**Δr [km]**	**Time of closest approach**		**P_c**
49597	24.84	December 3, 2021	01:21:44 UTC	$\sim 0\ (10^{-318})$
49633	13.48	December 2, 2021	21:30:13 UTC	$\sim 0\ (10^{-74})$
49633	6.11	December 11, 2021	17:44:20 UTC	$1.1115 \cdot 10^{-4}$
49654	23.62	December 6, 2021	02:57:08 UTC	$\sim 0\ (10^{-236})$
49669	15.61	December 5, 2021	01:24:16 UTC	$\sim 0\ (10^{-40})$
49683	20.53	November 30, 2021	20:40:47 UTC	$\sim 0\ (10^{-106})$
49835	9.05	December 8, 2021	02:12:17 UTC	$\sim 0\ (10^{-35})$
49865	9.23	December 7, 2021	23:06:17 UTC	$\sim 0\ (10^{-21})$
49843	13.35	December 9, 2021	20:48:05 UTC	$\sim 0\ (10^{-95})$
49859	7.47	December 12, 2021	12:19:08 UTC	$\sim 0\ (10^{-55})$

resulting trajectories, the closest approaches between each debris and the ISS are calculated. All the objects that enter within a 25 km distance of the ISS are selected for further study. These objects are then propagated to TCA using the DA-GMM method and the collision probability is calculated assuming a that the encounter geometry is a sphere surrounding the ISS.

For the settings of the DA-GMM software, it is chosen to use a single Gaussian Mixture Element (i.e., a Gaussian uncertainty propagation). As concluded from previous analysis, for short integration times the collision probability can be estimated with an accuracy within 1% with a single GME, which allows to properly estimate the order of magnitude of the collision risk. Since this analysis requires to analyse multiple scenarios, it is decided to perform a Gaussian propagation with Taylor expansion order three. The initial state uncertainty is set accordingly to the accuracy of the TLEs as $\sigma_{R,N} = 100$ m, $\sigma_T = 400$ m, $\sigma_{V_T,V_N} = 0.25$ m/s, and $\sigma_{V_R} = 1$ m/s. The remaining input parameters are obtained from the TLE.

9.5.2. Collision probability: single sphere

The screening of objects with the data published by December 13, 2021, led to the detection of ten potential close approaches with miss distances lower than 25 km. The geometry of the encounter is illustrated in Figure 9.10. Four close approaches were located over the northern hemisphere crossing of the orbital planes and six over the southern hemisphere. The characteristics for these approaches are summarised in Table 9.15, ordered by catalogue number. From the identified approaches, only four are below the 10 km miss distance threshold. Although these present the highest potential, the encounter geometry plays an important role in the resultant collision probability, and all the scenarios are analysed.

Table 9.15 also includes the collision probability calculated for these close encounters. For $P_c < 10^{-10}$ only the order of magnitude is presented. Inspecting the results, it is first observed that although there is

Table 9.16: Collision probability for the close approach of SAT #49633 on December 11, 2021, at 17:44:20 UTC.

Geometry	**Encounter period (s)**	$\mathbf{P_c}$
Single-sphere	1.9	$1.1115 \cdot 10^{-4}$
Multi-sphere, full body	1.3	$1.4902 \cdot 10^{-5}$
Multi-sphere, no solar panels	1.2	$1.8184 \cdot 10^{-6}$

a correlation between the miss distance and the collision probability, the collision risk cannot be evaluated from the miss distance. For example, object #49859 has a miss distance of 7.47 km and a collision probability much lower than objects #49835 and #49865 with greater miss distances. Second, it can be concluded that in nine out of the ten approaches the collision risk is negligible and there is no need for warning. However, object #49633, which threatens the ISS twice in the span of ten days, poses a very significant risk. A collision probability of $1.11 \cdot 10^{-4}$ is calculated which, according to the ISS operation guidelines requires to perform a collision avoidance manoeuvre (CAM) unless the burn itself will place the crew at greater risk. However, no CAM or collision risk was notified on this date. A possible explanation is that using more accurate tracking data leads to a smaller collision probability that assures a safe encounter.

This single collision event has caused one encounter with a significant risk of collision in the span of two weeks. In light of the rapidly rising amount of objects that populate LEo, this scenario is expected to repeat with increasing frequency. The debris of Cosmos-1408 will continue to be a threat to the ISS in the following years. Moreover, the deployment of satellite constellations, such as StarLink, increases the risk of a cascading effect. The result from this screening confirms the importance of accurate and fast collision detection techniques, that allow to assess the risk of multiple debris simultaneously.

9.5.3. Collision probability: multi-sphere

The results of the screening of potential encounters between the ISS and the debris of Cosmos-1408, object #49633 is identified as a potential threat. With a collision probability of $P_c = 1.11 \cdot 10^{-4}$, a collision avoidance manoeuvre should be performed. However, this collision probability is obtained considering an over-dimensioned ISS, leading to an over-estimate of the risk. Nominally, this would be considered as an additional safety factor that improves the safety of life and operations at the ISS. However, as the recent events have proven, this is not the case. Overestimating the risk can be harmful in some scenarios, where the real collision probability is required. Imagine a scenario in which multiple debris pose a simultaneous threat to the ISS, but the collision probability has been calculated with the spherical body assumption. In this case, the real risk from each debris is unknown, which complicates the task of deciding on a collision avoidance manoeuvre. In other scenarios, the collision avoidance manoeuvre might pose a higher risk than the collision itself. This is the case of manoeuvring during a docking operation. In these cases, it is crucial to know the real probability of collision to compare it to the risk of alternatives. With this objective, the multi-sphere collision probability model was designed to eliminate the assumption of a spherical shape for the ISS. To test this model, the threatening scenario presented by object #49633 is assessed and its collision probability is compared to the result obtained with the single-sphere model.

For this simulation, the model settings from the single-sphere simulation are kept except for the hard-body shape. For this, the ISS is modelled as a set of spheres, from which quadrature points are extracted and the boundaries are obtained as explained in Section 6.3. The body of the vehicle is formed by 20,000 spheres and the solar panels by 5,000 spheres each. To achieve a sufficient accuracy, 590 quadrature points are selected on each sphere. Since only one configuration for the solar panels is studied, the collision probability considering only the body of the ISS is also calculated. This provides a lower bound for the collision risk in the case where the arrays are aligned with the principal axis of the position uncertainty distribution.

Table 9.16 summarises the collision probability for this event detected assuming that the ISS is bounded by a sphere and considering the full body and the satellite without solar arrays respectively. Considering the real shape of the ISS results in a decrease of the collision risk of one order of magnitude. This risk reduces another order of magnitude by removing the effect of the solar panels. Therefore, it can be concluded that the collision probability of this encounter with the ISS lies in the range $[10^{-5} \ \ 10^{-6}]$. With a collision probability $< 10^{-5}$ it is not required to perform any collision avoidance manoeuvre. Therefore, although the initial results indicated that the debris posed a significant risk that required action, a more detailed analysis of the event proves that no action is required.

Figure 9.11 presents the evolution of the rate of change of collision probability with time. It is observed

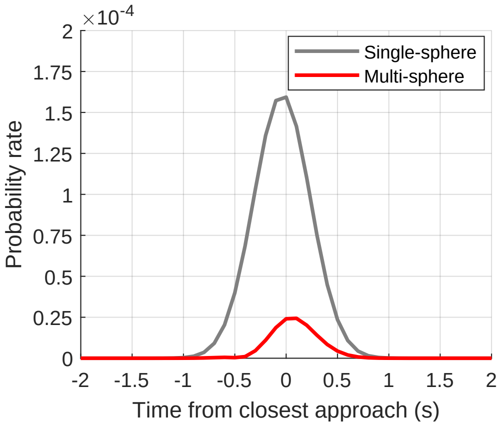


Figure 9.11: Collision probability rate for the close approach of SAT #49633 on December 11, 2021 at 17:44:20 UTC with the single-sphere and multi-sphere model.

that the multi-sphere model predicts a shorter collision which is displaced forward in time. This suggests that one second before time of closest approach, the risk was detected in the edge of the sphere, away from the ISS. For this reason, when considering a more complete model the collision probability is almost zero at this time. Since a difference of one second represents a change of approximately 7 km in the position of the ISS, the multi-sphere model allows to predict the encounter geometry and plan a mitigation strategy.

With these results, it is proven that assuming a spherical volume as the body of the ISS has a large impact on the estimated collision probability. This model allows a higher accuracy without a significant increase in in computational time and can be used for operational collision avoidance. A second advantage of the multi-sphere model is that the collision probability rate is calculated for every node on the surface of the satellite and then it is added up. Therefore, it is possible to obtain the results of the collision probability rate detected at each node. This allows to evaluate the relative risk within the spacecraft, identifying the regions with a higher change of impact and therefore the best option for a collision avoidance manoeuvre. An example of this effect is illustrated in Figure 9.12. Although the numbers found are difficult to interpret, since they represent the collision probability rate on every node, which needs to be added up and integrated over time to compute the total collision probability, the relative orders of magnitude are interesting to inspect. In this case, the maximum probability rate observed is in the order of 10^{-12}, represented by yellow colours, and the lower threshold for collision probability rate is set to five orders of magnitude lower. By analysing the views of the ISS with this colour code, it is observed that the back view presents a larger collision probability, in particular the central area. By observing the side view, the difference in collision probability between the front and the back can be clearly observed. The largest rate on the central nodes is due to the largest size of these spheres. Therefore, some sort of normalisation with the sphere radius would be required to visualise the collision risk without this dependence on size. In any case, this method allows to compute the areas of larger risk taking into account the full non-Gaussian state uncertainty of the target and chaser, and not only the nominal trajectory of the objects.

9.6. Results summary

In this chapter, five types of scenarios have been simulated to validate the collision probability calculation and study the risk of current events. This has allowed to draw several conclusion regarding the behaviour of the method for different initial state uncertainties and lead time. A summary of the findings obtained from the simulations performed for real and test cases with the DA-GMM method of collision probability follows:

- For short term encounters and 48 hours of lead time, the method achieves an error of 0.04%, providing better accuracy than all the methods tested by Alfano (2009), which correspond to the methods included in the collision probability calculation software by NASA and ESA.

- For methods with very low relative velocity and long-term encounters, the method achieves an accuracy of 4.4% compared to the ≈ 25% error obtained as the best estimate from conventional methods.

- The method does not have any limitations in terms of encounter geometry.

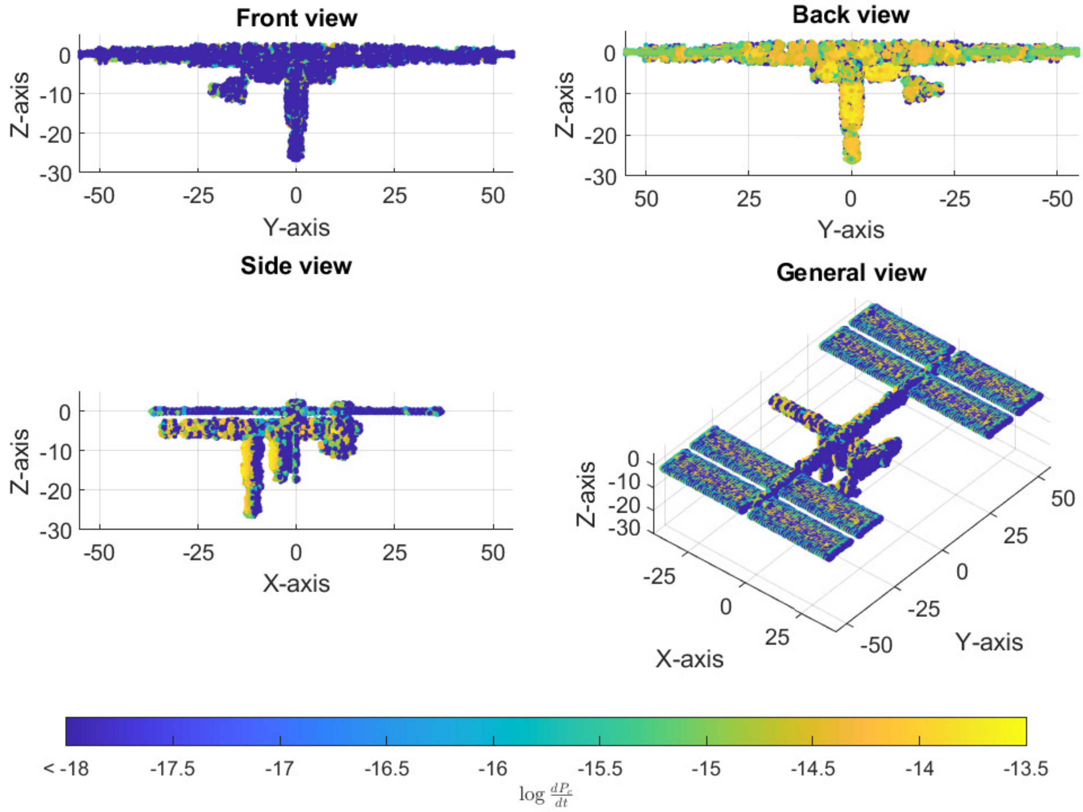


Figure 9.12: Collision probability rate between the ISS and SAT #49633 on each quadrature point evaluated at TCA.

- From the nominal crash scenario analysis, it is concluded that a reduction in initial state uncertainty of one order of magnitude yields to a reduction in collision probability of two orders of magnitude, due to uncertainty dilution.
- For the same scenario, an increase in lead time yields decrease in collision probability and increase in encounter time span, due to uncertainty dilution.
- It is crucial to include perturbations in the acceleration model to accurately calculate the collision risk and encounter time.
- The time of the Cosmos-2251/Iridium-33 is accurately predicted, although the collision probability is low. A more accurate initial state measurement is required to improve the risk estimate.
- For the Chang zheng-4C/Cosmos-2004 close encounter, the time of closest approach is correctly estimated. The collision risk, extrapolated to the initial state uncertainty of 50 meters, is slightly lower than the predictions by LeoLabs.
- From the screening of debris resulting from destruction of Cosmos-1408 over a period of 15 days, ten close encounters with the ISS are identified. From these, only one has a significant collision risk.
- Assuming a spherical encounter geometry, the collision probability calculated is above the allowed risk and a CAM should have been performed.
- Applying the multi-sphere model, the collision probability decreases by one order of magnitude when the full body is considered and by two orders of magnitude when the solar panels are neglected.
- For this scenario, the highest risk of collision is detected on the rear side of the ISS.

10
Conclusions & recommendations

This final chapter outlines the conclusions and recommendations from the work performed in this thesis. In Section 10.1, the main findings from this study are summarised and the research questions posed in Section 1.3 are answered. Moreover, this section reflects about the contributions of the DA-GMM method of collision probability calculation to the scientific community and the limitations of this work. In Section 10.2, the recommendations for future work expanding or relating to the method developed here are provided.

10.1. Conclusions

This research has developed a new method to compute the collision probability between space objects that can be applied to any encounter geometry and can be extended to include the three-dimensional shape of the satellites. The main finding from this process is that it is possible to reduce the computational time required to perform high accuracy propagation of uncertainty by using the hybrid Differential Algebra and Gaussian Mixture model (DA-GMM) method. This allows to avoid the traditional assumptions on linearisation of the dynamics and normality of the state distribution. Moreover, this method can be easily combined with the direct approach for collision probability calculation, which does not apply any assumptions on the satellites position or velocity and therefore can be used for all types of encounter scenarios. By expanding the surface integral with the Lebedev quadrature to a multi-sphere model, the collision probability can be computed considering any shape, which improves the accuracy of the prediction.

Regarding the uncertainty propagation part, a key finding is that the accuracy can be customised by modifying the expansion order and number of Gaussian Mixture Elements, for a given lead time. An increase in lead time will require a higher number of Gaussian Mixture Elements and potentially higher expansion order, which increase the computational load. However, for a higher lead time, the available time to compute this probability also increases. With the DA-GMM method, a lead time of three days is achieved with very high accuracy, and a greater lead time can be achieved but with accuracy below the specified threshold of $L_2 = 0.6$. The L_2 is the variable used through this work to measure the error in the uncertainty distribution, as it represents the integral of the squared difference between two probability density functions.

Regarding the collision probability calculation process, it has been verified to accurately predict the time of the encounter and the collision probability with a high accuracy, improving in some scenarios the results form traditional methods. The time required to compute the collision probability between two objects ranges between a few seconds to approximately an hour depending on the accuracy desired, the encounter time and the number of time-steps considered. Some key findings from the test cases studied are the need to include perturbations in the dynamical model and the important effect of the initial state uncertainty on the detected collision risk. By applying the collision method to the real-life scenarios of the Cosmos-2251/Iridium-33 collision and Chang Zheng-4C/Cosmos-2004 close encounter, it was validated that the time of maximum probability lied within 0.2 seconds from the time of closest approach. Moreover, by computing the collision probability rate as a function of time, the collision risk can be evaluated at different moments of the encounter and the encounter period is also known. Although the test cases simulated match the observed behaviour, it would have been interesting to perform the simulations with more accurate tracking data to avoid the effect of uncertainty dilution.

Finally, the screening of close encounters from the debris of the Cosmos-1408 satellite resulting from the

Russian anti-satellite missile test (ASAT), provide relevant findings. First, it is concluded that this event will continue to pose a risk to the crew inhabiting the ISS. In the time-span of two weeks of data analysed, ten close encounters were identified that violate the bounding box defined by NASA to consider a high risk encounter. From these ten encounters, one of them was found to have a collision probability above the threshold that requires to perform a collision avoidance manoeuvre. Since no manoeuvre was performed, it is assumed that the collision probability with more accurate tracking data was below the threshold. For this case, the collision probability was calculated considering the three-dimensional shape of the ISS, leading to a reduction in one to two orders of magnitude on the collision risk depending on the inclusion of the solar panels. With this decrease, a collision avoidance manoeuvre was no longer needed. This test confirms the utility of the multi-sphere model for collision probability, especially in the current times of increasing space debris population.

This thesis aimed to answer two research questions and corresponding sub-questions. The second sub-section was posed when the objective was to use the method of equivalent cross-section area (MECSA) to compute the collision probability for an arbitrary shape in two dimensions. This method was proposed by Chan (2008) and is developed for the short-term scenarios that consider a planar encounter geometry. Instead of using this method, a new method was developed in this work, which allows to compute the collision probability for an arbitrary shape and for any type of encounter. Therefore, research question *Q2* will be answered for the multi-sphere method instead of the MECSA. The research questions have been answered as follows:

Q1. *How can the hybrid DA-GMM method reduce the time to compute collision probability in LEO satisfying the operational requirements?*

Q1.1. *What is the error derived from propagating the trajectory using Taylor expansion integration in the DA framework?*
Depending on the Taylor expansion order, number of Gaussian Mixture elements and propagation time, the error ranges between $L_2 = 10^{-5}$ and $L_2 \approx 5$. The threshold for admissible error to achieve the desired accuracy is set to $L_2 = 0.6$.

Q1.2. *What environment and acceleration models can be included to meet the operational requirements?*
It is required to include the Earth gravitational acceleration with spherical harmonics up to degree and order six, the third-body, central gravitational acceleration from the Moon and the Sun and the aerodynamic acceleration.

Q1.3. *What is the maximum lead time that can be achieved within the operational requirements?*
The maximum lead time achieved is three days.

Q1.4. *How does the accuracy of the DA-GMM method relate to alternative collision detection algorithms?*
For long-term, low-velocity encounters, the method has proved to improve the accuracy in computing collision probability. For the cases studied, the error in collision probability calculation is reduced by 70% and 82%, respectively, with respect to the best result by alternative methods.

Q1.4. *Is the method able to predict the Iridium-33/Cosmos-2251 collision and the encounter time?*
Yes, the method detects the maximum collision probability within 0.2 seconds from the real encounter time and computes a collision probability of $\approx 10^{-6}$, 16 hours before the encounter and based on TE measurements. It is expected that data from closer to the encounter event and with higher tracking accuracy would have predicted a higher collision risk.

The second research question was regarding the method of cross sectional area to compute the collision probability accounting for the satellite shape. In this case, instead of applying this method, which is used for short-term encounters that assume a two-dimensional encounter geometry and therefore only consider a cross section of the satellite, the multi-sphere method has been developed. Therefore, these research questions will be answered for the multi-sphere method instead.

Q2. *How can the MECSA be applied to reduce the over-estimation of collision probability for the ISS?*

Q2.1. *What is the error derived from considering a constant attitude and only one cross section of the vehicle?*

In the multi-sphere method, this does not apply anymore. The full shape of the satellite is considered instead of a single cross section. The attitude of the satellite is considered to stay constant for this application, but this is not a limitation of the multi-sphere method developed. By applying frame transformations to the radial, transverse, normal (RTN) frame from the body frame, the attitude of the satellite at any time can be included in the model.

Q2.2. *Is the MECSA able to discard potential collisions identified by the verified interval orbit propagation that did not occur?*
Yes, for the collision detected between the ISS and the piece of debris from Cosmos-1408, applying the multi-sphere method reduces the risk in one to two orders of magnitude (depending on the inclusion of the solar panels), and therefore allows to discard the need for a collision avoidance manoeuvre.

Q2.3. *Can this method be used to determine the best collision avoidance strategy?*
Yes, this method provides the collision risk for each segment of the ISS and therefore allows to establish the best direction for an avoidance manoeuvre. Moreover, this process can be used, when a combination of debris are threatening the ISS.

The research questions have been successfully answered and the DA-GMM method of collision probability in combination with the multi-sphere model has proven to fill the research gap in computing the collision risk for low-velocity encounters in a timely manner and considering the real shape of the object. This software can be used independently with public tracking data and has the potential to be embedded in a system that takes sensor measurements in addition with tracking data.

Finally, it is important to restate the limitations of the DA-GMM method for collision probability. First, the minimum altitude that can be propagated is set to 400 km, since the method fails at lower altitudes due to the accumulation of error from the atmospheric perturbation. Moreover, the method cannot be applied in cases where the uncertainty (standard deviation) is in the same order of magnitude as the hard-body radius of the encounter. This is due to the numerical approximation from Lebedev's quadrature. Although this is not usually the case, it must still be taken into consideration. Moreover, by using the multi-sphere model, the geometry can be divided into smaller spheres if this was the case. Finally, it is difficult to include more degrees of spherical harmonics due to the exponential increase in computational load that it entails. Therefore, if the method was used to compute collision scenarios around another body (e.g., the Moon), that required more spherical harmonics, alternative solutions would be needed.

10.2. Recommendations

This section reflects on extensions of the method that could be developed in future work and elements of the research that could be more profoundly studied and have not been included due to time limitations. The following recommendations are made:

- The uncertainty in the atmospheric density and ballistic coefficient has been coded in the method, but has not been tested due to time constraints. It would be interesting to test the effect of adding this uncertainty and to check how it affects the results as a function of altitude.

- The weight of each Gaussian Mixture Element has been kept constant after the propagation of uncertainty. However, it is possible to update the weights by solving an additional optimisation problem. Although from the literature study it was concluded hat the error from keeping the weight constant is negligible, it would be interesting to study the effect in accuracy and computational time of updating the weights.

- The acceleration perturbations included are limited by the computational load of the propagation. Although the difference with respect to the full model has been found to be negligible, to obtain a more accurate propagation of uncertainty it is possible to include an extra variable representing the uncertainty in the dynamical model.

- The initial state uncertainty and propagation time have been found to have a large effect on the computed collision risk, due to the so-called uncertainty dilution. It would be interesting to find a method to quantify this effect and take these variables into account in the interpretation of the collision probability calculated.

- The multi-sphere method for collision probability has been developed assuming that the attitude of the satellite coincides with the RTN-frame. This simplification can be avoided by including the frame transformation from the body reference frame to the RTN-reference frame in the collision probability calculation.
- For the case of the ISS, the solar panels do not have a constant orientation. Instead, they rotate to align with the Sun vector. This can be added as a variable to model the shape of the ISS in the multi-sphere model and obtain an accurate collision probability calculation. This can be applied to any satellite, if the attitude is known.
- This collision probability calculation can be embedded in a system that incorporates sensors and a flight computer to perform these calculation autonomously on-flight.
- With the collision probability rate estimated on each node along the surface, it is possible to compute the direction for a collision avoidance manoeuvre in the scenario that multiple debris are threatening the vehicle simultaneously.

Bibliography

Adurthi, N. and Singla, P. (2015). “Conjugate Unscented Transformation-Based Approach for Accurate Conjunction Analysis”. In: *Journal of Guidance, Control, and Dynamics* Vol. 38, No. 9, pp. 1–17.

Agapov, V., Fateev, V., Sukhavnov, S., Burtsev, Y., Khutorovsky, Z., Kamensky, S., Stepanyants, V., and Samotokhin, A. (2009). “Collision Prediction for LEO Satellites. Analysis of Characteristics”. In: Advanced Maui Optical and Space Surveillance Technologies Conference.

Agu, F. and Francis, R. (2018). “Comparison of Goodness of Fit Tests for Normal Distribution”. In: *Asian Journal of Probability and Statistics* Vol. 1, No. 2, pp. 1–32.

Aida, S., Patzelt, T., Leushacke, L., Kirschner, M., and Kiehling, R. (2009). “Monitoring and Mitigation of Close Proximities in Low Earth Orbit”. In: 21st ISSFD conference.

Akella, M. R. and Alfriend, K. T. (2000). “Probability of Collision Between Space Objects”. In: *Journal of Guidance, Control, and Dynamics* Vol. 23, No. 5, pp. 769–772.

Alfano, S. (2004). “Relating Position Uncertainty to Maximum Conjunction Probability”. In: *Journal of the Astronautical Sciences* Vol. 53, No. 2, pp. 193–205.

Alfano, S. (2005). “A Numerical Implementation of Spherical Object Collision Probability”. In: *Journal of the Astronautical Sciences* Vol. 53, No. 1, pp. 103–109.

Alfano, S. (2006). *Addressing Nonlinear Relative Motion For Spacecraft Collision Probability.*

Alfano, S. (2007). “Beta Conjunction Analysis Tool”. In: AAS/AIAA Astrodynamics Specialist Conference. Vol. 07, pp. 19–23.

Alfano, S. (2009). “Satellite conjunction Monte Carlo analysis”. In: *Advances in the Astronautical Sciences* Vol. 134, pp. 2007–2024.

Altman, S. (1972). “A unified state model of orbital trajectory and attitude dynamics”. In: *Celestial Mechanics* Vol. 6, pp. 425–446.

Aristoff, J. and Poore, A. (2012). “Implicit Runge-Kutta Methods for Orbit Propagation”. In: AIAA/AAS Astrodynamics Specialist Conference.

Atkinson, K. and Han, W. (2012). *Spherical Harmonics and Approximations on the Unit Sphere: An Introduction.* Vol. 2044. Springer.

Badhwar, G. D. and Anz-Meador, P. D. (1989). “Determination of the area and mass distribution of orbital debris fragments”. In: *Earth, Moon, and Planets* Vol. 45, pp. 29–51.

Bai, X.-Z. and Chen, L. (2018). “Correlation characterization of orbital prediction error based on relative motion theory and its application”. In: 4th IAA Conference on Dynamics and Control of Space Systems.

Bergsma, M. (2015). “Application of Taylor Series Integration to Reentry Problems”. MSc Thesis. TU Delft.

Berz, M. (1999). “Chapter 2 - Differential Algebraic Techniques”. In: *Modern Map Methods in Particle Beam Physics.* Ed. by P. Hawkes. Vol. 108. Advances in Imaging and Electron Physics. Elsevier, pp. 81–117.

Bhusal, R. and Subbarao, K. (2018). “Uncertainty Quantification Using Generalized Polynomial Chaos Expansion for Nonlinear Dynamical Systems With Mixed State and Parameter Uncertainties”. In: *Journal of Computational and Nonlinear Dynamics* Vol. 14, No. 2, pp. 99–113.

Bhusal, R. and Subbarao, K. (2019). “Generalized Polynomial Chaos Expansion Approach for Uncertainty Quantification in Small Satellite Orbital Debris Problems”. In: *The Journal of the Astronautical Sciences* Vol. 67, No. 4, pp. 225–253.

Bradley, B. (2015). “Numerical Algorithms for Precise and Efficient Orbit Propagation and Positioning”. MSc Thesis. University of Colorado Boulder.

Bradshaw, G. and O’Sullivan, C. (2004). “Adaptive Medial-Axis Approximation for Sphere-tree Construction”. In: *ACM Transactions on Graphics* Vol. 23,

Bradshaw, G. and O’Sullivan, C. (2003). “Sphere-Tree Construction using Dynamic Medial Axis Approximation”. In: Symposium on Computer Animation, pp. 33–40.

Chan, F. (2008). *Spacecraft Collision Probability.* Aerospace Press.

Chan, K. (2004). “Spacecraft Collision Probability for Long-Term Encounters”. In: Vol. 116, No. 1, pp. 767–784.

Chan, K. (1997). “Collision Probability Analyses for Earth Orbiting Satellites”. In: 7th International Space Cpnference if Oacific-Basin Societies. Vol. 96, pp. 1033–1048.

Coppola, V. and McAdams, J. V. (2012). “Including Velocity Uncertainty in the Probability of Collision between Space Objects $AAS12-247$, 22nd, Spaceflight mechanics 2012”. In: Vol. 143, pp. 2159–2178.

Corbett, T. (2020). *LeoLabs indicates no collision of Soviet satellite and Chinese rocket stage*. `https://www.nasaspaceflight.com/2020/10/leolabs-tracking-high-risk-collision-probability/`. (Accessed on 12/09/2021).

DeMars, K., Bishop, R., and Jah, M. (2013). “Entropy-Based Approach for Uncertainty Propagation of Nonlinear Dynamical Systems”. In: *J. of Guidance, Control, and Dynamics* Vol. 36, No. 4, pp. 1047–1054.

DeMars, K., Cheng, Y., and Jah, M. (2014). “Collision Probability with Gaussian Mixture Orbit Uncertainty”. In: *Journal of Guidance, Control, and Dynamics* Vol. 37, No. 3, pp. 979–985.

Dempsey, R. (2018). *The International Space Station: Operating an Outpost in the New Frontier*. Chap. 8.

Di Lizia, P., Armellin, R., and Lavagna, M. (2008). “Application of high order expansions of two-point boundary value problems to astrodynamics”. In: *Celestial Mechanics and Dynamical Astronomy* Vol. 102, No. 4, pp. 355–375.

Doostan, A. and Owhadi, H. (2010). “A Non-adapted Sparse Approximation of PDEs with Stochastic Inputs”. In: *Journal of Computational Physics* Vol. 230, No. 8, pp. 3015–3034.

ESA (2008). *ESA - ATV: a very special delivery - Lesson notes*. `https://www.esa.int/Education/Space_In_Bytes/ATV_a_very_special_delivery_-_Lesson_notes`.

ESA (2021a). *ESA - Analysis and prediction*. `https://www.esa.int/Safety_Security/Space_Debris/Analysis_and_prediction`. (Accessed on 01/10/2022).

ESA (2021b). *Space Environment Statistics · Space Debris User Portal*. (Accessed on 01/27/2021).

Eshagh, M. and Alamdari, M. (2007). “Perturbations in orbital elements of a low earth orbiting satellite”. In: *J. Earth Space Phys.* Vol. 33, No. 1, pp. 1–12.

Flohrer, T., Krag, H., and Klinkrad, H. (2008). “Assessment and Categorization of TLE Orbit Errors for the US SSN Catalogue”. In: Advanced Maui Optical and Space Surveillance Technologies Conference.

Flores, R., Burhani, B. M., and Fantino, E. (2021). “A method for accurate and efficient propagation of satellite orbits: A case study for a Molniya orbit”. In: *Alexandria Engineering Journal* Vol. 60, No. 2, pp. 2661–2676.

Fortescue, P., Swinerd, G., and Stark, J. (2011). *Spacecraft Systems Engineering*. Wiley.

Foster, J. and Estes, H. (1992). *A Parametric Analysis of Orbital Debris Collision Probability and Maneuver Rate for Space Vehicles*. NASA, National Aeronautics and Space Administration, Lyndon B. Johnson Space Center.

Gautschi, W. (2011). “Numerical Analysis”. In: SpringerLink : Bücher. Birkhäuser Boston, pp. 73–77.

Geul, J., Mooij, E., and Noomen, R. (2017). “TLE uncertainty estimation using robust weighted differencing”. In: *Advances in Space Research* Vol. 59, No. 10, pp. 2522–2535.

Hedin, A. E. (1988). *High altitude atmospheric modeling*. National Aeronautics, Space Administration, Scientific, and Technical Information Office.

Henze, N. and Zirkler, B. (1990). “A class of invariant consistent tests for multivariate normality”. In: *Communications in Statistics-theory and Methods* Vol. 19, pp. 3595–3617.

Hilton, S., Cairola, F., Gardi, A., Sabatini, R., Pongsakornsathien, N., and Ezer, N. (2019). “Uncertainty Quantification for Space Situational Awareness and Traffic Management”. In: *Sensors* Vol. 19, p. 4361.

Hintz, G. (2008). “Survey of Orbit Element Sets”. In: *Journal of Guidance, Control, and Dynamics* Vol. 31, No. 3, pp. 785–790.

Hoogendoorn, R. (2016). “Statistical Impact Prediction of Space Debris. The Uncertainty Propagation Approach”. MSc Thesis. TU Delft.

Hoogendoorn, R., Mooij, E., and Geul, J. (2017). “Uncertainty Propagation for Statistical Impact Prediction of Space Debris”. In: *Advances in Space Research* Vol. 61,

Horstmann, A., Kebschull, C., Müller, S., Gamper, E., Hesselbach, S., Soggeberg, K., Ben Larbi, M. K., Becker, M., Lorenz, J., Wiedemann, C., and Stoll, E. (2018). “Survey of the Current Activities in the Field of Modeling the Space Debris Environment at TU Braunschweig”. In: *Aerospace* Vol. 5, No. 2, p. 9.

Horwood, J., Aragon, N., and Poore, A. (2011). “Gaussian Sum Filters for Space Surveillance: Theory and Simulations”. In: *Journal of Guidance, Control, and Dynamics* Vol. 34, No. 6, pp. 1839–1851.

IADC (2020). *IADC Space Debris Mitigation Guidelines, Revision 2*.

ISO Central Secretary (2019). *Space systems — Space debris mitigation requirements*. Standard. Geneva, CH: International Organization for Standardization.

Isserlis, L. (1918). “On a Formula for the Product-Moment Coefficient of any Order of a Normal Frequency Distribution in any Number of Variables”. In: *Biometrika* Vol. 12, No. 1/2, pp. 134–139.

Jones, B. and Doostan, A. (2013). “Satellite collision probability estimation using polynomial chaos expansions”. In: *Advances in Space Research* Vol. 52, No. 11, pp. 1860–1875.

Kelso, T. (2009). “Analysis of the Iridium 33Cosmos 2251 Collision”. In: *Proceedings of the 19th AIAA/AAS Astrodynamics Specialist Conference* Vol. 135, pp. 1–11.

Kessler, D. J. and Cour-Palais, B. G. (1978). “Collision frequency of artificial satellites: The creation of a debris belt”. In: *Journal of Geophysical Research: Space Physics* Vol. 83, No. A6, pp. 2637–2646.

Klees, R. and Dwight, R. (2020). *AE2220-I Applied Numerical Analysis.* Lecture notes. TU Delft.

Kumar, M. (2012). “Accurate Propagation of Initial Uncertainty in the Two Body Problem”. In: *AIAA/AAS Astrodynamics Specialist Conference 2012.*

Le Maître, O. and Knio, O. (2010). *Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics.*

Lebedev, V. (1976). “Quadratures on a sphere”. In: *USSR Computational Mathematics and Mathematical Physics* Vol. 16, No. 2, pp. 10–24.

LeoLabs (2021). `https://platform.leolabs.space/system_metrics`. (Accessed on 12/07/2021).

Lunghi, P., Armellin, R., Lizia, P. D., Mease, K. D., and Lavagna, M. R. (2018). “Atmospheric Entry Guidance Based on Differential Algebra for High Elevation Mars Landing”. In: *2018 Space Flight Mechanics Meeting.*

Luo, Y.-z. and Yang, Z. (2017). “A review of uncertainty propagation in orbital mechanics”. In: *Progress in Aerospace Sciences* Vol. 89, pp. 23–39.

Massari, M., Di Lizia, P., and Rasotto, M. (2017). “Nonlinear Uncertainty Propagation in Astrodynamics Using Differential Algebra and Graphics Processing Units”. In: *Journal of Aerospace Information Systems* Vol. 14, No. 9, pp. 493–503.

Matsumoto, M. and Nishimura, T. (1998). “Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator”. In: *ACM Trans. Model. Comput. Simul.* Vol. 8, No. 1, pp. 3–30.

McKinley, D. (2006). “Development of a Nonlinear Probability of Collision Tool for the Earth Observing System”. In: *AIAA/AAS Astrodynamics Specialist Conference and Exhibit.* eprint: `https://arc.aiaa.org/doi/pdf/10.2514/6.2006-6295`.

Merz, K., Virgili, B. B., Braun, V., Flohrer, T., Funke, Q., Krag, H., and Lemmens, S. (2017). “Current Collision Avoidance service by ESA’s Space Debris Office”. In: 7th European Conference on Space Debris.

Montenbruck, O., Gill, E., and Lutze, F. (2002). “Satellite orbits: models, methods, and applications”. In: *Appl. Mech. Rev.* Vol. 55, No. 2, B27–B28.

Mooij, E. (2016). *AE4870B Re-entry systems.* Lecture notes. TU Delft.

Moore, R. E. (1968). “Practical aspects of interval computation”. In: *Applications of Mathematics* Vol. 13, pp. 52–92.

Morselli, A., Armellin, R., Di Lizia, P., and Bernelli-Zazzera, F. (2012). “Computing Collision Probability using Differential Algebra and Advanced Monte Carlo Methods”. In: *Proceedings of the International Astronautical Congress, IAC* Vol. 3,

NASA (2020). *NASA Spacecraft Conjunction Assessment and Collision Avoidance Best Practices Handbook.*

NASA (2021a). *NASA - NSSDCA - Cosmos-1408 - Details.* `https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=1982-092A`. (Accessed on 12/10/2021).

NASA (2021b). *NASA to Air Spacewalk to Swap Communications Antenna on Space Station | NASA.* `https://www.nasa.gov/press-release/nasa-to-air-spacewalk-to-swap-communications-antenna-on-space-station`. (Accessed on 12/10/2021).

Papanikolaou, T. and Tsoulis, D. (2016). “Assessment of numerical integration methods in the context of low Earth orbits and intersatellite observation analysis”. In: *Acta Geodaetica et Geophysica* Vol. 51, No. 4, pp. 619–641.

Park, R. and Scheeres, D. (2007). “Nonlinear Semi-Analytic Methods for Trajectory Estimation”. In: *Journal of Guidance Control and Dynamics* Vol. 30, No. 6, pp. 1668–1676.

Patera, R. P. (2001). “General Method for Calculating Satellite Collision Probability”. In: *Journal of Guidance, Control, and Dynamics* Vol. 24, No. 4, pp. 716–722.

Patera, R. P. (2003). “Satellite Collision Probability for Nonlinear Relative Motion”. In: *Journal of Guidance, Control, and Dynamics* Vol. 26, No. 5, pp. 728–733.

Patera, R. P. (2005). “Calculating Collision Probability for Arbitrary Space Vehicle Shapes via Numerical Quadrature”. In: *Journal of Guidance, Control, and Dynamics* Vol. 28, No. 6, pp. 1326–1328. eprint: `https://doi.org/10.2514/1.14526`.

Pavlis, N. K., Holmes, S. A., Kenyon, S. C., and Factor, J. K. (2012). “The development and evaluation of the Earth Gravitational Model 2008 (EGM2008)”. In: *Journal of Geophysical Research: Solid Earth* Vol. 117, No. B4, Picone, J., Hedin, A., Drob, D., and Aikin, A. (2002). "NRLMSISE-00 empirical model of the atmosphere: Statistical comparison and scientific issues". In: *Journal of Geophysical Research* Vol. 107, No. A12, pp. 1468–1483.

Psiaki, M. L., Schoenberg, J. R., and Miller, I. T. (2015). "Gaussian Sum Reapproximation for Use in a Nonlinear Filter". In: *Journal of Guidance, Control, and Dynamics* Vol. 38, No. 2, pp. 292–303.

Rabbath, C. and Corriveau, D. (2019). "A comparison of piecewise cubic Hermite interpolating polynomials, cubic splines and piecewise linear functions for the approximation of projectile aerodynamics". In: *Defence Technology* Vol. 15, No. 5, SI: 2019 International Symp. Ballistics, pp. 741–757.

Rasotto, M., Morselli, A., Wittig, A., Massari, M., Di Lizia, P., Armellin, R., Valles, C., and Ortega, G. (2016). "Differential algebra space toolbox for nonlinear uncertainty propagation in space dynamics". In: 6th International Conference on Astrodynamics Tools and Techniques (ICATT).

Razali, N. and Yap, B. (2011). "Power Comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson Darling Tests". In: *Joutnal of Statistical Modelling and Analytics* Vol. 2, No. 1, pp. 21–33.

Reeger, J. A., Fornberg, B., and Watts, M. L. (2016). "Numerical quadrature over smooth, closed surfaces". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* Vol. 472, No. 2194, p. 20160401.

Römgens, B. (2011). "Verified interval orbit propagation with an application to satellite collision detection". MSc Thesis. TU Delft.

Rongzhi, Z. and Kaizhong, Y. (2020a). "5 - Spacecraft collision warning orbit calculation method". In: *Spacecraft Collision Avoidance Technology*. Ed. by Z. Rongzhi and Y. Kaizhong. Academic Press, pp. 123–154.

Rongzhi, Z. and Kaizhong, Y. (2020b). "Outline of spacecraft collision warning". In: *Spacecraft Collision Avoidance Technology*. Ed. by Z. Rongzhi and Y. Kaizhong. Academic Press, pp. 1–9.

Ronse, A. and Mooij, E. (2013). "Statistical Impact Prediction of Decaying Objects". In: *Journal of Spacecraft and Rockets* Vol. 51, No. 6, pp. 1797–1810.

Rylaarsdam, J. B. (1996). *International Space Station traffic modeling and simulation.*

Sabol, C., Binz, C., Segerman, A., Roe, K., and Jr, P. (2012). "Probability of collision with special perturbations dynamics using the Monte Carlo method". In: *Advances in the Astronautical Sciences* Vol. 142, pp. 1081–1093.

Shum, C., Abusali, P., Lee, H., Ogle, J., Raney, R., Ries, J., Smith, W., Svehla, D., and Zhao, C. (2008). "Orbit determination requirements for abyss: The proposed science payload on space station". In: *Advances in the Astronautical Sciences* Vol. 130, pp. 1207–1218.

Shuster, S. (2017). "A Survey and Performance Analysis of Orbit Propagators for LEO, GEO, and Highly Elliptical Orbits". MSc Thesis. Utah State University.

Silvey, S. D. (1959). "S. N. Roy, Some Aspects of Multivariate Analysis (John Wiley &amp; Sons), 214 pp., £3, 4s." In: *Proceedings of the Edinburgh Mathematical Society* Vol. 11, No. 3, pp. 191–192.

Sobolev, S. (1962). "Cubature formulas on a sphere invariant under the dihedral group D2h". In: *Siberian Electronic Mathematical Reports* Vol. 3, pp. 1307–1310.

Somodi, B. and Foldvary, L. (2011). "Application of numerical integration techniques for orbit determination of state-of-the-art LEO satellites". In: *Periodica Polytechnica Civil Engineering* Vol. 55, No. 2, pp. 99–106.

SpaceTrack (2020). *Spaceflight Safety Handbook for Satellite Operators.*

Sun, Z.-J., Luo, Y.-Z., di Lizia, P., and Zazzera, F. B. (2019). "Nonlinear orbital uncertainty propagation with differential algebra and Gaussian mixture model". In: *Science China Physics, Mechanics, and Astronomy* Vol. 62, No. 3, pp. 1–11.

Terejanu, G., Singla, P., Singh, T., and Scott, P. (2008). "Uncertainty Propagation for Nonlinear Dynamic Systems Using Gaussian Mixture Models". In: *AIAA Journal of Guidance, Control and Dynamics* Vol. 31, No. 6, pp. 1623–1633.

Thomopoulos, N. (2013). *Essentials of Monte Carlo simulation: Statistical methods for building simulation models*, pp. 1–171.

U.S. Space Command (2021). *Russian direct-ascent anti-satellite missile test creates significant, long-lasting space debris > United States Space Command > Article Display*. `https://www.spacecom.mil/News/Article-Display/Article/2842957/russian-direct-ascent-anti-satellite-missile-test-creates-significant-long-last/`. (Accessed on 12/28/2021).

Vallado, D. and Crawford, P. (2008). "SGP4 Orbit Determination". In:

Valli, M., Armellin, R., Di Lizia, P., and Lavagna, M. R. (2013). "Nonlinear Mapping of Uncertainties in Celestial Mechanics". In: *Journal of Guidance, Control, and Dynamics* Vol. 36, No. 1, pp. 48–63.

Vishwajeet, K., Singla, P., and Jah, M. (2014). “Nonlinear Uncertainty Propagation for Perturbed Two-Body Orbits”. In: *Journal of Guidance, Control, and Dynamics* Vol. 37, No. 5, pp. 1415–1425.

Vittaldev, V., Mooij, E., and Naeije, M. (2012). “Unified State Model theory and application in Astrodynamics”. In: *Celestial Mechanics and Dynamical Astronomy* Vol. 112, No. 3, pp. 253–282.

Vittaldev, V. and Russell, R. (2016). “Space Object Collision Probability Using Multidirectional Gaussian Mixture Models”. In: *Journal of Guidance, Control, and Dynamics* Vol. 39, No. 9, pp. 1–7.

Vittaldev, V., Russell, R., and Linares, R. (2016). “Spacecraft Uncertainty Propagation Using Gaussian Mixture Models and Polynomial Chaos Expansions”. In: *Journal of Guidance, Control, and Dynamics* Vol. 39, No. 12, pp. 2161–2167.

Wakker, K. (2015). *AE4874-I Fundamentals of Astrodynamics - Lecture notes*. TU Delft.

Weisman, R., Majji, M., and Alfriend, K. T. (2016). “Solution of Liouville’s Equation for Uncertainty Characterization of the Main Problem in Satellite Theory”. In: *Computer Modeling in Engineering & Sciences* Vol. 111, No. 3, pp. 269–304.

Wermuth, M., Montenbruck, O., Helm, A., and Cacciapuoti, L. (2012). “Precise Orbit Determination and Prediction of the ISS in the frame of the ACES Mission”. In: 6th ESA Workshop on Satellite Navigation Technologies & European Workshop on GNSS Signals and Signal Processing.

Wertz, J., Space Technology (U.S.), N. C. for, and (U.S.), N. R. L. (2001). *Mission Geometry: Orbit and Constellation Design and Management : Spacecraft Orbit and Attitude Systems*. Space technology library. Microcosm Press.

Yang, Z., Luo, Y.-Z., Zhang, J., and Tang, G.-J. (2016). “Uncertainty Quantification for Short Rendezvous Missions Using a Nonlinear Covariance Propagation Method”. In: *Journal of Guidance, Control, and Dynamics* Vol. 39, No. 9, pp. 2170–2178.

A

Numerical methods

This chapter introduces the numerical methods that are required for the development of the DA-GMM method and can be found in generic engineering applications. Section A.1 explains the procedure for finding the roots of a function, required to solve the optimisation process and convert from Cartesian components to Kepler elements. In Section A.2, the interpolation methods used to compute the trajectory of the satellites between the numerically integrated points, are explained. Section A.3 details the numerical integration methods and their settings and Section A.4 treats the statistical sampling and solving techniques used in this work.

A.1. Root-finding methods

A root finding method is an algorithm for calculating the values x for which a function $f(x) = 0$, also called the *roots* or *zeroes*. This functionality is commonly used in any engineering problem. In this case it is required to calculate the optimisation problem for a general GMM splitting, for the GMM splitting of a uniform distribution and to solve the eccentric anomaly to calculate the position of the satellite in the orbit. To be able to cover a range of root-finding problems, the three most general methods which are implemented in MatLab, Python are introduced in this section through a pseudo-code demonstrating how to implement them and showing their logic.

A.1.1. Bisection method

The bisection method is one of the simplest root finding algorithms and one of the most widely used. The method is very simple but in turn it can be very slow to reach the converged solution. As a result, this method will be used to obtain an initial approximation of the root and use it as the initial value for other root-finding schemes. The method is applied to continuous functions, generally in one dimension although it can be

Algorithm 1 Bisection method

```
    **Input:** Function $f$, initial values $a, b$, tolerance $\epsilon$, maximum iterations $n_{max}$
    **Output:** Root = approximate root with tolerance less than $\epsilon$
procedure CODE
    $n = 1$
    while $n \leq n_{max}$ do
        $c = (a + b)/2$
        if $f(c) = 0$ or $(b - a)/2 < \epsilon$ then
            $root = c$
            end
        if sign($f(c)$ = sign($f(a)$) then
            $a = c$
        else
            $b = c$
        $n = n + 1$
```

Algorithm 2 Newton-Raphson method

```
Input: Function $f$, function derivative $f'$, initial value $x_0$, tolerance $\epsilon$, maximum iterations $n_{max}$
Output: Root = approximate root with tolerance less than $\epsilon$
procedure CODE
    $n = 1$
    while $n \le n_{max}$ do
        $y = f(x_0)$
        $y' = f'(x_0)$
        $x_1 = x_0 - \frac{y}{y'}$
        if $|x_1 - x_0| < \epsilon$ then
            $root = x_1$
            end
        $x_0 = x_1$
        $n = n + 1$
```

expanded to multiple dimensions with a considerable increase in complexity. In this case, only the one-dimensional case is considered.

This algorithm is based on the intermediate value theorem, which states that if a continuous function is evaluated a two points a, b that take opposite values, there exists a root in the interval $[a, b]$. The bisection method starts with two points with different function evaluation signs and calculates the midpoint $c = \frac{a+b}{2}$ and the function evaluation $f(c)$. If $f(c)$ is zero then the root is already found. Otherwise, wither $f(a)$ or $f(b)$ has opposite sign to $f(c)$. The sub-interval which contains a root crossing (different sign function evaluation) is selected and the process is repeated. In each step, the width of the interval is reduced to half. The process is repeated until the width of the interval is smaller than the tolerance specified. Algorithm 1 presents a pseudo-code of the bisection method.

The main benefit of this algorithm is that only the function has to be evaluated, and there is no need to compute any function derivatives. Therefore this algorithm will be used in cases where a one-dimensional solution is required and where the required tolerance allows to obtain the solution without a high number of iterations. More complex algorithms are introduced for the cases where these conditions don't apply.

A.1.2. Newton-Raphson method

The Newton-Raphson method was derived in the 17^{th} century to solve root-finding problems of continuous differentiable variables. This technique is powerful since the convergence speed is quadratic. This method will be used to solve the optimisation problem that allows to calculate the standard deviation of each GME required to split a uniform distribution into a GMM.

Starting with an initial guess (which for the method to converge fast it should be close to the root), the function at this point is approximated by its tangent and the $x-$intercept is calculated. This value will be a closest approximation to the root and will be used as the initial point for the next iteration. Algorithm 2 introduces the pseudo-code for this method.

The main drawback of this method is that it requires to calculate the function derivative $f'(x)$ analytically. In some cases this calculation might not be available or might be computationally prohibitive to calculate in a recursive way. The secant method presents a solution to this problem. A second drawback of this method is that it can fail to converge due to the initial point selected, to overshoot or to encountering a stationary point. The initial point problem can be overcome by using a simpler method (e.g. bisection method) to calculate a first estimate of the root. The overshoot issue is consists in the divergence of first derivative for a given point. Finally, if a stationary point is encountered ($f'(x) = 0$), the tangent line estimate encounters a division by zero and the method fails.

A.1.3. Secant method

When the derivative of the function $f(x)$ cannot be easily calculated, an approximation can be used in the root-finding algorithm. This applies to the optimisation problem required to update the weights of a GMM. In this case, the objective function is already difficult to solve and requires numerical integration. Calculating the derivative is not a feasible option, and the secant method will be used if required to solve this problem. This method approximates the derivative to a straight line between two points of the function. Compared to the Newton-Raphson method, the secant method takes more iterations (order of convergence is ≈ 1.618 instead of 2). However, the overall cost of the method can be smaller than for Newton-Raphson's method depending on the cost of evaluating the derivative. In the secant method, the function derivative at point x_1 is evaluated using the finite difference approximation:

$$f'(x_1) \approx \frac{f(x_1) - f(x_0)}{x_1 - x_0} \tag{A.1}$$

and therefore, the calculation of the root x_2 starting from the initial values x_0 and x_1 is calculated following:

$$x_2 = x_1 - f(x_1)\frac{x_1 - x_0}{f(x_1) - f(x_0)} \tag{A.2}$$

and similarly to the Newton-Raphson algorithm, the process is iterated until $|x_n - x_{n-1}| < \epsilon$.

A.2. Interpolation methods

Interpolation is required to calculate the values of a function at an arbitrary point x when only a set of discrete data points of the function is known. This often happens in trajectory propagation problems, since many of the parameters defining the environmental model of the propagation are tabulated values. Moreover, when solving a problem by numerical integration, the solution is only calculated at a set of points determined by the integration step and interpolation will be required if the function evaluation is required at other values.

In the problem under study, the environmental model includes a set of parameters which are tabulated, such as the ephemeris of third bodies (Moon and Sun), the time variation of the gravity field coefficients and the parameters defining the atmosphere model. In aerodynamic software, the interpolator used for each of these variables can be customised. In the case of Tudat, a sixth order Lagrange interpolator is selected by default for the ephemeris and spherical harmonics. For the atmospheric models a linear interpolator is used by default. Based on the current information on the problem, these default interpolation methods will be kept.

For the result of numerical integration, interpolation is required. Two situations can be recognised. First, interpolation of the target and chaser trajectories to find the TCA. Second, to integrate the collision probability over a period of time, for which the numerical integration step size might be insufficient and more data points can be required. In his case, cubic splines are recommended as the interpolation method. Two types of cubic spline are commonly used: standard and Hermite cubic splines. Bergsma (2015) argues that standard splines are more suitable to integrate the results of numerical integration, since the method produces more accurate results when the original data is smooth. However, the method can encounter under- and over-shooting issues which reduce the accuracy. Both methods will be considered for the interpolation of the numerical integration results and the most accurate will be selected after testing. In this section, the interpolation methods that will be used in the problem are introduced following the description by Klees and Dwight (2020).

A.2.1. Linear interpolation

Linear interpolation is the simplest method to obtain the values of a function f which is only known at discrete points. The method is based in fitting the points using linear polynomials, in a way that the linear interpolant becomes a set of straight lines joining the known points. This approach is very limited but it can be useful for cases where the function to approximate is quasi linear or when a large number of points is known in a way that the linear approximation between points is acceptable. To solve for a point $y = f(x)$ which is located in the interval bounded by two known values $y_0 = f(x_0)$ and $y_1 = f(x_1)$, the following expression is applied:

$$y = y_0 + (x - x_0)\frac{y_1 - y_0}{x_1 - x_0} \tag{A.3}$$

By default, Tudat uses this method to interpolate the values of the atmospheric density obtained from the tabulated results of the NRLMSISE-00 model. Since Tudat has been thoroughly tested and verified, it is assumed that this method will be valid for this application and will be kept as the method of choice.

A.2.2. Standard cubic spline interpolation

A cubic spline interpolation is the most widely used algorithm for function interpolation. The underlying idea is similar to the linear interpolation: to connect the known points with a polynomial. In this case, a polynomial of order three is used which allows to better approximate the function since it solves one of the main limitations of the linear method, that the derivative is usually discontinuous. Using third order polynomials allows to match the first and second order derivatives of the splines connecting at a point in addition to the values. This method is chosen to interpolate the trajectory of the satellites to obtain the closest approach.

Let's define a problem where a total of $p+1$ points (nodes) are known. This allows to fit a total of p polynomials P_i, which should satisfy the conditions $P_i(x) = P_{i+1}(x)$, $P_i'(x) = P_{i+1}'(x)$ and $P_i''(x) = P_{i+1}''(x)$ for $i \neq p$. The general form of a general cubic polynomial P_i and its first and second derivatives is:

$$P_i(x) = a_i\,(x-x_i)^3 + b_i\,(x-x_i)^2 + c_i\,(x-x_i) + d_i \tag{A.4}$$

$$P_i'(x) = 3a_i\,(x-x_i)^2 + 2b_i\,(x-x_i) + c_i \tag{A.5}$$

$$P_i''(x) = 6a_i\,(x-x_i) + 2b_i \tag{A.6}$$

From this relation, when the function is evaluated at each node x_i, $P_i(x_i) = d_i = f(x_i)$, where $f(x_i)$ is the known value of the node. From the relation $P_i''(x_i) = P_{i+1}''(x_i)$ and $P_i''(x_i) = 2b_i = M_i/2$ where M_i is the yet unknown value of the second derivative at node i, the coefficient a_i is:

$$a_i = \frac{M_{i+1} - M_i}{6(x_{i+1} - x_i)} = \frac{M_{i+1} - M_i}{6h_i} \tag{A.7}$$

where $h_i = x_{i+1} - x_i$. Inserting these expressions for a_i, b_i and d_i the expression for $P_{i+1}(x_{i+1})$, the expression for c_i is obtained as:

$$c_i = \frac{f_{i+1} - f_i}{h_i} - \frac{h_i}{3}M_i - \frac{h_i}{6}M_{i+1} \tag{A.8}$$

And Equation (A.4) becomes:

$$P_i(x) = \frac{M_{i+1} - M_i}{6h_i}\,(x-x_i)^3 + \frac{M_i}{2}\,(x-x_i)^2 + \left(\frac{f_{i+1} - f_i}{h_i} - \frac{h_i}{3}M_i - \frac{h_i}{6}M_{i+1}\right)(x-x_i) + f_i \tag{A.9}$$

The remaining task is to compute the values of the second derivative at each node M_i. From the first derivative condition $P_i'(x_i) = P_{i+1}'(x_i)$, the following system of $p-1$ equations and $p+1$ unknowns is defined:

$$\frac{h_{i-1}}{6}M_{i-1} + \frac{(h_{i-1} + h_i)}{3}M_i + \frac{h_i}{6}M_{i+1} = \frac{f_{i+1} - f_i}{h_i} - \frac{f_i - f_{i-1}}{h_{i-1}} \tag{A.10}$$

To find a unique solution, two constraints should be imposed. Several approaches can be defined, such as the natural cubic spline which sets the second derivatives at endpoints to zero.

A.2.3. Hermite cubic spline interpolation

Although cubic spline interpolation is one of the most widely used methods, it presents an issue when the derivatives of the function at the known points present fast changes. This results in under- and over-shooting which impacts the accuracy of the interpolation. Hermite interpolation highly reduces this problem (Rabbath and Corriveau, 2019). In this case, instead of imposing the condition of equal first and second derivatives at the connecting splines, the first derivative is at the data points is known and the first order derivative of the splines are matched to this value. Higher order Hermite interpolation matches higher order derivatives to its value at the data points. This approach allows to preserve two important characteristics of the data point. First, the spline is monotonic in intervals where the data points are monotonic (strictly increasing or decreasing). Second, the minimums and maximums of the spline match the minimums and maximums of the data points. For cubic spline polynomials, P_i, four conditions should be satisfied at each point:

$$P_i(x_i) = P_{i+1}(x_i) = f(x_i) = f_i \qquad P_i'(x_i) = P_{i+1}'(x_i) = f'(x_i) = f_i' \tag{A.11}$$

The interpolation polynomial between node i and node $i+1$ can be written with Hermite basis functions $h_{00}, h_{10}, h_{01}, h_{11}$ in the form: $P_i(x) = h_{00}(x)f_i + h_{10}(x)f_i' + h_{01}(x)f_{i+1} + h_{11}(x)f_{i+1}'$. For an arbitrary interval interpolating x between the nodes x_i and x_{i+1}, this expression is:

$$\begin{aligned} P_i(x) =& f_i\left(1 + 2\frac{x - x_i}{x_{i+1} - x_i}\right)\left(\frac{x - x_{i+1}}{x_i - x_{i+1}}\right)^2 + f_i'\left(\frac{x - x_i}{x_{i+1} - x_i}\right)\left(\frac{x - x_{i+1}}{x_i - x_{i+1}}\right)^2 \\ &+ f_{i+1}\left(1 + 2\frac{x - x_{i+1}}{x_i - x_{i+1}}\right)\left(\frac{x - x_i}{x_{i+1} - x_i}\right)^2 + f_{i+1}'\left(\frac{x - x_{i+1}}{x_{i+1} - x_i}\right)\left(\frac{x - x_i}{x_{i+1} - x_i}\right)^2 \end{aligned} \tag{A.12}$$

Which can be represented in the form of Equation (A.4), with coefficients:

$$\begin{aligned} a_i &= 2\tfrac{f_i - f_{i+1}}{h_i^3} + \tfrac{f'_i + f'_{i+1}}{h_i^2} \\ b_i &= 3\tfrac{f_{i+1} - f_i}{h_i^2} - 2\tfrac{f'_i + f'_{i+1}}{h_i} \\ c_i &= f'_i \\ d_i &= f_i \end{aligned} \tag{A.13}$$

where again $h_i = x_{i+1} - x_i$.

The main drawback of this interpolation method is that it requires the knowledge of the function derivatives at all the data points. This information is not always available. In the case of interpolating the trajectory obtained from numerical integration, the derivative is known at any time. Therefore, this method can be easily applied and is the best option to reduce under- and over- shooting the results. This method is selected to interpolate the trajectories, both to obtain an integrated value of the collision probability and to find the TCA.

A.2.4. Lagrange interpolation

The use of Lagrange polynomials allow to create an interpolator by a polynomial of order k. In Tudat, by default interpolation of order six is applied to the tabulated data from the ephemeris and spherical harmonics. This interpolation method will be kept for these applications. For the complete proof and derivation of this method, see the work by Gautschi (2011).The basis of this method is the Lagrange polynomial:

$$\ell_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{k} \frac{x - x_j}{x_i - x_j} = \frac{(x - x_0)}{(x_i - x_0)} \cdots \frac{(x - x_{i-1})}{(x_i - x_{i-1})} \frac{(x - x_{i+1})}{(x_i - x_{i+1})} \cdots \frac{(x - x_k)}{(x_i - x_k)}, i = 0, 1, \ldots, k \tag{A.14}$$

This function evaluation results in $\ell_i(x_i) = 1$ and for all other points $j \neq i$, $\ell_i(x_j) = o$ since there is always a numerator $x_j - x_j$.

By multiplying this polynomial by the value at each node $f_i = f(x_i)$, the Laplace polynomial produces the correct value at x_i. If the polynomials are added, the resulting polynomial correctly interpolates all the points:

$$p(x) = \sum_{i=0}^{k} f_i \ell_i(x), \tag{A.15}$$

This polynomial of order k can be used to approximate a function of $k + 1$ points. Since the total number of discrete points will usually be larger than $k+1$, a set of interpolating polynomials will be defined to interpolate the complete function.

A.3. Numerical integration methods

The approach suggested in this work is to propagate the trajectory uncertainty by means of Taylor series integration using differential algebra. This is itself a numerical integration procedure that does not require the traditional numerical integration routines. However, traditional integration will still be required to propagate the trajectory using a Monte Carlo method as means of verification.

Numerical integration algorithms have been developed as a way to solve ordinary differential equations subject to initial conditions. Multiple algorithms have been developed, and their performance depends on the problem to solve. Therefore there is not a single answer to what is the best integrator to use. In this section, a brief description of the available integration methods is provided, followed by a trade-off to select the best integrator type for the reference mission.

A.3.1. Description of numerical integration methods

A general description of the numerical integrator methods is provided, tailored to the methods available in Tudat. Three type of integrators are considered: multi-stage, multi-step and extrapolation methods, represented by Runge-Kutta (RK), Adams-Bashforth-Moulton (ABM) and Bulirsch-Stoer (BS) respectively. For each method, a description of the algorithm and its benefits and drawbacks is provided.

Multi-stage integration

In multi-stage methods, every time step requires multiple function evaluations. The Runge-Kutta family is one of the most widely used integrator methods. It can be used with fixed or variable time step The general structure of a Runge-Kutta algorithm is (Klees and Dwight, 2020):

$$y(t_{n+1}) = y(t_n) + \sum_{i=1}^{N} c_i k_i \tag{A.16}$$

where N is the number of stages required, c_i are weights and k_i depends on the time step and the function evaluation. An extension of the Runge-Kutta approach is the Runge-Kutta-Fehlberg (RKF) method. This consists in combining two RK methods of order p and $p-1$ with common intermediate steps. Assuming that the higher-order method provides a better result, it is possible to estimate the integration error of the lower-order integrator. This allows to adapt the integrator step-size such that the estimated integrator error is lower than a given user-defined threshold: the relative and absolute tolerance. Once the step size is estimated, the propagation can be performed with the lower-order method or with the higher-order method. In Tudat, the former approach is available for the integrators RKF4(5), RKF5(6) and RKF7(8) while the latter is approach (labelled Dormand-Prince method) is available for the integrator DOPRI8(7). These algorithms have shown to provide a good solution for a wide range of problems, and specially for high eccentricities.

Multi-step integration

Single-step methods such as RK use function evaluations from a single step to predict the following state. In turn, multi-step methods use information of several previously calculated steps. This approach requires to store the values from previous function evaluations but in general the number of function evaluations required for a given order of accuracy is reduced. The multi-step method included in Tudat, which will be considered in this algorithm selection is the Adams-Bashforth (explicit) and Adams -Moulton (implicit) methods. The general expression governing these methods is:

$$\begin{aligned} y\left(t_{n+1}\right) &= y\left(t_n\right) + \Delta t \sum_{j=a}^{N} b_j f\left(t_{n-j}, \bar{y}_{n-j}\right) \\ t_{n-j} &= t_n - j\Delta t \end{aligned} \tag{A.17}$$

where $a = 0$ for explicit methods and $a = -1$ for implicit methods. This method requires the knowledge of N data points for initialisation. For the implicit method, a predictor-corrector approach is followed, there the explicit method is used to estimate the state $y\left(t_{n+1}\right)$ and this value is used to evaluate the implicit method and correct the result. This combination is called the Adams-Bashforth-Moulton (ABM) method. This method is useful for low to moderate eccentricities and therefore should be considered as an option for this problem.

Extrapolation integration

The Bulirsch-Stoer (BS) integrator combines the concept of extrapolation with a multi-stage integrator. The concept consists in performing the integration of a given step with varying number of integration sub-steps. Assuming that with more sub-steps the solution will improve, the results from different number of sub-steps is extrapolated to infinite sub-steps. This approach is very useful for long-term integrations, for which there is no need for a dense output. However, this is not the case of the reference mission under study. To calculate the TCA of objects in LEO, a dense output is required and therefore this method is discarded.

A.3.2. Selection of numerical integration method

Integrator type

From the integrator types presented and available in Tudat, two families can be considered for to integrate the trajectory of objects in LEO with a dense output. These are the Runge-Kutta family (which includes the Dormand-Prince variation) and the Adams-Bashforth-Moulton which has variable order. In this section, without entering into the selection of the step-size or the tolerance, the integrator to be used in the problem to carry the Monte Carlo simulations for verification is selected. To do so, several studies which compare the performance of these integrator types are analysed.

Bradley (2015) studied the performance of a set of numerical integration methods to improve the computational performance of orbit propagation. The performance was studied in terms of the root-mean-square of the position error and the number of function calls. It was concluded that both for LEO and for highly eccentric orbits (Molniya) the DOPRI8(7) algorithm outperformed the rest of lower order integrators of the Runge-Kutta family, being the optimum pareto front. Somodi and Foldvary (2011) performed a similar analysis to compare the performance of the ABM, BS and the RK family. The Dormand-Prince integrator was not included in this study. This study found that for LEO, the ABM method outperformed the BS and RK4 by two orders of magnitude in the position error, and achieved this with shorter runtime.

Therefore, the methods to trade-off are the DOPRI8(7) and the ABM. Aristoff and Poore (2012) compares the performance of these two methods to the Gauss-Legendre implicit Runge-Kutta method (GL-IRK). Altough the latter shows to outperfrom DOPRI8(7) and ABM, this method is discarded since it is not included in Tudat and the benefit of coding it does not balance the improvement in accuracy and runtime. This study shows that the DOPRI8(7) and ABM have approximately the same accuracy and runtime, with the best method depending on the tolerance. Therefore the choice in terms of performance is not clear. To make a decision, two other criteria are considered: the use of the integrator method in similar problems and the simplicity. In other space debris collision analysis problems, the DOPRI8(7) algorithm is commonly used as the method to develop the verification solution as in the case of Valli et al. (2013); Jones and Doostan (2013). Adding this to the fact that the DOPRI8(7) method does not require previous data points for initialisation, this algorithm is selected to perform the numerical integration of the problem.

Integrator tolerance and step size

Once the DOPRI8(7) integrator is selected, the settings have to be specified. To do so, first it is decided whether a variable or fixed step size will be chosen and then, the bounds of the step size and the value of the absolute and relative tolerance is selected. In this election it must be taken into account that the DOPRI8(7) integrator misbehaves for very strict tolerances ($\leq 10^{-15}$). The selection of integrator settings depends on the performance of the integrator in terms of runtime and integration error.

To select the step size, it is first required to gain insight on the problem. From a review of studies that have performed numerical integration in similar scenarios, it is concluded that the characteristic step size for LEO integration is one second (Somodi and Foldvary, 2011; Hoogendoorn, 2016; Shuster, 2017). However, this value can also go as low as 10^{-4} seconds and as high as 30 seconds (Papanikolaou and Tsoulis, 2016). Taking the interval ($10^{-4} - 30$ s) as reference, a study of the position error as a function of step size will be performed to determine the boundary between rounding and truncation error. Once this threshold step size is encountered, the effect of the step size in runtime will be analysed. Since the numerical integration method will be used for verification purposes, the runtime is not a major concern, and solution accuracy will be of primordial importance.

A similar analysis can be performed to select the integrator tolerance. For similar problems using the DOPRI8(7) integrator the range of relative absolute tolerance included is ($10^{-9} - 10^{-14}$) (Valli et al., 2013; Jones and Doostan, 2013). The effect of tolerance in final position and velocity accuracy and runtime will be studied. Unless a tolerance results in prohibitive integrator time (e.g. resulting in a one-week propagation to generate 10^5 Monte Carlo runs), the tolerance which provides the highest accuracy is selected.

A.4. Statistical sampling and solving methods

The use of statistical method is crucial in the propagation of space debris collision probability. The main algorithm to be used relies on Taylor series integration by differential algebra and directly works with the statistical descriptors obtained from Taylor coefficients. However, a Monte Carlo approach is followed to validate the results and calculate the accuracy of the method. For this approach, statistical sampling and solving methods are required to generate the initial population of each random variable and estimate the final probability density function.

A.4.1. Statistical sampling

To validate the results from the DA-GMM uncertainty propagation, the initial states will be sampled according to their probability density function and propagated to the final state following a Monte Carlo approach. For the variables under study, three distributions will be sampled: normal, log-normal and uniform.

The most common method to obtain a random sample from a PDF is the inverse transformation method, which is based on the property that the cumulative density function (CDF) of the random variable x, $F(x)$ is uniformly distributed in the interval $[0,1]$ (Thomopoulos, 2013). Therefore, the random samples derived from a given probability distribution can be obtained from uniformly distributed random variables as:

$$x = F^{-1}(u) \tag{A.18}$$

where u is a uniform random number in $[0,1]$. The inverse transformation method is already implemented for the common probability distributions (including the ones required for this study) in commercial programming software such as Matlab and Python. Tudat also has a an integrated implementation of this method.

The problem has reduced to generating a uniform random sample. Multiple algorithms to generate random samples have been developed and are already integrated in commercial software. For reproducibility of the results it is important to document which algorithm is selected and which seed is used to generate the random value. Moreover, since different software will be used for propagation and post-processing, a single algorithm will be used in case the sampling method has to be reproduced by a different software. In Tudat, the Mersenne Twister method as described by Matsumoto and Nishimura (1998) is used by default to generate uniform random samples. This is one of the most widely used pseudo-random number generation algorithms and is also available in Matlab and Python. This algorithm is selected as the basis to generate pseudo-random uniform samples which in turn will be mapped to values of a random variable x which follows a given probability distribution.

A.4.2. Parameter estimation

While statistical sampling is used to draw individuals from a known probability distribution, parameter estimation allows to approximate the parameters of the distribution that governs a given data set. This will be required for two applications. First, to estimate the uncertainty distribution of the perturbing accelerations not included in the model simplification. From different orbits and initial conditions the difference between the simplified and the full model acceleration will be computed, and this data will be fitted to a given distribution. Ideally, the error could be approximated to a Gaussian distribution and it would be directly added as a state variable. Second, this method is used to validate the GMM assumption which states that for a large number of GMEs with low standard deviation, the final distribution of each GME that has been propagated through the nonlinear dynamics can still be modelled as a Gaussian. To test this assumption, the propagated samples drawn from an initial GME will be approximated by a Gaussian distribution and the distribution parameters will be compared to those obtained by the DA-GMM method. In Subsection A.4.3 the method to evaluate the goodness of fit is explained.

Matlab and Python contain built-in functions to estimate the parameters that approximate a data set to a number of distributions. In general, the method of maximum likelihood estimation is applied. For the Gaussian and log-normal distributions the parameters can be directly computed from the data-set without need of further fitting. It should be noted that since the data set is formed by random samples, the estimated parameters are also random variables. This means that if the two data sets are drawn from the same distribution, the estimated parameters will not necessarily be equal. To assess this effect and get a more accurate estimate of the distribution parameters, multiple sets of Monte Carlo simulations can be run, with the samples drawn from the same initial distribution to obtain an estimate of the range of values that the estimated parameters can obtain.

A.4.3. Hypothesis testing

The DA-GMM approach is based on the assumption that by dividing the initially Gaussian distribution into a large set of sub-distributions which are also Gaussian, the propagation of the individual elements through the non-linear dynamics will correctly approximate the final non-Gaussian distribution. This implies that the error in keeping the individual GMEs as Gaussian distributions through the propagation is negligible. Testing this hypothesis will be part of the validation process and will also be used to select the number of GMEs that is required. To this end, a goodness of fit test will be conducted on the propagated GMEs to assess if the hypothesis of keeping a Gaussian distribution can be accepted. Several goodness of fit procedures have been designed to test if a given data set is normally distributed. The most common tests are Pearson's χ^2 test, the Kolmogorov-Smirnov (KS) test and the Anderson-Darling (AD) test.

Agu and Francis (2018) evaluated the performance of the χ^2 test and the KS test applied on normal distributions. This study concluded that the KS goodness of fit is more precise than that of the χ^2 method. Razali and Yap (2011) performed a similar study to compare the power of several normality tests methods and concluded that the AD test is more powerful than the KS test. However, all methods have low reliability for small sample sizes. Since this variable is controllable in the case of a Monte Carlo approach, the sample size will be selected to be sufficiently large.

These tests are developed for univariate distributions, while the uncertainty in the satellite state that we wish to evaluate is a multi-variate distribution. Therefore, two algorithms to evaluate the uncertainty in a multi-variate distribution are used: Henze-Zirkler's Multivariate Normality Test (Henze and Zirkler, 1990) and Roy's normality test (Silvey, 1959)

Based on this, the five tests are performed on the Monte Carlo propagation results for multiple GMEs drawn from the initial Gaussian distribution. The number of GMEs must be selected so that the Gaussian assumption is valid for the propagated GMEs according to the three methods. Once this threshold is calculated, several runs of the DA-GMM method will be conducted with varying number of GMEs to observe the effect of this variable on the method error. The hypothesis testing algorithms are available in Matlab and Python as built-in functions and will form part of the pre-processing to calculate the required number of GMEs.

Kolmogorov-Smirnov
The KS test is based on comparing the sample to an empirical distribution (Gaussian in the case of a normality test). To do so, the samples are standardised and compared to a normal distribution and the conclusion of the test depends on the chosen significance level. The KS statistic is defined as:

$$D = \max_{1 \leq i \leq N} \left(F(Y_i) - \frac{i-1}{N}, \frac{i}{N} - F(Y_i) \right) \tag{A.19}$$

where F is the empirical cumulative distribution to which the samples are compared, Y_i are the sample points, and N is the number of samples. The result of the KS test is that the hypothesis regarding the form of the distribution (in this case that the data comes from a normal distribution) can be rejected if D is greater than the critical value for a specified significance level.

Anderson-Darling test
Following the same concept of the test statistic, the AD statistic is defined as:

$$A^2 = -N - \sum_{i=1}^{N} \frac{(2i-1)}{N} \left[\ln F(Y_i) + \ln(1 - F(Y_{N+1-i})) \right] \tag{A.20}$$

Again, the hypothesis regarding the form of the distribution can be rejected if A is greater than the critical value for a specified significance level.

χ^2 test
The chi-square (χ^2) test is an alternative to determine if the samples are randomly distributed. In this case, the statistic depends on the choice of bins o define the histogram of the distribution, since it is based on the frequency, O_i, and expected frequency E_i of each bin, i. The χ^2 statistic is defined as:

$$\chi^2 = \sum_{i=1}^{k} (O_i - E_i)^2 / E_i \tag{A.21}$$

where the expected frequency of each bin is defined as:

$$E_i = N(F(Y_u) - F(Y_l)), \tag{A.22}$$

with Y_u and Y_l being the upper and lower bounds of the data in bin i, respectively. Again, the hypothesis regarding the form of the distribution can be rejected if χ^2 is greater than the critical value for a specified significance level.

Henze-Zirkler test
The HZ statistic is based on measuring the distance between the observed and expected probability distributions, whixh is calculated as:

$$\mathrm{HZ} = \int |P(t) - Q(t)|^2 f_{b(t)} dt \tag{A.23}$$

where P and Q are the cumulative functions of the sampled data and empiric distribution respectively and $f_b(t)$ is a weight distribution given by:

$$w_\beta(t) = \left(2\pi\beta^2\right)^{-d/2} \exp\left(-\frac{\|t\|^2}{2\beta^2}\right) \tag{A.24}$$

being β a fixed constant, and d the number of dimensions.

Roy test
In this case, the the test relies on eigenvalues of the test matrix and can take into account the full multivariate distribution. It is an extension of the univariate Shapiro Wilk normality test. The details of this algorithm are complex and will not be discussed further. The purpose of these normality tests for this work is to provide an extra input in the selection of the optimum number of GMEs. Therefore, these tests are used as a black-box that outputs an acceptance/rejection of the null hypothesis for a given acceptance level.